Japan-Tunisia Workshop on Informatics
JTWIN 2001

University of Tsukuba
October 25-26, 2001

ISE–TR–01–184

Preface

This proceedings contains the papers presented at Japan-Tunisia Workshop on Informatics 2001 (JTWIN 2001) held at University of Tsukuba on October 25 - 26, 2001. The collected papers are the contributions of the invited authors in response to the invitation by the organizing committee of JTWIN.

JTWIN aims at bringing together Japanese and Tunisian researchers in informatics to promote cultural exchanges between the two countries in the field of information science and technology. JTWIN is conceived at the initiative of Japanese and Tunisian Governments that recognize the need to strengthen the ties between the two countries. Both envisage closer cooperations in the areas of strategic importance such as informatics.

JTWIN 2001 is the first forum that concentrates on creating the format of cooperation in academic and industry of the two nations. In the workshop we seek to exchange up-to-date research results as well as to identify topics of future cooperation. Particular emphasis (but not limited to) is on software science and engineering, and on computational intelligence.

We would like to thank the support of the Japan Society for the Promotion of Science (JSPS), Tunisian Embassy in Tokyo and University of Tsukuba.

October 2001 Tetsuo Ida

Organizing Committee:

Adel Cherif	Japan Advanced Institute of Science and Technology
Kaoru Hirota	Tokyo Institute of Technology
Tetsuo Ida	University of Tsukuba
Yasuhiko Minamide	University of Tsukuba

Table of Contents

Software Engineering

Computational Intelligence

Proof-Directed Compilation for Inter-operable Languages

Atsushi Ohori
School of Information Science
Japan Advanced Institute of Science and Technologies
Tasunokuchi, Ishikawa 923-1292, JAPAN
ohori@jaist.ac.jp

Abstract— **Due to extensive research on type theory of programming languages, the relationship between types and operational semantics is well understood, and a number of type safe and high-level programming languages have been developed. The type system and the semantics of most of these languages are however closed and incompatible with those of the other languages. In order to overcome this problem, we are developing a proof-directed framework for compiling an inter-operable language, and implementing a high-level programming language, IML – Inter-operable ML – within the framework. The guiding principle is Curry-Howard isomorphism, which enables us to implement the language as a series of proof transformations from the source language to the target machine code. Due to the universal nature of the underling intuitionistic logic, we can achieve inter-operability. This paper is a preliminary report on our ongoing project. We outline the goals, the rationale of its design, the underlying principle and the research issues. We hope that this will serve as a seed of further discussion and collaboration with the participants of Japan Tunisia Workshop on Informatics (JTWIN) and others.**

I. INTRODUCTION

An important achievement in theory of programming languages is the establishment of *static typing* discipline for type safe execution of a program. This is based on the *type soundness* property, which guarantees that a program having a static type τ yields a value of the type τ. Since error does not have any type, this property immediately implies that "a well typed program cannot go wrong" – a well recognized slogan advocated by the designer of the programming language ML [6]. This principle has been successfully used in number of type safe programming languages, including Standard ML [7], Haskell [5], and Java [4] to name a few.

The usual practice of implementing a programming language with a static type system has been to follow the following three steps.

1. To define an abstract semantics of the language.
2. To show that the semantics is a model of the syntactic theory of the type system.
3. To implement a compiler so that it correctly realizes the abstract semantics.

While the first two steps are rigorous mathematical development in type theory, the third step remains informal and ad-hoc. This implies that type safety of an actual implementation of a language directly depends on the implementor's (usually undocumented and informal) effort of correctly realizing the abstract semantics.

Since constructing a compiler and its associated runtime system is a complicated and error-prone process, it is difficult for the current ad-hoc method to take full advantage of formal results of type soundness in type theory. More seriously, this ad-hoc nature of compiler construction makes it difficult to scale type safety to certain language features. We will discuss two important ones below.

A. Type-Dependent Optimization

An ad-hoc implementation works reasonably well for a conventional functional language implementation using the untyped λ-calculus as an intermediate language. Since the untyped λ-calculus can be regarded as a representation of an element in a universal semantic domain, the type soundness theorem based on a universal domain such as that of [6] can be transfered to this implementation relatively easily.

The resulting implementation, however, suffers from inefficiency due to the lack of optimization specific to various data types. Except perhaps for interpreter-based prototype languages, this inefficiency is unacceptable for practical programming languages with advanced static type system. As an example, consider the function `name` defined below.

```
fun name(x) = x.Name
```

where $e.l$ is a term to denote operation to select the l field from a record e. This function is polymorphic in the sense that it can be applied to any record containing a `Name` field. Although the type of this function cannot be represented in the conventional ML-style type system [3], there are polymorphic type disciplines that can represent this form of polymorphism. Here we consider the one based on record kind [9]. In this work, this function is given the following type

$$\forall t_1{::}U.\forall t_2{::}\{\!\{\texttt{Name} : t_1\}\!\}.t_2 \to t_1$$

indicating the fact that `name` is a function of type $t_2 \to t_1$ for any t_1 and t_2 provided that the domain type t_2

is a record type containing a `Name` : t_1 field. Due to this flexible typing, `name` can be applied to any record containing a `Name` filed, and returns the value of that filed.

The issue we are concerned with is the semantics of this function. Under the usual abstract semantics where a labeled record is a mapping from a finite set of labels to values, the soundness of this typing is trivial. However, such abstract semantics corresponds to unacceptably inefficient naive implementation, where filed selection is done by dynamically searching for the desired label. In any practical production language, a record must be represented as a vector of values and field selection must be done by index operation or some equivalent efficient operation. The run-time semantics of the `name` function should then depend on the type of the argument. If we assume that a labeled record is represented as a vector of filed values sorted by labels, then the run-time semantics of the application of `name` to {`Age=21,Name="Joe"`} is to select the 2nd field from `(21,"Joe")` and the application of `name` to {`Name="Joe",Office="51B"`} is to select the first filed from `("Joe","51B")`.

This example demonstrates that the semantics of the compiled code may be quite different from that of the source program, and compilation may involve complicated optimization process. For such a language implementation, type soundness property for abstract semantics of the source language no longer implies type safety of the compiled code. In order to establish type safety of an implementation involving sophisticated optimization process, we need to refine the step 3 of ad-hoc approach to the following.

3-1 To define an intermediate language as a typed calculus.
3-2 To show the type soundness of the target language with respect to its operational semantics.
3-3 To define the necessary optimization process in a compiler as as a type-preserving translation from the source language to the intermediate language.

This process needs to be repeated for each language feature that requires non-trivial compilation.

We believe that this form of translation is essential for efficient implementation of high-level languages. For the case of record polymorphism, it is shown in [9] that one such type-preserving translation is possible. This and other existing approach to type-directed compilation are, however, still ad-hoc in the sense that they are specific translations between particular typed calculi. Our motivation is to generalize this paradigm so that each compilation process can be regarded as an instance of a well formalized general process. Such paradigm would enable us to establish a framework for systematic development of optimizing compilers.

B. *Inter-operability*

The other important language feature for which type safety is a subtle issue is inter-operability. In order to take full advantage of recently emerging network computing, programming languages should be able to communicate with each other by exchanging various resources including executable code, and dynamically using them. Achieving type safety is an important technical challenge in this newly emerging computation environment.

The ad-hoc implementation we outlined in Introduction only works for a closed system using a single language. Type safety is assured by the fact that the language compiler generates code that conforms to its type system under a number of assumptions on the run-time system of the language. These set of assumptions are determined by the language implementor's (reasonable but to large extent arbitrary) choices. As a consequence, two implementations sharing the same type structures cannot easily communicate each other.

To see the seriousness of this issue, consider representations of integers in Standard ML and C. In a typical implementation of Standard ML, an integer i is represented by the value $i * 2 + 1$ to satisfy the assumptions set by the run-time memory management system, while in C it is represented as i. In the case of ML and C, even the simplest primitive types cannot be easily shared – let alone any other data types such as records and functions. Ironically, when the language's type system becomes more advanced, it becomes more difficult to communicate with other languages. It is unfortunate that advanced and secure typing discipline becomes an obstacle in inter-language communication.

C. *IML – Inter-operable ML*

The goal of this work is to develop a compilation framework for both type-directed sophisticated compilation and type safe inter-operable implementation, and to implement a prototype programming language Inter-operable ML or IML.

Our initial investigation reveals that the technical machinery for type-directed compilation is essentially the same as the one needed for inter-operability. Type-directed compilation such as record polymorphism [8], [9], unboxed operational semantics of ML [14], specialization of polymorphism [12], is regarded as a process to fill the gap between abstract semantics of the high-level source language and that of a low-level machine. Inter-language communication has essentially the same structure; it involves a process of converting representations corresponding to different semantic assumptions. When the language has an advanced type system, such conversion can only be done using type information, as in type-directed compilation. A simple example can be found in inter-language RPC stub generation [13].

This and other type-directed approaches are however ad-hoc as we mentioned earlier. In order to establish a framework for compiling inter-operable languages, we need a universal principle on which each type-directed compilation can be based. Investigation of logical interpretation of compilation processes [10], [11] provides some hint. The careful analysis of low-level machine code reveals that a proof system of intuitionistic propositional logic is appropriate for representing types and operational semantics of low level code. These results suggest that proof theory of intuitionistic logic can serve as a common framework for inter-operable languages.

The goal of IML project is to integrate various techniques of type-directed compilation mentioned above in a proof-theoretical approach. This paper is a preliminary report on this project I have just started. I outline the goals, the rationale of its design, and the underlying principles. Instead of reporting specific technical results, I include discussion on existing problems, research issues and possible approaches and solutions. My hope is that this report will serve as a seed of further discussion and collaboration with the JTWIN participants and others.

The organization of the rest of this paper is as follows. Section II explains the logical approach for compilation. Section III is on type theoretical analysis of inter-operability. Section IV outlines the design of IML language. Section V describes implementation strategies. Section VI discusses some open problems and suggests possible approaches.

II. Logical Approach to Compilation

Our guiding principle in establishing a desired framework for compilation is *Curry-Howard isomorphism.* In its original form, this states that the typed λ-calculus is isomorphic to the intuitionistic natural deduction proof system. Since typed functional languages can naturally be regarded as a variant of the typed λ-calculus, this principle yields a logical interpretation of functional languages. Our claim is that this principle can be applied to most of programming languages by defining appropriate proof systems. Since all the proof systems are those of the same logic and therefore equivalent, translation between various programming languages can be represented through proof transformation.

In what follows, we explain this approach using a product type $A \times B$ as an example. Construction of a value of this type from a constant c_A of type A and a constant c_B of type B can be written in a number of ways dependent on the language used. In the λ-calculus and most of the high-level programming languages, it is written as the following term.

$$\texttt{(}c_A\texttt{,}c_B\texttt{)}$$

In a bytecode language, it is typically written as a sequence of instructions of the form

$$\texttt{const}\ c_A\texttt{;}\ \texttt{const}\ c_B\texttt{;}\ \texttt{pair;}$$

where `const` c pushes constant c onto the stack and `pair` pops the stack two elements and pushes the pair of the two. In an assembly language for a register-based machine, it would be written as a sequence of instructions of the form

$$\texttt{r1 <- }c_A\texttt{; r2 <- }c_B\texttt{; r3 <- mkeBlock(r1,r2)}$$

where `makeBlock` is a macro for allocating a memory block.

Although these are radically different in their syntax and formation rules, all of them can be interpreted as proofs of conjunction $A \wedge B$, whose constructive interpretation asserts that there are evidences of both A and B.

It is well known that λ term $\texttt{(}c_A\texttt{,}c_B\texttt{)}$ corresponds to the following derivation in natural deduction.

$$\frac{\Gamma \vdash c_A : A \quad \Gamma \vdash c_B : B}{\Gamma \vdash \texttt{(}c_A\texttt{,}c_B\texttt{)} : A \wedge B}$$

It is shown in [11] that analogous logical connection can be established for low-level code languages. The bytecode $\texttt{const(}c_A\texttt{);const(}c_B\texttt{);pair}$ produces a pair on the top of the stack, which is to be used by the code that follows. Let *cont* be the subsequent bytecode. Then the operation realized by these three instructions can be interpreted as the following logical inference step.

$$\cfrac{\cfrac{\cfrac{\Gamma, A \wedge B \vdash \mathit{cont} : C}{\Gamma, A, B \vdash \texttt{pair;}\mathit{cont} : C}}{\Gamma, A \vdash \texttt{const(}c_B\texttt{);pair;}\mathit{cont} : C}}{\Gamma \vdash \texttt{const(}c_A\texttt{);const(}c_B\texttt{);pair;}\mathit{cont} : C}$$

where the set of assumptions Γ in each logical sequent is interpreted as a description of the state of the run-time stack. When we change the representation of Γ to a named collection, then the above inference steps corresponds to code for a register machine we have shown above.

Other instructions can also be interpreted as inference rules of intuitionistic logic, and proof systems corresponding to low-level code languages can be defined. Compilation from the λ-calculus to a code language is then represented as a proof transformation in intuitionistic logic. For example, compilation scheme for $\texttt{(}M_a\texttt{,}M_b\texttt{)}$ is given by the following inductive equation:

$$[\![\texttt{(}M_A\texttt{,}M_B\texttt{)}]\!] = [\![M_A]\!];[\![M_B]\!];\mathrm{pair}$$

This compilation scheme corresponds to the proof transformation shown in Fig. 1, where *cont*[ ] is a context that uses the result of the pair $\texttt{(}M_A\texttt{,}M_b\texttt{)}$

$$\begin{array}{c} \Gamma \vdash M_A : A \quad \Gamma \vdash M_B : B \\ \hline \Gamma \vdash (M_A, M_b) : A\wedge B \\ \hline \vdots \\ \hline \Gamma \vdash \mathit{cont}[(M_A, M_b)] : C \end{array}$$

$$\Longrightarrow \qquad \begin{array}{c} \vdots \\ \hline \Gamma, A\wedge B \vdash \llbracket \mathit{cont} \rrbracket : C \\ \hline \Gamma, A, B \vdash \mathtt{pair};\llbracket \mathit{cont} \rrbracket : C \\ \hline \Gamma, A \vdash \llbracket M_B \rrbracket;\mathtt{pair};\llbracket \mathit{cont} \rrbracket : C \\ \hline \Gamma \vdash \llbracket M_A \rrbracket;\llbracket M_B \rrbracket;\mathtt{pair};\llbracket \mathit{cont} \rrbracket : C \end{array}$$

Fig. 1. Compilation of products through proof transformation

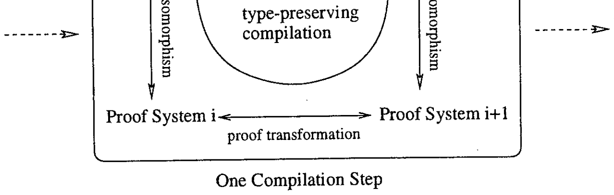
From these results, we have the following observations.

1. Properly defined and sufficiently rich programming language corresponds to a proof system of the intuitionistic propositional calculus. Curry-Howard isomorphism between the λ-calculus and the natural deduction proof system is a particular instance. Furthermore, this correspondence is not limited to high-level languages. Under suitable interpretation of its type, low-level languages including various machine code languages also correspond to proof systems of intuitionistic logic.
2. As being proof systems of the same underlying logic, these proof systems are mutually translatable. Furthermore, each translation can be specified as a composition of translation schemes between proof rules, yielding a clean inductively defined algorithm that preserves type and meaning. Program transformations including optimization and compilation are typical instances of proof transformation.

Based on these properties, we take intuitionistic propositional logic as the universal meta language for programming language design and implementation. In this approach, development of a compiler can be done by repeating the following two steps until the target language becomes a code language that can be executed by a machine.

1. Define the source and the target languages as proof systems.
2. Develop a proof transformation between the two proof systems.

Each compilation step has the following structure.



One Compilation Step

The type safety of the resulting implementation is a trivial corollary of this logical construction.

Furthermore, this structure opens up type safe inter language communication. Since the target machine code is represented as a proof system, the necessary interface is explicitly specified as a formal proof rule. In order to call other language implementation, we can generate a proof transformer that connects the two proof systems. Compared to the conventional adhoc notion of "wrapper" or "stub" generation, this approach achieves robust and type safe interaction. Moreover, due to the inductive nature of proof transformation, this proof-directed approach scales up to various complicated data types such as function types. In the next section, we examine the necessary features of type system for inter-operability.

III. Static Typing for External Access

One practical motivation of IML project is to develop a programming language suitable for those applications that manipulate external resources such as databases on the Internet. In addition to robust implementation, such a language should have a flexible polymorphic type system for representing external resources. This section discuss the rationale of the design decision of the type system of IML.

For a type system to represent external resources, the following two features are crucial.

1. *Data Abstraction.*
   In the perspective of typing, the property that distinguishes external resources is not merely the fact that they reside in some remote site but rather the fact that they do not conform to the set of assumptions presupposed by the type system. In this sense, external resources are abstract data whose implementation details cannot be known. To deals with those data, the type system must be able to manipulate abstract data types.
2. *Structural Polymorphism.*
   In accessing an external resource, the user is usually interested in some part of the external resource. Also, in many cases, only some part of the external resource can be available or representable. To deals with this situation, the type system of the language must be able to represent partial structure of a potentially large data. One type theoretical solution to this problem is to require any program manipulating a partially specified external resource to have a polymorphic type ranging over all the possible types of the data having the same partial structure. This achieves both the flexibility and type safety. The property of parametricity of polymorphism guarantees that any function having such a polymorphic type does not examine any part other than the partial structure explicitly specified.

A type system having these two features is we believe a prerequisite for the language. We see two possible approaches to construct such a type system.

One is to take an *object-oriented approach* and design a type system with sufficiently powerful subtyping relation. An object is an encapsulated entity suitable for representing an external resource. The mechanism of inheritance provides polymorphism necessary for representing partial structure of an external data. Unfortunately, however, various notions such as inheritance in object-oriented languages are inherently dynamic, and it is difficult to represent them as static typing rules. This property makes it rather difficult to develop a logical interpretation for an object-oriented language. In our view, object-oriented features are derived notions and are too strong to be taken as primitive mechanisms in a framework for language implementation.

The approach we decide to take is based on *polymorphic record typing* we have introduced in the previous section. By combining a mechanism for data abstraction, it can represent partially specified external objects. It is also entirely static so that it is relatively easier to develop a logical interpretation for it. In the rest of this section, we outline the features of polymorphic record calculus and explain a method for representing external data in its type system.

There are two typical approaches to polymorphic record typing. One is the work of [16] based on "row variables". In this approach, a partially known record structure is represented as $\rho^{\{l_1,\ldots,l_n\}}\{l_1 : \tau_1, \ldots, l_n : \tau_n\}$ which is the combination of the set of known fields $\{l_1 : \tau_1, \ldots, l_n : \tau_n\}$ and a row variable $\rho^{\{l_1,\ldots,l_n\}}$ representing the rest of the record not containing the fields $\{l_1, \ldots, l_n\}$. A polymorphic function operating on records has a type of the form:

$$\forall \rho^{\{l_1,\ldots,l_n\}}.\rho^{\{l_1,\ldots,l_n\}}\{l_1 : \tau_1, \ldots, l_n : \tau_n\} \rightarrow \tau$$

This approach yields powerful typing discipline for polymorphic record operations. However, since the variable $\rho^{\{l_1,\ldots,l_n\}}$ represents the unknown part of a record not containing the labels $\{l_1, \ldots, l_n\}$, the language with this type system is difficult to compile to a language with a different semantics of records, and is not entirely suitable for representing external objects.

The other alternative is the work of [9], where a partially known record structure is represented as a type variable t with a kind constraint $\{\!\{l_1 : \tau_1, \ldots, l_n : \tau_n\}\!\}$ on t indicating the set of known fields. A polymorphic function operating on records has a type of the form:

$$\forall t{::}\{\!\{l_1 : \tau_1, \ldots, l_n : \tau_n\}\!\}.t \rightarrow \tau$$

Different from the previous approach, the type variable t always denotes the entire record. Due to this property, it is possible to develop a proof-directed compilation of this calculus into a polymorphic calculus with vectors and index operation. The purpose of the work of [9] is to implement record operations efficiently. However, this same method can also be used to generate code suitable for external object access.

This approach can be combined with data abstraction. We regard an external data having a partial structure denoted by a record kind $\{\!\{l_1 : \tau_1, \ldots, l_n : \tau_n\}\!\}$ as an abstract object represented by a *kinded existential type* of the form:

$$\exists t{::}\{\!\{l_1 : \tau_1, \ldots, l_n : \tau_n\}\!\}.t$$

Applicable functions are restricted to those that only perform field access of the known fields. This restriction is precisely specified by the following typing rule.

$$\frac{\begin{array}{c}\Gamma \vdash M : \forall t{::}\{\!\{l_1 : \tau_1, \ldots, l_n : \tau_n\}\!\}.t \rightarrow \tau' \\ \Gamma \vdash N : \exists t{::}\{\!\{l_1 : \tau_1, \ldots, l_n : \tau_n\}\!\}.t\end{array}}{\Gamma \vdash \texttt{use}\ N\ \texttt{in}\ M : \tau'}$$

This rule guarantees type safe manipulation of an external object without knowing its exact structure.

In an actual implementation, however, the semantics of an external object differs among systems, and does not conform to the semantics of records in the accessing language. To compile a program involving this typing rule, we need to specialize the code of M for each different external object N. In [9], we have developed a method to compile a polymorphic record function to a code specialize for each application. Essentially the same mechanism can be used to generate specialized code for each different external object N. The necessary refinement is to translate a type of the form $\exists t{::}\{\!\{l_1 : \tau_1, \ldots, l_n : \tau_n\}\!\}.t$ to an infinite union type whose tag is unique to each external object. The record compilation can be refined to include tag-checking when generating the necessary access code.

IV. SOURCE LANGUAGE DESIGN

This section outline the source language IML we plan to develop. In addition to polymorphic record operations and external data access as explained above, IML uniformly integrates the following features in its type system.

- Extended pattern matching.
  Pattern matching of ML is extended to polymorphic records, external data, and database operations, so that the programmer can easily access external databases and other resources.
- Rank 1 polymorphism.
  One annoying limitation of ML polymorphism is its *value restriction*, which requires that a term that can be given a polymorphic type to be a syntactic value. This not only prohibits many useful type correct programs such as `(fn x => fn y =>(x,y)) 1` but it also introduces extra run-time cost in proof-directed compilation. IML type in-

ference extends ML polymorphism to rank 1 polymorphism, which eliminates most of the limitation of value restriction.

- database operations.
  A major practical motivation of IML design is to provide easy access to external databases from a high-level and type safe programming language such as ML. Combining the above features, IML provides a satisfactory answer. For example, the following function can be defined in IML.

```
fun wealthy(S) = select x
                 from x in S
                 where Salary.x > 100000;
```

  Using polymorphic record typing and external typing, the type system infers the following polymorphic type for this function.

$$\forall t :: \{\!\{\texttt{Salary} : \mathit{int}\}\!\}.Set(t_1) \to \mathit{int} \to Set(t_1)$$

  where $Set(t_1)$ is a database set type. By the typing mechanism discussed in the previous section, this function can be safely applied to external databases having the following type.

$$\exists t :: \{\!\{\texttt{Salary} : \mathit{int}, \ldots\}\!\}.Set(t)$$

  IML also integrates other useful features of polymorphic database languages proposed in [2].

V. PLAN AND STRATEGIES FOR DEVELOPING IML

In this section, we outline the general plan of our development of a proof-directed compilation framework and a prototype implementation of IML, and discusses various strategies to carry out the plan. As in an initial stage of research, many of them require further investigation.

The compilation process is a series of translations, each of which elaborates some of high-level features of IML until it become executable code. The major phases include the following.

1. The IML front-end
2. Patterns matching compilation
3. Compiling polymorphic record operations
4. A-normal from translation.
5. Abstract machine code generation.
6. Register allocation.
7. Access code generation and linking.

The main task of the front-end is to construct an explicitly typed second-order term through type inference. This is done by combining the methods of record type inference [9] and rank-1 type inference [15] with the additional treatment of existential types for external data we outlined in Section 2.

Each of the remaining 6 phases are realized as a proof transformation between typed calculi. We discuss each of them below.

Pattern Matching. In the existing works [1], pattern matching is characterized as a form of sequent calculus. Taking this view, pattern match compilation can be modeled by a proof transformation from a calculus with arbitrary nested patterns to a calculus with primitive patterns. The existing works, however, do not consider those features as constant patterns, non-exhausted patterns, and overlapping patterns. These extra features are the sources of complications in an actual compilation process. To deal with these features, we are currently developing a calculus with non-standard patterns using non-standard type theory. Our strategy is to use this new calculus to represent the practical pattern matching as the composition of the following proof transformation.

$$\begin{array}{c} \Lambda^{Pat?} : \text{the non-exact pattern calculus} \\ \Downarrow \\ \Lambda^{Pat} : \text{the exact pattern calculus} \\ \Downarrow \\ \Lambda^{Rec} : \text{the polymorphic record calculus} \end{array}$$

The second step can be done as an inductive proof transformation. The major complication is in the first step, which requires some heuristic strategy to minimize the run-time tag tests.

Record Compilation. This phase translates the polymorphic record calculus Λ^{Rec} to a polymorphic calculus with products and sums. The required translation is obtained by extending the method presented in [9] with the following additional mechanism. We consider ordinary record types as a member of an infinite and extensible disjoint union types containing all the possible external objects whose implementation differs. Record operation performs tag checking to distinguish ordinary records from external objects. For the case of an ordinary record, the required access information is an index value. In this case, the required tag information can be embedded in the index value in a way similar to tagging integer values for garbage collections. Using this technique, the proof-directed translation achieves both the efficiency of internal record access and transparent external object access.

A-normal Translation. We have shown in [10] that A-normal translation, which is a major translation step in compilation before code generation, can be modeled by a proof translation from the natural deduction system to a variant of sequent calculus known as Kleene's G3 system. Using this result, we implement this phase as a translation from the second-order calculus Λ^{prod} to $G3$.

Abstract Machine Code Generation. We must generate code that have clean and well defined semantics. To achieve this, we translate a given G3 proof to a proof of the logical abstract machine [11]. The logical abstract machine is a low level machine that can represent both bytecode languages and register transfer languages. Different from type systems of low-level languages so far proposed, its type system is isomorphic to a proof system of the intuitionistic propositional logic. This connection is obtained by considering memory manipulation instructions as structural rules in a proof system. For example, an instruction to load a register r_2 with the content of another register r_1 is modeled by the contraction rule as follows.

$$\frac{\Gamma\{r_1 : \tau, r_2 : \tau\} \vdash C \ : \ \tau}{\Gamma\{r_1 : \tau\} \vdash r_2 \leftarrow r_1; C \ : \ \tau}$$

indicating the property that the instruction changes the memory state $\Gamma\{r_1 : \tau\}$ to $\Gamma\{r_1 : \tau, r_2 : \tau\}$. Because of this connection, code generation is done as a clean inductive proof transformation from the G3 system. Moreover, since the semantics and the associated interface requirement is made explicit as a form of logical formula, connection to other language systems can be safely done.

Register Allocation. The logical abstract machine represents low-level code languages with unlimited memory, such as a bytecode language with an unbounded stack or a register transfer language with unbound number of registers. In order to generate native machine code, register allocation must be performed. Our recent study reveals that register allocation can be modeled by special structural rules in a proof system. We see above that the contraction rule corresponds to the instruction to copy a value between registers. In the above rule, we require that Γ does not contain r_2. This requirement yields a simple proof system where the number of assumptions in a sequent monotonically increases when we go up the proof tree. This corresponds to a register-transfer language that does not re-use any register. Register allocation and re-use can be introduced by explicitly adding a variant of weakening rule of the form

$$\frac{\Gamma\{r : \top\} \vdash C \ : \ \tau}{\Gamma\{r : \tau\} \vdash \mathtt{discard}(r)\mathtt{;}C \ : \ \tau}$$

where $\top$ is the type of garbage and `discard(`r`)` is a pseudo-instruction indicating the property that the content in r will not be used. We change the rule for loading a register above to

$$\frac{\Gamma\{r_1 : \tau, r_2 : \tau\} \vdash C \ : \ \tau}{\Gamma\{r_1 : \tau, r_2 : \top\} \vdash r_2 \leftarrow r_1; C \ : \ \tau}$$

requiring that the content of a register must be explicitly discarded before being loaded with a new value. Using these structural rules in controlled way, register allocation can be represented as a proof transformation.

Interface Generation and Linking. This phase is independent of compiling a program into an executable machine code. External data access is done by linking a foreign code. One promissing appraoch is to define a proof system for a foreign code in a way similar to that of the logical abstract machine, and to generate the necessary interface code. The sequential sequent calculus that underlies the logical abstract machine is designed to be as general as possible so that variety of sequentially executing low-level machine languages including Java Virtual Machine code and native machine code can be represented. Once this mapping is done then we can generate necessary interface code in a proof-directed way. Fig. 3 shows code generation for IML system.

The Implicitly Typed IML Program
$\Downarrow$ (type inference)
$\Lambda^{Pat?}$:the non-exact pattern calculus
$\Downarrow$ (pattern match elaboration)
Λ^{Pat}:the exact pattern calculus
$\Downarrow$ (match compilation)
Λ^{Rec}:the polymorphic record calculus
$\Downarrow$ (record compilation)
Λ^{Prod}:the polymorphic calculus with products
$\Downarrow$ (A-normal translation)
G3:Kleene's sequent calculus
$\Downarrow$ (compilation)
LAM:Logical Abstract Machine

Fig. 2. Compilation as a series of proof transformations

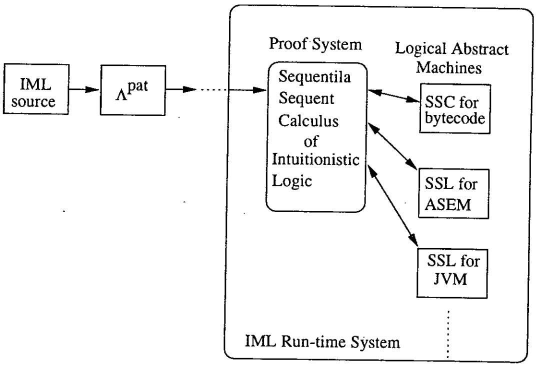


Fig. 3. Code generation for IML system

VI. Research Issues

The plan and strategies for establish a framework for proof-directed compiler and IML implementation discussed in the previous section are far from complete, and there are a number of issues to be investigated. In this section, we will elaborate some of them below.

Curry-Howard Isomorphism for Polymorphism The inter-operability of IML is based on type-directed compilation of record polymorphism, which was formulated in a second-order type system. For seamless integration of this compilation in our general paradigm of proof-direct compilation, we would like to establish its logical interpretation. Introducing the second-order quantification may not be so difficult. However, its constructive meaning remains to be investigated. This is related to the semantics of polymorphism in the theory of programming language. There have been two major approaches for interpreting second-order typing of the form $M : \forall t.\sigma$. One is to interpret it as the meta-level assertion saying that for all possible type τ, M has type $[\tau/t]\sigma$. The other is to interpret M as a function taking a type τ and yields a value of type $[\tau/t]\sigma$. Both of them are based on idealized view of polymorphism and do not reflect implementation of polymorphic functions. As we have explained, polymorphic functions require type-dependent compilation. This suggests that there should be refined interpretation of polymorphism. Discovering logical interpretation that accounts for type-dependent compilation would enable us to integrate polymorphism seamlessly in our framework of proof-directed compilation.

Realistic Pattern Calculus. We are investigating a realistic pattern matching compilation as a proof transformation. This requires us to construct a proof system for a pattern calculus with overlapping patterns and redundant patterns. Our initial investigation shows that it is possible to develop such a calculus using non-standard type theory. In addition to flexible typing, pattern matching compilation requires some extra-logical strategy for minimizing run-time cost. An interesting challenge is to parameterize a pattern matching proof transformation with some strategy such as a decision tree for examining patterns.

Garbage Collection and Other Resource Usage. As we have mentioned, logical abstract machines can be refined to represent register usage and allocation. This eliminates one simplifying assumption on unlimited number of registers. We believe that this approach can be extended to other resources such as stacks and environments. Such refinement can serve as a foundation for various optimizations such as closure conversion and uncurring optimization.

VII. Conclusions

This paper is a preliminary report on the framework of proof-directed compilation we are developing, and our implementing project of IML, Inter-operable ML. We have explained the rationale of the design of IML, the underlying proof-directed approach, and implementation strategies. We have also discussed some important research issues in establishing the framework. The proof-directed compilation we have advocated in this paper is a new approach to compilation and a number of interesting and important issues remain to be investigated.

Acknowledgments

The author thanks Kiyoshi Yamatodani for discussion on type system for external data access, and Tomoyuki Higuchi for discussion on logical analysis on Java Virtual Machine.

References

[1] B. Breazu-Tannen, D. Kesner, and L. Puel. A typed pattern calculus. In *Proc. IEEE Symposium on Logic in Computer Science*, pages 262–274, 1993.

[2] P. Buneman and A. Ohori. Polymorphism and type inference in database programming. *ACM Transactions on Database Systems*, 21(1):30–74, 1996.

[3] L. Damas and R. Milner. Principal type-schemes for functional programs. In *Proc. ACM Symposium on Principles of Programming Languages*, pages 207–212, 1982.

[4] J. Gosling, B. Joy, and G. Steele. *The Java Language Specification.* Addison-Wesley, 1996.

[5] P. Hudak, S. Peyton Jones, P. Wadler, B. Boutel, J. Fairbairn, J. Fasel, M. Guzman, K. Hammond, J. Hughes, T. Johnsson, D. Kieburtz, R. Nikhil, W. Partain, and J. Perterson. Report on programming language Haskel a non-strict, purely functional language version 1.2. *SIGPLAN Notices, Haskell special issue*, 27(5), 1992.

[6] R. Milner. A theory of type polymorphism in programming. *Journal of Computer and System Sciences*, 17:348–375, 1978.

[7] R. Milner, R. Tofte, M. Harper, and D. MacQueen. *The Definition of Standard ML.* The MIT Press, revised edition, 1997.

[8] A Ohori. A compilation method for ML-style polymorphic record calculi. In *Proc. ACM Symposium on Principles of Programming Languages*, pages 154–165, 1992.

[9] A. Ohori. A polymorphic record calculus and its compilation. *ACM Transactions on Programming Languages and Systems*, 17(6):844–895, 1995. A preliminary summary appeared at ACM POPL, 1992 under the title "A compilation method for ML-style polymorphic record calculi".

[10] A. Ohori. A Curry-Howard isomorphism for compilation and program execution. In *Proc. Typed Lambda Calculi and Applications, Springer LNCS 1581*, pages 258–179, 1999.

[11] A Ohori. The logical abstract machine: a Curry-Howard isomorphism for machine code. In *Proceedings of International Symposium on Functional and Logic Programming*, 1999.

[12] A. Ohori. Type-directed specialization of polymorphism. *Journal of Information and Computation*, 1999. To appear. A preliminary summary appeared in Proc. International Conference on Theoretical Aspects of Computer Software, Springer LNCS 1281, pages 107–137.

[13] A Ohori and K. Kato. Semantics for communication primitives in a polymorphic language. In *Proc. ACM Symposium on Principles of Programming Languages*, pages 99–112, 1993.

[14] A. Ohori and T. Takamizawa. A polymorphic unboxed calculus as an abstract machine for polymorphic languages. *J. Lisp and Symbolic Comput.*, 10(1):61–91, 1997.

[15] A. Ohori and N. Yoshida. Type inference with rank 1 polymorphism for type-directed compilation of ML. In *Proc. ACM International Conference on Functional Programming*, pages 160–171, 1999.

[16] D. Remy. Typechecking records and variants in a natural extension of ML. In *Proc. ACM Symposium on Principles of Programming Languages*, pages 242–249, 1989.

Simultaneous Checking of Completeness and Ground Confluence

Adel Bouhoula

LORIA – INRIA Lorraine
BP. 101, 54602 Villers-lès-Nancy cedex, France
E-mail: Adel.Bouhoula@loria.fr
URL: http://www.loria.fr/~bouhoula

Ecole Supérieure des Communications de Tunis
Cité Technologique des Communicaions
2083 Ariana, Tunisie
E-mail: Adel.Bouhoula@supcom.rnu.tn

Abstract

Algebraic specifications provide a powerful method for the specification of abstract data types in programming languages and software systems. Completeness and ground confluence are fundamental notions for building algebraic specifications in a correct and modular way. In this paper, we present a procedure for simultaneously checking completeness and ground confluence for specifications with free/non-free constructors and parameterized specifications. If the specification is not complete or not ground confluent, then our procedure will output the set of patterns on whose ground instances a function is not defined and it can easily identify the rules that break ground confluence. Our procedure is the first one which is complete and always terminates under the assumption of an oracle for deciding (joinable-) inductive properties. In contrast to previous work, our method does not rely on completion techniques and does not require the computation of critical pairs of the axioms. The method has been implemented in the prover SPIKE. *This system allowed us to prove the completeness and the ground confluence of many specifications in a completely automatic way where related techniques diverge or generate very complex proofs.*

1 Introduction

One of the important aspects in the process of program construction is the choice and the treatment of the basic data structures. Algebraic specification methods provide techniques for data abstraction and the validation and analysis of data structures. The basic idea of the algebraic approach [23] is to describe data structures by just giving the names of the different sets of data and the names of the basic functions and their characteristic properties. Therefore, algebraic specifications provide a powerful method for the specification of software systems in an abstract way and independently of its effective implementation. For the description of large data structures and complex systems, we can use parameterized specifications [21] which allow one to compose specifications in a modular way and to build larger specifications from smaller ones. Often algebraic specifications are built with conditional equations. Semantically, the motivation for this is the existence of initial models; operationally, the motivation is the ability to use term rewriting techniques for computing and automatic prototyping.

The completeness and ground confluence properties are very important for building algebraic specifications in a correct and modular way. Ground confluence is particularly useful to guarantee the refutational completeness of inductive theorem proving, which implies that every conjecture that is not valid in the initial model will be detected in finite time. This property is very important since practice shows that code is usually buggy, and therefore many properties expected to hold do not.

Completeness means that any ground (i.e., variable-free) term should return a result built upon constructor symbols. Many techniques have been developed for checking this property for non-conditional specifications [15, 16] and conditional specifications [3, 5]. Ground confluence guarantees the property of uniqueness in computation with ground terms. Several works have proposed sufficiency criteria for confluence of conditional systems [11, 17]. However, little work has been carried out on checking ground confluence. This is mostly due to the fact that the problem is much harder. Indeed, ground confluence is undecidable [18] even for equational theories with only unary function symbols. Plaisted has proposed a semantic confluence test [22], but he has not shown how his test can be automated. Completion techniques are used in [13, 12]. It is generally accepted that such techniques may be very inefficient since the completion procedure often diverges even for very small specifications. Other methods were developed in [14, 20, 1, 2] which do not rely on the completion framework. The key idea of these methods is to compute all critical pairs between axioms, and then to check each critical pair w.r.t. a sufficient criterion for ground confluence. The main drawback of these methods is that they generate a lot of critical pairs and most of the ground confluence criteria are restrictive and very hard to automate. Finally, we note that no known complete procedure exists for checking ground confluence, even for a restricted class of specifications.

The key idea of our method comes from the observation that completeness and ground confluence are interdependent. Indeed, to prove completeness for a conditional specification $\mathcal{SP}$, $\mathcal{SP}$ must be ground confluent [3], and to prove that $\mathcal{SP}$ is ground confluent using semantic techniques as in [22], $\mathcal{SP}$ must be complete. This observation motivates us to develop a procedure for simultaneously checking completeness and ground confluence for specifications with free/non-free constructors and parameterized specifications. Our procedure computes a pattern tree for every defined symbol, and identifies a set of rules whose inductive validity has to be checked. The leaves of the tree give a partition of the possible arguments for defined functions. If all the leaves are ground reducible and non-ambiguous, and if the identified rules are inductively valid, then we conclude that the given specification is complete and ground confluent. If the specification is not complete or not ground confluent, then our procedure will output the set of patterns on whose ground instances a function is not defined and it can easily detect the rules that break ground confluence.

As opposed to previous work, our procedure is complete and always terminates assuming an oracle for deciding (joinable-) inductive properties. Our approach is very powerful since it does not rely on completion techniques, and it does not require the computation of critical pairs of the axioms. The method has been implemented in the prover SPIKE. We have tested this system on several examples which have highlighted the simplicity and the efficiency of our approach compared to related techniques.

2 Basic Concepts

We assume that the reader is familiar with the basic concepts of rewriting and mathematical logic. We introduce the notation used later and refer to [10] for a more detailed presentation.

A many sorted signature Σ is a pair $(\mathcal{S}, \mathcal{F})$ (or simply $\mathcal{F}$, for short) where $\mathcal{S}$ is a set of *sorts* and $\mathcal{F}$ is a finite set of function symbols. Let $\mathcal{X}$ be a family of sorted variables. The set of well-sorted terms will be denoted by $\mathcal{T}(\mathcal{F}, \mathcal{X})$. We use capitals letters L, R, S, T, U, V, W for terms.

Terms are identified with finite labelled trees as usual. *Positions* are strings of positive integers. Λ is the empty string (root position). The *subterm* of M at position p is denoted by $M|_p$. The result of replacing $M|_p$ with N at position p in M is denoted by $M[N]_p$. This is also used to indicate that N is a subterm of M, in which case p may be omitted. We use $\mathcal{V}ar(M)$ for the set of variables of M. Variable-free terms are called *ground*. By $\mathcal{T}(\mathcal{F})$ we denote the set of ground terms. A term M is *linear* if every variable in $\mathcal{V}ar(M)$ occurs exactly once in M. We assume that each sort contains a ground term.

Well-sorted substitutions are written as $\{x_1 \mapsto M_1, \ldots, x_n \mapsto M_n\}$ where M_i and x_i are different terms of the same sort. If M_i is a variable $\forall i \in [1..n]$, the substitution is a *renaming*. We use small Greek letters for substitutions and postfix notation for their application.

We say that two terms S and T *unify* if there exists a substitution σ such that $S\sigma = T\sigma$. The set of unifiers of two given terms S, T possesses a unique minimal unifier with respect to subsumption, called the *most general unifier* of S and T. We say that a term S *matches* a term T with a substitution σ if $S\sigma = T$.

We denote by $\overline{U}$ the list (or vector) $(U_1, \ldots, U_n)$. Given two vectors $\overline{U}$ and $\overline{U'}$ of equal length over the respective sets E and E', and a binary relation $*$ over $E \times E'$, we use the notation $\overline{U} * \overline{U'}$ as an abbreviation for the vector $(U_1 * U'_1, \ldots, U_n * U'_n)$. When E and E' are sets of formulae, $\overline{U} * \overline{U'}$ will instead denote the formula $(U_1 * U'_1 \wedge \ldots \wedge U_n * U'_n)$.

We assume that the signature comes in two parts, a set of constructors $\mathcal{C}$, and a set of defined symbols $\mathcal{D}$ along with a rewrite ordering $\succ$, that is, an ordering on terms which is monotonic with respect to contexts and substitutions [9]. We use $\mathcal{T}(\mathcal{C}, \mathcal{X})$ and $\mathcal{T}(\mathcal{D}, \mathcal{X})$ for the respective sets of terms.

A *conditional rule* is an equation of the form: $\overline{V} = \overline{W} \Rightarrow f(\overline{L}) \rightarrow R$ satisfying a *reductivity* condition: (i) $f(\overline{L})$ and R are of the same sort, (ii) $f(\overline{L}) \succ R$, and (iii) $f(\overline{L}) \succ V, W$, $\forall V \in \overline{V}, \forall W \in \overline{W}$.

The term $f(\overline{L})$ is the *lefthand side* (lhs in short) of the rule. A set of reductive conditional rules is a *reductive conditional rewrite system*. A constructor is *free* if it is not the root of a left-hand side of a rule. We say that a rule $P \Rightarrow L \rightarrow R$ is *left-linear* if L is linear.

Definition 1 *(Critical Pair) If the lefthand side G of a rule $P \Rightarrow G \rightarrow D$ unifies, via a most general unifier σ, with a non-variable subterm S at position u in a lefthand side L of a rule $P' \Rightarrow L \rightarrow R$, then the conditional equation*

$$P\sigma \wedge P'\sigma \Rightarrow L\sigma[D\sigma]_u = R\sigma$$

is called a critical pair *between the two rules, where $L\sigma[D\sigma]_u$ is obtained by replacing S in L by D and applying σ.*

A critical pair $P \Rightarrow L = R$ is trivial *if L is identical to R.* ◇

To each non-left-linear rule $P \Rightarrow L \rightarrow R$, we associate its *linearized*[1] version $P' \wedge P'' \Rightarrow L' \rightarrow R'$, such that L' is

---

[1] This shows that left-linearity is not a real restriction when dealing with conditional rules.

```
specification: NAT
sorts Nat, Bool;
constructors:
0                 :              → Nat;
s _               : Nat          → Nat;
true, false       :              → Bool;
defined functions:
_ + _, _ * _      : Nat Nat      → Nat;
_ < _             : Nat Nat      → Bool;
even _, odd _     : Nat          → Bool;
axioms:
(1)   0+x → x;
(2)   x+0 → x;
(3)   s(x)+y → s(x+y);
(4)   x+s(y) → s(x+y);
(5)   (x+y)+z → x+(y+z);
(6)   0 * x → 0;
(7)   s(x) * y → (x * y)+y;
(8)   x * 0 → 0;
(9)   x<0 → false;
(10)  0<s(x) → true;
(11)  s(x)<s(y) → x<y;
(12)  even(0) → true ;
(13)  even(s(0)) → false ;
(14)  even(s(s(x))) → even(x) ;
(15)  even(x)=true ⇒ odd(x) → false ;
(16)  even(s(x))=true ⇒ odd(x) → true ;
(17)  even(x+x) → true ;
(18)  even(s(x+x)) → false ;
(19)  even(x+(y+y)) → even(x) ;
(20)  odd(s(s(x))) → odd(x) ;
(21)  odd(x+x) → false ;
(22)  odd(s(x+x)) → true ;
(23)  odd(x)=true ∧ odd(y)=true ⇒ odd(x+y) → false ;
(24)  even(x)=true ∧ even(y)=true ⇒ even(x+y) → true ;
(25)  odd(x)=true ∧ odd(y)=false ⇒ odd(x+y) → true ;
(26)  even(x)=true ∧ even(y)=false ⇒ even(x+y) → false ;
(27)  even(x * s(s(y))) → even(x * y) ;
(28)  even(x)=true ⇒ even(x * y) → true ;
(29)  odd(x)=true ∧ odd(y)=true ⇒ even(x * y) → false ;
end
```

Figure 1: A specification of natural numbers

linear, $L = L'\sigma$ for some renaming σ, $R = R'\sigma$, $P = P'\sigma$, and $x = y \in P''$ $\forall x,\ y \in \mathcal{V}ar(L')$ such that $x\sigma = y\sigma$.

A conditional rule is used to rewrite terms by replacing an instance of the left-hand side with the corresponding instance of the right-hand side (but not in the opposite direction) provided that the conditions hold. The conditions are checked recursively. Termination is ensured because the conditions are smaller (w.r.t. $\succ$) than the left-hand side. Now we introduce the notion of term rewriting (w.r.t. $\succ$) with conditional rules:

Definition 2 *(Conditional Rewriting) Let $\mathcal{R}$ be a conditional rewrite system. Let T be a term and u a position in T. We write: $T[L\sigma]_u \xrightarrow[\mathcal{R}]{} T[R\sigma]_u$ if there is a substitution σ and a conditional rule $\overline{V} = \overline{W} \Rightarrow L \rightarrow R$ in $\mathcal{R}$ such that $\overline{V}\sigma \downarrow_{\mathcal{R}} \overline{W}\sigma$, where $V \downarrow_{\mathcal{R}} W$ stands for $V \xrightarrow[\mathcal{R}]{*} U \xleftarrow[\mathcal{R}]{*} W$ for some term U.* ◇

A term S is *irreducible* by $\mathcal{R}$ if there is no T such that $S \xrightarrow[\mathcal{R}]{} T$. A substitution σ is *irreducible* by $\mathcal{R}$ if $x\sigma$ is *irreducible* by $\mathcal{R}$ for every variable x of its domain. We also use $\xrightarrow[\mathcal{R}]{*}$ and $\xrightarrow[\mathcal{R}]{+}$ for, respectively, the reflexive, transitive, and the transitive closures of a rewrite relation $\xrightarrow[\mathcal{R}]{}$. A term T is *ground reducible* if $T\gamma$ is reducible for every irreducible ground substitution γ.

Being reductive, the rewrite relation $\xrightarrow[\mathcal{R}]{}$ is terminating, i.e., there is no infinite chain of terms $T_1, T_2, \ldots, T_n$ such that $T_i \xrightarrow[\mathcal{R}]{} T_{i+1}$ for all i.

A formula φ is a *deductive theorem* of $\mathcal{R}$ if it is valid in every model of $\mathcal{R}$. This will be denoted by $\mathcal{R} \models \varphi$. A formula φ is an *inductive theorem* of $\mathcal{R}$ if it is valid in the initial model of $\mathcal{R}$. This will be denoted by $\mathcal{R} \models_{\mathcal{I}nd} \varphi$.

3 Completeness and Ground Confluence

When for all possible arguments the result of a defined operator can be expressed with constructors only, we say that this operator is completely defined w.r.t. the constructors.

Definition 3 *Let $\mathcal{R}$ be a rewrite system. The operator $f \in \mathcal{D}$ is completely defined w.r.t. $\mathcal{C}$ iff for all $T_1, \ldots, T_n$ in $\mathcal{T}(\mathcal{C})$, there exists T in $\mathcal{T}(\mathcal{C})$ such that $f(T_1, \ldots, T_n) \xrightarrow[\mathcal{R}]{+} T$. We say that $\mathcal{R}$ is complete iff each defined operator $f \in \mathcal{D}$ is completely defined w.r.t. $\mathcal{C}$.* ◇

Ground confluence guarantees the property of uniqueness in computation with ground terms.

Definition 4 *Let $\mathcal{R}$ be a rewrite system. We say that $\mathcal{R}$ is ground confluent iff for any ground terms U, V, $W \in \mathcal{T}(\mathcal{F})$, if $V \xleftarrow[\mathcal{R}]{*} U \xrightarrow[\mathcal{R}]{*} W$, then $V \downarrow_{\mathcal{R}} W$.* ◇

The specification given in Figure 1 is complete and ground confluent as proved in Example 13. Now, if we remove the rule $x < 0 \rightarrow false$ from Figure 1, then the specification NAT will be not complete, since, for example, the term $0 < 0$ cannot be reduced to a constructor term. If we change the rule $even(s(x)) = true \Rightarrow odd(x) \rightarrow true$ as $even(s(s(x))) = true \Rightarrow odd(x) \rightarrow true$, then the specification NAT will not be ground confluent, since, for example, $false \xleftarrow[\mathcal{R}]{} odd(0) \xrightarrow[\mathcal{R}]{} true$, and $false$ and $true$ are in normal form.

4 How to Check Completeness and Ground Confluence

The main idea behind our test for completeness and ground confluence is to compute a *pattern tree* for every defined operator f in $\mathcal{D}$. A pattern tree for f is a tree whose nodes are labelled by patterns, whose root is labelled by the initial pattern $f(x_1,\dots,x_n)$ where n is the arity of f and $x_1,\dots,x_n$ are distinct variables, and such that the successors of any internal node labelled by the pattern $f(\overline{T})$ are obtained by covering the set of values of an induction variable in $f(\overline{T})$. The restriction to induction variables allows us to build a pattern tree which captures the structure of the axioms. To compute pattern trees, we use the following notions.

Definition 5 *We call the* structural scheme *of sort s, denoted by $\mathcal{SC}(s)$, the set of terms $c(x_1,\dots,x_n)$ such that c is a constructor term with range s, and arity n, and $x_1,\dots,x_n$ are distinct variables. Note that $x_1,\dots,x_n$ is empty if c is a constant symbol.* ◇

For the example of Figure 1, $\mathcal{SC}(Nat) = \{0, s(x)\}$.

In the following, we call a *pattern* a term of the form $f(\overline{T})$ such that $f \in \mathcal{D}$ and $T_i \in \mathcal{T}(\mathcal{C},\mathcal{X})$ for every $T_i \in \overline{T}$.

Definition 6 *The set $\mathcal{I}nd\mathcal{P}os(f)$ of* induction positions *of $f \in \mathcal{F}$ is the set of positions $p \neq \Lambda$ such that there exists in $\mathcal{R}$ a rewrite rule of lefthand side $f(\overline{L})$ such that $L_i \in \mathcal{T}(\mathcal{C},\mathcal{X})$ for each $L_i \in \overline{L}$, and p is the position in $f(\overline{L})$ of a non-variable subterm.*

The set of induction variables *of a term T, written $\mathcal{I}nd\mathcal{V}ar(T)$, is the subset of $\mathcal{V}ar(T)$ whose elements occur in a subterm of T of the form $f(\overline{S})$, such that $S_i \in \mathcal{T}(\mathcal{C},\mathcal{X})$ for each $S_i \in \overline{S}$, at an induction position of f.* ◇

In the example of Figure 1, $+, *, <, even, odd$ have the respective sets of induction positions $\{1,2\}$, $\{1,2\}$, $\{1,2\}$, $\{1,1.1\}$, $\emptyset$, and x is an induction variable of $even(x)$.

Definition 7 *A pattern T is* strongly reducible *if (i) T is reducible, or (ii) the formula $P_1\sigma_1 \vee \ldots \vee P_n\sigma_n$ is an inductive theorem of $\mathcal{R}$, where $\{P_i \Rightarrow L_i \rightarrow R_i\}_{i\in[1..n]}$ is the non-empty set of linearized rules in $\mathcal{R}$ such that each L_i matches T with the substitution σ_i.* ◇

Thus, if a pattern T is *strongly reducible*, then all its ground instances are reducible. In Figure 1, the pattern $odd(x)$ is strongly reducible since $even(x) = true \vee even(s(x)) = true$ is an inductive theorem. However, the pattern $even(x)$ is not strongly reducible since there is no axiom whose lefthand side matches $even(x)$.

Definition 8 *A formula $\varphi = \overline{U} = \overline{V} \Rightarrow \overline{G} = \overline{D}$ is a* joinable-inductive theorem *of $\mathcal{R}$ if and only if for each ground substitution σ, if $\overline{U}\sigma \downarrow_{\mathcal{R}} \overline{V}\sigma$, then there exists $G = D \in \overline{G} = \overline{D}$ such that $G\sigma \downarrow_{\mathcal{R}} D\sigma$.* ◇

Of course, all joinable-inductive theorems are inductive theorems. It is easy to show that both notions coincide if $\mathcal{R}$ is ground confluent.

In [7, 4], we have developed *test set induction.* It is a goal-directed proof technique which combines the full power of explicit induction and proof by consistency. It works by computing an appropriate explicit induction scheme called *a test set*, to trigger the induction proof, and then applies a refutation principle using proof by consistency techniques. This method has been implemented in the prover SPIKE. This system is particularly powerful for mechanizing simultaneous induction and it allows us to prove inductive properties with less interaction than other provers, as exemplified by the proof of the Gilbreath card trick [7].

To check inductive joinability, we can use the inference system obtained by our inference system given in [7], by assuming that an induction hypothesis or a conjecture which is not yet proved can be used in the simplification process provided it is a reductive rule of the form:

$$P \Rightarrow L \rightarrow R$$

where R is ground irreducible (for example R is a constructor term if the constructors are free). We can easily show that the obtained inference system is sound and refutationally complete, even if the axioms are not complete and not ground confluent.

Definition 9 *A pattern T is* non-ambiguous *if for all rules $P_1 \Rightarrow L_1 \rightarrow R_1$ and $P_2 \Rightarrow L_2 \rightarrow R_2$ in $\mathcal{R}$ such that $T = L_1\sigma_1 = L_2\sigma_2$, the formula $P_1\sigma_1 \wedge P_2\sigma_2 \Rightarrow R_1\sigma_1 = R_2\sigma_2$ is a joinable-inductive theorem of $\mathcal{R}$* ◇

For the example of Figure 1, the pattern $odd(x)$ is non-ambiguous since

$$even(x) = true \wedge even(s(x)) = true \Rightarrow false = true.$$

is a joinable-inductive theorem.

Let $\mathcal{R}$ be the following rewriting system: $\{f(x) \rightarrow 0,\ f(x) \rightarrow g(x),\ g(0) \rightarrow 0,\ g(x) = 0 \Rightarrow g(s(x)) \rightarrow 0\}$. $f(x)$ is non-ambiguous since there are only two rules in $\mathcal{R}$ whose lefthand sides match $f(x)$ and $g(x) = 0$ is a joinable-inductive theorem.

5 Specifications with Free Constructors

Our procedure for checking the completeness and ground confluence of $\mathcal{R}$ is presented in Figure 2 as a set of inference rules operating on $(\mathcal{P}, \mathcal{MP}, \mathcal{AP}, \mathcal{UR}, \mathcal{NP})$, where $\mathcal{P}$ is a set of patterns labelling the leaves of the tree constructed so far, $\mathcal{MP}$ is its set of non-strongly reducible leaves, $\mathcal{AP}$ is its set of ambiguous leaves, $\mathcal{UR}$ is the set of rules in $\mathcal{R}$ whose lefthand sides match the leaves of the tree, and $\mathcal{NP}$ is the set of patterns for which we still must compute their pattern trees.

Decompose Variable applies when a pattern T is not strongly reducible and has an induction variable x of sort s.

Then, it instantiates x by terms in the structural scheme of s. **Checking Leaf** applies when a leaf T is found such that **Decompose Variable** cannot be applied. **Success** applies when the sets $\mathcal{P}$, $\mathcal{MP}$, $\mathcal{AP}$ and $\mathcal{NP}$ are empty and all the rules in $\mathcal{R} - \mathcal{UR}$ are inductive theorems of $\mathcal{UR}$. We can then conclude that $\mathcal{R}$ is complete and ground confluent (see Theorem 11). **Ground Confluence** applies when the sets $\mathcal{P}$, $\mathcal{AP}$ and $\mathcal{NP}$ are empty and all the rules in $\mathcal{R} - \mathcal{UR}$ are inductive theorems of $\mathcal{UR}$. We can then conclude that $\mathcal{R}$ is ground confluent (see Theorem 11). **Missing Patterns** applies when the sets $\mathcal{P}$ and $\mathcal{NP}$ are empty, but $\mathcal{MP}$ is not. In this case, $\mathcal{R}$ is not complete (see Theorem 11), the user is prompted to complete the specification of the patterns in $\mathcal{MP}$. **Ambiguous Patterns** applies when the sets $\mathcal{P}$ and $\mathcal{NP}$ are empty, but $\mathcal{AP}$ is not. Then, $\mathcal{R}$ is not ground confluent (see Theorem 11), the user is prompted to correct the specification to make the patterns in $\mathcal{AP}$ non-ambiguous. **Non-Valid Rules** applies when the sets $\mathcal{P}$, $\mathcal{MP}$, $\mathcal{AP}$ and $\mathcal{NP}$ are empty, but some rules in $\mathcal{R} - \mathcal{UR}$ are not inductive theorems of $\mathcal{UR}$. In this case, $\mathcal{R}$ is not ground confluent (see Theorem 11), the user is prompted to correct the specification to make the rules in $\mathcal{R} - \mathcal{UR}$ inductively valid.

To use the inference system $\mathcal{CGC}$, we start by the application of the rule **Initialization,** then we saturate the application of the rules **Checking New Function, Decompose Variable** and **Checking Leaf** until the sets $\mathcal{P}$ and $\mathcal{NP}$ will be empty. Finally, one of the rules **Success, Ground Confluence, Missing Patterns, Ambiguous Patterns** or **Non-Valid Rules** will be applied.

The height of the pattern tree, computed by the inference system $\mathcal{CGC}$, is bounded. This result is shown by the following lemma:

Lemma 10 *The pattern trees computed by the inference system $\mathcal{CGC}$ are finite.*

Proof: Let $f \in \mathcal{D}$. The rules of $\mathcal{R}$ which have the function symbol f at the top form a finite set. This means that the set $\mathcal{I}nd\mathcal{P}os(f)$ is finite too. As a consequence the set $\mathcal{I}nd\mathcal{V}ar(T)$ decreases during the construction of the tree since consecutive grafts in the same branch of the tree are made at deeper and deeper positions. Consequently, the height of the pattern tree is bounded. □

We can now address the soundness [2] of our method:

Theorem 11 (Soundness) *Let $\mathcal{R}$ be a reductive conditional rewriting system over free constructors. If $\mathcal{CGC}$ succeeds then $\mathcal{R}$ is complete and ground confluent. If the rule* **Ground Confluence** *applies, then $\mathcal{R}$ is ground confluent.*

Initialization:

$$\frac{}{(\emptyset,\ \emptyset,\ \emptyset,\ \emptyset,\ \bigcup_{f\in\mathcal{D}}\{f(\overline{x})\})}$$

Checking New Function:

$$\frac{(\emptyset,\ \mathcal{MP},\ \mathcal{AP},\ \mathcal{UR},\ \mathcal{NP}\cup\{T\})}{(\{T\},\ \mathcal{MP},\ \mathcal{AP},\ \mathcal{UR},\ \mathcal{NP})}$$

Decompose Variable:

$$\frac{(\mathcal{P}\cup\{T[x]_p\},\ \mathcal{MP},\ \mathcal{AP},\ \mathcal{UR},\ \mathcal{NP})}{(\mathcal{P}\cup\bigcup_{i\in I}\{T[S_i]_p\},\ \mathcal{MP},\ \mathcal{AP},\ \mathcal{UR},\ \mathcal{NP})}$$

`if` T is not strongly reducible and $x \in \mathcal{I}nd\mathcal{V}ar(T)$

`where` $\{S_i\}_{i\in I}$ is a structural scheme of the sort of x

Checking Leaf:

$$\frac{(\mathcal{P}\cup\{T\},\ \mathcal{MP},\ \mathcal{AP},\ \mathcal{UR},\ \mathcal{NP})}{(\mathcal{P},\ \mathcal{MP}\cup\mathcal{MP}',\ \mathcal{AP}\cup\mathcal{AP}',\ \mathcal{UR}\cup\mathcal{UR}',\ \mathcal{NP})}$$

`if` T is strongly reducible or $\mathcal{I}nd\mathcal{V}ar(T) = \emptyset$

$$\texttt{where}\ \begin{cases} \mathcal{MP}' = \text{ if } T \text{ is strongly reducible then } \emptyset \text{ else } \{T\} \\ \mathcal{AP}' = \text{ if } T \text{ is ambiguous then } \{T\} \text{ else } \emptyset \\ \mathcal{UR}' \text{ is the set of rules in } \mathcal{R} \text{ whose lhs matches } T \end{cases}$$

Success:

$$\frac{(\emptyset,\ \emptyset,\ \emptyset,\ \mathcal{UR},\ \emptyset)}{\mathcal{R} \text{ is complete and ground confluent}}$$

`if` $\mathcal{UR} \models_{\mathcal{I}nd} (\mathcal{R} - \mathcal{UR})$

Ground Confluence:

$$\frac{(\emptyset,\ \mathcal{MP},\ \emptyset,\ \mathcal{UR},\ \emptyset)}{\mathcal{R} \text{ is ground confluent}}$$

`if` $\mathcal{UR} \models_{\mathcal{I}nd} (\mathcal{R} - \mathcal{UR})$

Missing Patterns:

$$\frac{(\emptyset,\ \mathcal{MP},\ \mathcal{AP},\ \mathcal{UR},\ \emptyset)}{\mathcal{MP}} \qquad \%\ \mathcal{R} \text{ is not complete}$$

`if` $\mathcal{MP} \neq \emptyset$

Ambiguous Patterns:

$$\frac{(\emptyset,\ \mathcal{MP},\ \mathcal{AP},\ \mathcal{UR},\ \emptyset)}{\mathcal{AP}} \qquad \%\ \mathcal{R} \text{ is not ground confluent}$$

`if` $\mathcal{AP} \neq \emptyset$

Non-Valid Rules:

$$\frac{(\emptyset,\ \emptyset,\ \emptyset,\ \mathcal{UR},\ \emptyset)}{\mathcal{NVR}} \qquad \%\ \mathcal{R} \text{ is not ground confluent}$$

`if` $\mathcal{NVR} \subseteq (\mathcal{R} - \mathcal{UR})$ and $\mathcal{UR} \not\models_{\mathcal{I}nd} \mathcal{NVR}$

Figure 2: $\mathcal{CGC}$: Rules for completeness and ground confluence of $\mathcal{R}$

---

[2] The soundness proof of our method shows why we need to simultaneously check completeness and ground confluence. Indeed, if $\mathcal{CGC}$ succeeds, then to prove that $\mathcal{R}$ is complete we first have to show that $\mathcal{R}$ is ground confluent. If **Non-Valid Rules** applies, then to prove that $\mathcal{R}$ is not ground confluent, we first have to show that $\mathcal{UR}$ is complete.

If the rule **Missing Patterns** *applies, then $\mathcal{R}$ is not complete. If one of the rules* **Ambiguous Patterns** *or* **Non-Valid Rules** *applies, then $\mathcal{R}$ is not ground confluent.*

Proof: **(i)** Let us prove that if $\mathcal{CGC}$ succeeds then $\mathcal{R}$ is complete and ground confluent. To show that $\mathcal{R}$ is ground confluent, it is sufficient to show that $\mathcal{UR}$ is ground confluent since the rules in $\mathcal{R}-\mathcal{UR}$ are inductive theorems of $\mathcal{UR}$. Since $\mathcal{AP}$ is empty (i.e., all the leaves of the pattern trees are non-ambiguous) and all the rules in $\mathcal{UR}$ are of the form $P \Rightarrow f(\overline{T}) \rightarrow R$ where $f \in \mathcal{D}$ and $T_i \in \mathcal{T}(\mathcal{C}, \mathcal{X})$ for each $T_i \in \overline{T}$, then all critical pairs between rules in $\mathcal{UR}$ are trivial or joinable-inductive theorems. Therefore, we conclude that $\mathcal{UR}$ is ground confluent.

Now, let us show that $\mathcal{R}$ is complete. Let $T \in \mathcal{T}(\mathcal{F})$ and T' be the normal form of T with respect to $\mathcal{R}$. If T' is a constructor term, we are done. Otherwise, T' must contain a subterm T'' of the form $g(\overline{T})$ where $g \in \mathcal{D}$ and for all $i \in [1..n]$, $T_i \in \mathcal{T}(\mathcal{C})$. Since the leaves of the pattern tree exhaust all cases by construction, this subterm must be an instance of a leaf S. Since S must be strongly reducible, $\mathcal{R}$ is ground confluent and the rewrite relation $\xrightarrow[\mathcal{R}]{}$ is terminating, we can easily show that T'' is reducible by $\mathcal{R}$. This contradicts the fact that T' is in normal form.

(ii) If the rule **Ground Confluence** applies, then by following the same reasoning as in (i), we conclude that $\mathcal{R}$ is ground confluent.

(iii) If the rule **Missing Patterns** applies, then there exists a pattern $f(\overline{T})$ which is not strongly reducible and $\mathcal{I}nd\mathcal{V}ar(f(\overline{T})) = \emptyset$. Assume that $f(\overline{T})$ is not matched by any lefthand side of an axiom in $\mathcal{R}$. Let $f(\overline{T})\tau$ be a ground instance of $f(\overline{T})$ such that τ is built upon constructor symbols. Since $\mathcal{I}nd\mathcal{V}ar(f(\overline{T})) = \emptyset$, then $f(\overline{T})\tau$ is not matched by any lefthand side of an axiom in $\mathcal{R}$. So $f(\overline{T})\tau$ is irreducible at the root. On the other hand, $f(\overline{T})\tau$ cannot be reduced to a non-root position, since the constructors are free. Therefore we conclude that $\mathcal{R}$ is not complete. Otherwise, $f(\overline{T})$ is matched by the lefthand sides of n linearized rules $\mathcal{LR} = \{P_i \Rightarrow L_i \rightarrow R_i\}_{i\in[1..n]}$ but the formula $P_1\sigma_1 \vee \ldots \vee P_n\sigma_n$ is not an inductive theorem of $\mathcal{R}$. Then, there exists a ground and irreducible substitution τ such that $R \not\models P_1\sigma_1\tau \vee \ldots \vee P_n\sigma_n\tau$. We can easily show that $f(\overline{T})\tau$ is irreducible by $\mathcal{R}$. Therefore we conclude that $\mathcal{R}$ is not complete.

(iv) If the rule **Ambiguous Patterns** applies, we can easily show that there exists three ground terms $U, V, W \in \mathcal{T}(\mathcal{F})$ such that $V \xleftarrow[\mathcal{R}]{*} U \xrightarrow[\mathcal{R}]{*} W$ but $V \not\downarrow_{\mathcal{R}} W$. Therefore, we conclude that $\mathcal{R}$ is not ground confluent.

(v) Assume that the rule **Non-Valid Rules** applies, then $\mathcal{MP}$ and $\mathcal{AP}$ are empty, therefore $\mathcal{UR}$ is complete and ground confluent as shown in (i). On the other hand, there exists a rule $\overline{U} = \overline{V} \Rightarrow G \rightarrow D$ in $\mathcal{R}-\mathcal{UR}$ which is not an inductive theorem of $\mathcal{UR}$. Hence, there exists a ground substitution τ such that: $\mathcal{UR} \models \overline{U}\tau = \overline{V}\tau$ and $G\tau \not\downarrow_{\mathcal{UR}} D\tau$. Since $\mathcal{UR}$ is ground confluent and the rewrite relation $\xrightarrow[\mathcal{UR}]{}$ is terminating, all equalities in $\overline{U}\tau = \overline{V}\tau$ can be proved by normalization, and therefore the rule $\overline{U} = \overline{V} \Rightarrow G \rightarrow D$ can be applied to simplify $G\tau$ into $D\tau$. Now, let G' (resp., D') be the normal form of $G\tau$ (resp., $D\tau$) via $\mathcal{UR}$. Then, we have $G' \xleftarrow[\mathcal{UR}]{*} G\tau \xrightarrow[\mathcal{UR}\cup\{U=V\Rightarrow G\rightarrow D\}]{} D\tau \xrightarrow[\mathcal{UR}]{*} D'$. Since $\mathcal{UR}$ is complete, G' and D' are two constructor terms. On the other hand, the constructors are free in $\mathcal{R}$ and $G\tau \not\downarrow_{\mathcal{UR}} D\tau$. Hence, $G' \not\downarrow_{\mathcal{R}} D'$ and therefore, $\mathcal{R}$ is not ground confluent. □

Our inference system is also complete:

Theorem 12 (Completeness) *Let $\mathcal{R}$ be a reductive conditional rewriting system over free constructors. Assume an oracle for deciding (joinable-) inductive properties. If $\mathcal{R}$ is complete and ground confluent, then $\mathcal{CGC}$ will succeed.*

Proof: If $\mathcal{R}$ is complete and ground confluent, then after the application of the rule **Initialization** and the saturation of the application of the rules **Checking New Function, Decompose Variable** and **Checking Leaf** we obtain $\mathcal{MP} = \emptyset$ (otherwise, by Theorem 11, we conclude that $\mathcal{R}$ is not complete), $\mathcal{AP} = \emptyset$ and $\mathcal{UR} \models_{\mathcal{I}nd} (\mathcal{R} - \mathcal{UR})$ (otherwise, by Theorem 11, we conclude that $\mathcal{R}$ is not ground confluent). Hence **Success** applies. □

6 Specifications with Non-Free Constructors

Assume that $\mathcal{R}$ comes in two parts: $\mathcal{R} = \mathcal{R}_\mathcal{C} \cup \mathcal{R}_\mathcal{D}$ where $\mathcal{R}_\mathcal{C}$ is the set of axioms for constructors consisting of left-linear and non-conditional rewrite rules. To check completeness and ground confluence for $\mathcal{R}$, we start by computing the tree automaton recognizing the set of irreducible ground terms. From this automaton, we extract for each non-free sort s, a *cover sort* [5] for s, which describes a partition of the irreducible ground terms of sort s. After this step, we start the computation of pattern trees of defined functions. The only difference with our technique for non-free constructors is that the successors of any internal node labelled by the pattern $f(\overline{T})$ can also be obtained by covering the sort of values of an induction variable in $f(\overline{T})$.

Let $\mathcal{UR}$ be the set of rules in $\mathcal{R}$ whose lefthand sides match the leaves of the trees. To conclude that $\mathcal{R}$ is complete and ground confluent, we require that (i) all the leaves of the trees are strongly reducible and non-ambiguous, (ii) the rules in $\mathcal{R}_\mathcal{D} - \mathcal{UR}$ and all critical pairs between a rule in $\mathcal{R}_\mathcal{C}$ and a rule in $\mathcal{UR}$ are inductive theorems of $\mathcal{UR} \cup \mathcal{R}_\mathcal{C}$, and (iii) the axioms for constructors $\mathcal{R}_\mathcal{C}$ are ground confluent.

The number of critical pairs between a rule in $\mathcal{R}_C$ and a rule in $\mathcal{UR}$, is very small in practice. Moreover, assumption (iii) can be easily verified since the constructor rules are assumed to be unconditional.

However, we can avoid the computation of critical pairs as well as the test of ground confluence of $\mathcal{R}_C$, by transforming the specification with non-free constructors as in [8], into an order-sorted specification where every function symbol is either a free constructor or a completely defined function. Therefore, our procedure for checking completeness and ground confluence for specifications with free constructors can be applied.

7 Parameterized Specifications

Parameterization is very important for building up larger data types and software systems from generic specifications in a highly reusable way.

A *parameterized specification* is a pair $\mathcal{PS} = (\mathcal{P}, \mathcal{B})$ with $\mathcal{P} \subseteq \mathcal{B}$. We call $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{E}_\mathcal{P})$ the *parameter* specification, and $\mathcal{B} = (\mathcal{F}_\mathcal{B}, \mathcal{E}_\mathcal{B})$ the *body* specification, where $\mathcal{E}_\mathcal{P}$ is the set of parameter *constraints* consisting of equational clauses over $\mathcal{F}_\mathcal{P}$, and $\mathcal{E}_\mathcal{B}$ is the set of axioms of the parameterized specification. We assume that the axioms in $\mathcal{E}_\mathcal{B} - \mathcal{E}_\mathcal{P}$ are reductive rewrite rules over $\mathcal{F}_\mathcal{B}$.

We have extended our technique to check the completeness and the ground confluence for parameterized specifications in the case where no symbol from $\mathcal{F}_\mathcal{P}$ occurs on the lefthand side of any rewrite rule from $\mathcal{E}_\mathcal{B} - \mathcal{E}_\mathcal{P}$.

8 Implementation and Computer Experiments

We have implemented our new technique in the SPIKE system. The program is able to check both completeness and ground confluence for parameterized and non-parameterized conditional specifications with free constructors. The program starts by computing a pattern tree for every defined symbol, and identifies a set of rules that we must check for validity. If all the leaves are strongly reducible and non-ambiguous, and if the identified rules are inductively valid, then we conclude that the given specification is complete and ground confluent.

The root of a pattern tree is displayed first, and each level of the tree is indented to ease the reading. There are two kind of leaves: leaves which are strongly reducible by case (i), and leaves which are strongly reducible by case (ii) (see Definition 7). For the last case, we display for each leaf L: $R_1\sigma_1$ if $P_1\sigma_1, \ldots, R_n\sigma_n$ if $P_n\sigma_n$ where $\forall i \in [1..n]$: $P_i \Rightarrow L_i \to R_i \in \mathcal{R}$ and $L = L_i\sigma_i$. With each leaf comes a comment indicating whether it is strongly reducible and non-ambiguous.

```
Pattern tree of +:
 x1 + x2
   0 + x2 -Ok-
   s(x3) + x2 -Ok-

Pattern tree of *:
 x1 * x2
   0 * x2 -Ok-
   s(x3) * x2 -Ok-

Pattern tree of <:
 x1 < x2
   0 < x2
     0 < 0 -Ok-
     0 < s(x1) -Ok-
   s(x3) < x2
     s(x3) < 0 -Ok-
     s(x3) < s(x1) -Ok-

Pattern tree of even:
 even(x1)
   even(0) -Ok-
   even(s(x2))
     even(s(0)) -Ok-
     even(s(s(x1))) -Ok-

Pattern tree of odd:
 odd(x1)    -Ok-
   False if even(x1)=True
   True if even(s(x1))=True

All the leaves of the trees are strongly reducible
  and non-ambiguous.

The following rules are inductively valid w.r.t
  the remainder of the axioms:

x1+0=x1;
x1+s(x2)=s(x1+x2);
(x1+x2)+x3=x1+(x2+x3);
x1*0=0;
even(x1+x1)=True;
even(s(x1+x1))=False;
even(x1+(x2+x2))=even(x1);
odd(s(s(x1)))=odd(x1);
odd(x1+x1)=False;
odd(s(x1+x1))=True;
odd(x1)=True,odd(x2)=True=>odd(x1+x2)=False;
even(x1)=True,even(x2)=True=>even(x1+x2)=True;
odd(x1)=True,odd(x2)=False=>odd(x1+x2)=True;
even(x1)=True,even(x2)=False=>even(x1+x2)=False;
even(x1*s(s(x2)))=even(x1*x2);
even(x1)=True=>even(x1*x2)=True;
odd(x1)=True,odd(x2)=True=>even(x1*x2)= False

Then, all the axioms are complete and ground confluent.
```

Figure 3: Proving the Completeness and the Ground Confluence of Nat

$$
\begin{array}{l}
even(x_1 + (x_2 + (x_3 + (x_2 + x_3)))) = even(x_1) \\
even(x_1 + (x_2 + (x_3 + (x_4 + (x_2 + (x_3 + x_4)))))) = even(x_1) \\
\vdots \\
even(s(x_1)) = true \Rightarrow even(s((x_1 * s(s(x_2))) + x_2)) = false \\
even(s(x_1)) = true \Rightarrow even(x_1 * s(x_2 + (x_3 + (x_2 + x_3)))) = false \\
even(s(x_1)) = true \Rightarrow even((x_1 * s(x_2 + (x_3 + x_3))) + x_2) = false \\
\vdots
\end{array}
$$

Figure 4: Divergence of the Completion procedure

```
parameter specification: TOSET
sorts Elem, Bool;
functions:
true, false :               → Bool;
_<_         : Elem Elem     → Bool;
constraints:
true = false ⇒;
x < x = false;
x < y = true ∨ x < y = false;
x < y = false ∨ y < x = false;
x < y = false ∨ y < z = false ∨ x < z = true;
end
```

Figure 5: The parameter specification TOSET

```
specification: SORTING[T::TOSET]
sorts List;
constructors:
nil      :              → List;
cons__   : Elem List    → List;
defined functions:
insert__ : Elem List    → List;
isort_   : List         → List;
sorted_  : List         → Bool;
axioms:
insert(x,nil) → cons(x,nil);
x < y=true ⇒ insert(x,cons(y,z)) → cons(x,cons(y,z));
x < y=false ⇒ insert(x,cons(y,z)) → cons(y,insert(x,z));
isort(nil) → nil ;
isort(cons(x,l)) → insert(x,isort(l));
sorted(nil) → true;
sorted(cons(x,nil)) → true;
x < y=false ⇒ sorted(cons(x,cons(y,z))) → false;
x < y=true ⇒ sorted(cons(x,cons(y,z))) → sorted(cons(y,z));
sorted(insert(x,y)) → sorted(y);
sorted(isort(x)) → true;
end
```

Figure 6: The parameterized specification SORTING

Example 13 *Figure 3 shows the transcript of a session with* SPIKE *for proving the completeness and the ground confluence of the specification given in Figure 1. Using completion techniques, we obtain an infinite set of critical pairs (see Figure 4). This example cannot be checked by the methods of [14, 1], since they are designed for non-conditional specifications. To prove the ground confluence of $\mathcal{R}$ using the methods of [20, 2], we need to compute more than 120 critical pairs! In addition, the test of the validity of most critical pairs fails or it is very hard to achieve.*

For example, the critical pair [3] *between rules (7) and (27):*

$$(31)\ even(s(x_1) * x_2) \rightarrow even((x_1 * s^2(x_2)) + s^2(x_2))$$

is not valid w.r.t. the sufficient criterion for ground confluence given in [20]:

```
Pattern tree of insert:
 insert(x1,x2)
   insert(x1,Nil) -Ok-
   insert(x1,Cons(x3,x4))    -Ok-
     Cons(x1,Cons(x3,x4)) if x3 < x1=False
     Cons(x3,insert(x1,x4)) if x3 < x1=True

Pattern tree of isort:
 isort(x1)
   isort(Nil) -Ok-
   isort(Cons(x2,x3)) -Ok-

Pattern tree of sorted:
 sorted(x1)
   sorted(Nil) -Ok-
   sorted(Cons(x2,x3))
     sorted(Cons(x2,Nil)) -Ok-
     sorted(Cons(x2,Cons(x1,x4)))    -Ok-
       False if x1 < x2=True
       sorted(Cons(x1,x4)) if x1 < x2=False

All the leaves of the trees are strongly reducible
  and non-ambiguous.

The following rules are inductively valid w.r.t
  the remainder of the axioms:

sorted(insert(x1,x2)) = sorted(x2);
sorted(isort(x1)) = True

Then, all the axioms are complete and ground confluent.
```

Figure 7: Proving the Completeness and the Ground Confluence of SORTING

```
...

All the leaves of the trees are strongly reducible
  and non-ambiguous.

One of the following rules is not inductively valid
  w.r.t the remainder of the axioms:

x1<x2=True => sorted(Cons(x2,Cons(x1,Nil)))=True;
sorted(insert(x1,x2)) = sorted(x2);
sorted(isort(x1)) = True

Then, all the axioms are complete
  but not ground confluent.
```

Figure 8: Detecting a non-Ground Confluent Specification

Indeed, let us consider the instance of equation (31) by the test set substitution $\{x_1 \mapsto s^3(x),\ x_2 \mapsto s^3(y)\}$:

$$(32)\ even(s^4(x) * s^3(y)) = even((s^3(x) * s^5(y)) + s^5(y))$$

The term $even(s^4(x) * s^3(y))$ is simplified by the axiom (7) into $even((s^3(x) * s^3(y)) + s^3(y))$. We obtain:

$$(33)\ even((s^3(x) * s^3(y)) + s^3(y)) = even((s^3(x) * s^5(y)) + s^5(y))$$

(31) is not valid since (33) is not joinable by $\mathcal{R} \cup \{(31)\}$.

Example 14 *Figure 7 shows the transcript of a session with* SPIKE *for proving the completeness and the ground confluence of the parameterized specification given in Figures 5 and 6.*

Now let us add the rule

$$x < y = true \Rightarrow sorted(Cons(y, cons(x, nil))) = true$$

to the specification given in Figure 6. Then, the obtained specification is complete but not ground confluent (see Figure 8).

---

[3] In order to simplify notation we write $s^n(x)$ instead of $s(\ldots s(x))$

9 Conclusion

We have presented a procedure for simultaneously checking completeness and ground confluence for specifications with free/non-free constructors and parameterized specifications. As opposed to previous work for checking ground confluence, our procedure is complete and always terminates assuming an oracle for deciding (joinable-) inductive properties.

Our technique has been implemented in the prover SPIKE. Computer experiments show that our method is more powerful and practical than related approaches. Indeed, as shown in Example 13, our procedure allows us to check the ground confluence of specifications where the classical completion techniques generate an infinite set of critical pairs. Moreover, our proof of the ground confluence of the specification given in Figure 1 is completely automatic and does not require the computation of critical pairs. However, the methods of [20, 2] need to compute more than 120 critical pairs! In addition, the test of the validity of most critical pairs fails or is very hard to achieve.

We plan to extend our technique for membership equational theories [6] as well as Associative/Commutative (AC) theories. The extension of related approaches for AC theories can be very inefficient since they are based on the computation of critical pairs, and therefore they need to compute the minimal complete set of AC-unifiers that has a double-exponential cardinality in the worst case [19]. Unlike these approaches, our method should be also efficient for AC theories since it does not use unification, but matching only.

References

[1] K. Becker. Proving ground confluence and inductive validity in constructor based equational specifications. In Proc. *TAPSOFT '93*, volume 668 of *Lecture Notes in Computer Science*, pages 46–60. Springer-Verlag, 1993.

[2] K. Becker. How to Prove Ground Confluence. SEKI-report SR-96-02, Universität Kaiserslautern, 1996.

[3] A. Bouhoula. Using induction and rewriting to verify and complete parameterized specifications. *Theoretical Computer Science*, 170(1-2):245–276, 1996.

[4] A. Bouhoula. Automated theorem proving by test set induction. *Journal of Symbolic Computation*, 23(1):47–77, 1997.

[5] A. Bouhoula and J.-P. Jouannaud. Automata-driven automated induction. In Proc. *12th IEEE Symp. Logic in Computer Science, Warsaw (Poland)*, pages 14–25. IEEE Comp. Soc. Press, 1997.

[6] A. Bouhoula, J.-P. Jouannaud, and J. Meseguer. Specification and proof in membership equational logic. *Theoretical Computer Science*, 236(1-2):35–132, 2000.

[7] A. Bouhoula and M. Rusinowitch. Implicit induction in conditional theories. *Journal of Automated Reasoning*, 14(2):189–235, 1995.

[8] H. Comon. Inductive proofs by specifications transformations. In Proc. *3rd Conference on Rewriting Techniques and Applications, Chapel Hill (N.C., USA)*, volume 355 of *Lecture Notes in Computer Science*, pages 76–91. Springer-Verlag, April 1989.

[9] N. Dershowitz. Corrigendum to termination of rewriting. *Journal of Symbolic Computation*, 4:409–410, 1987.

[10] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In the *Handbook of Theoretical Computer Science*. Elsevier Science Publishers B. V. (North-Holland), 1990.

[11] N. Dershowitz, M. Okada, and G. Sivakumar. Confluence of conditional rewrite systems. In Proc. *1st International Workshop on Conditional Term Rewriting Systems, Orsay (France)*, volume 308 of *Lecture Notes in Computer Science*, pages 31–44. Springer-Verlag, July 1987.

[12] L. Fribourg. A strong restriction of the inductive completion procedure. *Journal of Symbolic Computation*, 8(3):253–276, 1989.

[13] H. Ganzinger. Ground term confluence in parametric conditional equational specifications. In Proc. *STACS 87*, volume 247 of *Lecture Notes in Computer Science*, pages 286–298. Springer-Verlag, 1987.

[14] R. Göbel. Ground confluence. In Proc. *2nd Conference on Rewriting Techniques and Applications, Bordeaux (France)*, volume 256 of *Lecture Notes in Computer Science*, pages 156–167. Springer-Verlag, 1987.

[15] G. Huet and J.-M. Hullot. Proofs by induction in equational theories with constructors. *Journal of Computer and System Sciences*, 25(2):239–266, 1982.

[16] J.-P. Jouannaud and E. Kounalis. Automatic proofs by induction in theories without constructors. *Information and Computation*, 82:1–33, 1989.

[17] S. Kaplan. Simplifying conditional term rewriting systems: Unification, termination and confluence. *Journal of Symbolic Computation*, 4(3):295–334, 1987.

[18] D. Kapur, P. Narendan, and F. Otto. On ground confluence of term rewriting systems. *Information and Computation*, May 1990.

[19] D. Kapur and P. Narendran. Double-exponential complexity of computing a complete set of AC-unifiers. In Proc. *LICS'92*, Santa-Cruz (California, USA), June 1992.

[20] E. Kounalis and M. Rusinowitch. Studies on the ground convergence property of conditional theories. In Proc. *AMAST, Iowa-City*, Workshop in Computing. Springer-Verlag, 1991.

[21] P. Padawitz. The equational theory of parameterized specifications. *Information and Computation*, 76:121–137, 1988.

[22] D. A. Plaisted. Semantic confluence tests and completion methods. *Information and Control*, 65(2/3):182–215, 1985.

[23] M. Wirsing. Algebraic specification. In the *Handbook of Theoretical Computer Science*, volume B, chapter 13. Elsevier Science Publishers B. V. (North-Holland) and The MIT press, 1990.

Theory of Judgments and Derivations

Masahiko Sato
Graduate School of Informatics
Kyoto University
Kyoto, Japan
masahiko@kuis.kyoto-u.ac.jp

Abstract— **We propose a computational and logical framework NF (Natural Framework) which is suitable for presenting mathematics formally. Our framework is an extendable framework since it is *open-ended* both computationally and logically in the sense of Martin-Löf's theory or types. NF is developed in three steps. Firstly, we introduce a theory of expressions and schemata which is used to provide a universe for representing mathematical objects, in particular, judgments and derivations as well as other usual mathematical objects. Secondly, we develop a theory of judgments within the syntactic universe of expressions. Finally, we introduce the notions of derivation and derivation game and will show that we can develop mathematics as derivation games by regarding mathematics as an open-ended process of defining new concepts and deriving new judgments. Our theory is inspired by Martin-Löf's theory of expressions and Edinburgh LF, but conceptually much simpler. Our theory is also influenced by Gentzen's natural deduction systems.**

Keywords— **Logical Framework, Natural Deduction, Martin-Löf's philosophy of mathematics.**

I. INTRODUCTION

The continuous development of mathematics has been supported by the introduction of new *concepts*. For example, the concept *number* has been extended starting from *natural number*, through *integer*, *rational number*, *real number* up to the concept *complex number*. For each object s and a concept P we can form a *judgment* $s : P$ which asserts that the object s falls under the concept P.[1] As is customary in modern mathematics, we can assert a judgment when and only when we could derive the judgment, namely, when we could construct a *derivation* whose conclusion is the judgment. Here the derivation provides an *evidence* for the judgment. Firstly, the derivation makes the judgment evident to the very person who constructed the derivation because, supposedly, he has just constructed it by applying *derivation rules* correctly. Secondly, the derivation makes the judgment evident to any reader of the derivation, because[2] the reader can read and check the *correctness* of the derivation with respect to a given set of rules and thereby he can *experience* the process of constructing the derivation for himself.

In order to make the above analysis valid, it is necessary that (1) the derivation is written in a common language shared by the writer and reader of the derivation and (2) a derivation rule in the derivation sends evident judgments to an evident judgment. In this paper we are mostly interested in the first condition of the above sentence, and we will provide a general and uniform framework NF (Natural Framework) for defining various formal systems conveniently. We will fulfill the first condition by realizing Kreisel's dictum (see [1], [4], [6], [7], [8], [15]) which asks the (primitive) recursive decidability of whether or not a given derivation is a correct derivation of a given judgment. The point is that the *correctness* of a derivation can be checked mechanically against a given set of formal rules while the *evidence* of the judgment proven by the derivation is determined semantically. We will achieve the decidability by defining the possible forms of *judgments* and *derivations* syntactically without referring to the meanings of judgments. Thus our method is along the line of Hilbert's program (see, e.g. [5], [7], [12], [14]), according to which formalistic notions of formulas and theorems must be defined and treated in a *finitistic* manner.

There are already numerous attempts to provide general frameworks for defining formal systems. Among these works, our work is greatly influenced by Martin-Löf's theory of expressions [11] and the Edinburgh Logical Framework [4].[3] Martin-Löf developed a theory of expressions which is used as an underlying language for representing his intuitionistic theory of types. Edinburgh LF was inspired by Martin-Löf's theory of expressions. It is a dependently typed λ-calculus and it can encode various logical systems by using the judgments-as-types principle. We will compare our work with these works in detail later. Here, we only mention that we have designed our framework

---

[1] See, e.g., Bell [2, page 9] and Frege [3].

[2] Here, we assume that the reader also accepts the rules as evident. So, if the reader is an intuitionist and the derivation contained an application of the law of the excluded middle for an undecidable proposition, then the derivation is not evident to the reader. But, even in such a case, the reader can check whether or not the derivation is a *correct* derivation obeying the rules of classical mathematics.

[3] Throughout this paper, the Edinburgh Logical Framework will be called Edinburgh LF or, simply, LF.

NF so that it will become much simpler than these frameworks. We think the simplicity of the framework is crucial since we wish to provide a foundational framework for defining various formal systems on top of it.

Martin-Löf [9], [10] classified judgments into *categorical judgments* which are judgments made without hypotheses and *hypothetical judgments* which are judgments made under several hypotheses. We will write $H \Rightarrow J$ for the hypothetical judgment which asserts the judgment J under the hypothesis H. In Martin-Löf's notation, this judgment is written as $J\ (H)$. Thus, the hypothetical judgment $J\ (H_1, \ldots, H_n)$ in Martin-Löf's notation can be written as $H_1 \Rightarrow \cdots \Rightarrow H_n \Rightarrow J$ in our notation. A typical hypothetical judgment takes the form $J\ (x : A)$ where x is a variable and it means that for any x, if $x : A$ holds, then J also holds. Therefore, the variable x is universally quantified in the hypothetical judgment. To make this quantification explicit, we introduce judgments of the form $(x)\,[J]$ and call such judgments *universal judgments.* Using universal judgments, Martin-Löf's hypothetical judgment $J\ (x : A)$ can be written as $(x)\,[x : A \Rightarrow J]$[4] which we will also abbreviate as $(x : A)\,[J]$.

In order to represent judgments and derivations formally, we first introduce a set E of symbolic expressions. The set E contains two infinite denumerable sets of variables and constants, and it is closed under the operation of forming a *pair* $\mathsf{<}e \,|\, f\mathsf{>}$ of two expressions e and f, and the operation of forming an *abstract* $(x)\,[e]$ which is obtained by binding a specified variable x in a given expression e. The set E is rich enough to represent judgments naturally. For instance, the universal judgment $(x)\,[x : A \Rightarrow J]$ can be written as the expression

$$(x)\,[\mathsf{<}\Rightarrow, \mathsf{<}:, x, A\mathsf{>}, J\mathsf{>}]$$

where '$\Rightarrow$' and ':' are constants, and $\mathsf{<}:, x, A\mathsf{>}$, for example, is a list which we can construct by the pairing operation in a usual manner. We will see that derivations can be represented naturally by expressions in E.

Formal systems are commonly presented by specifying derivation rule-schemata whose instances may be used to construct a new derivation D from already constructed derivations $D_1, \ldots, D_n$. Such a rule-schema can be displayed as a figure of the form:

$$\frac{H_1 \quad \cdots \quad H_n}{J}\ N(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$$

where N is the name of the rule schema (which is a constant), $J, H_1, \ldots, H_n$ are judgment-schemata whose free variables stand for unspecified expressions, and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ is a sequence of distinct variables containing all the free variables in $J, H_1, \ldots, H_n$. For example, we can define a formal system which defines the set of natural numbers by the two rule-schemata below.

$$\frac{}{\mathtt{0 : Nat}}\ \mathtt{zero()} \qquad \frac{\boldsymbol{n}\mathtt{ : Nat}}{\mathtt{s(}\boldsymbol{n}\mathtt{) : Nat}}\ \mathtt{succ(}\boldsymbol{n}\mathtt{)}$$

We can represent this formal system by the following expression Nat.

$$\begin{aligned} \mathsf{Nat} &:\equiv \mathsf{<zero, succ>}, \\ \mathsf{zero} &:\equiv \mathsf{<}\mathtt{0 : Nat}, \mathsf{<zero>}, \mathsf{<>}\mathsf{>}, \\ \mathsf{succ} &:\equiv \mathsf{<}\mathtt{s(}\boldsymbol{n}\mathtt{) : Nat}, \mathsf{<succ}, \boldsymbol{n}\mathsf{>}, \mathsf{<}\boldsymbol{n}\mathtt{ : Nat}\mathsf{>>}. \end{aligned}$$

In the formal system Nat we can construct the following derivation which shows that 2, which is represented by $\mathtt{s(s(0))}$, is a natural number.

$$\cfrac{\cfrac{\cfrac{}{\mathtt{0 : Nat}}\ \mathtt{zero()}}{\mathtt{s(0) : Nat}}\ \mathtt{succ(0)}}{\mathtt{s(s(0)) : Nat}}\ \mathtt{succ(s(0))}$$

The rule-schemata for Nat are *elementary* in the sense that all the premises and conclusions are categorical judgments. Such elementary rule-schemata are often sufficient for presenting syntax and operational semantics of *computational* systems. We will need non-elementary rule-schemata to present *logical* systems in natural deduction style. Consider, for example, the following rule of mathematical induction:

$$\frac{P[0] \quad \begin{matrix}(P[x]) \\ P[\mathtt{s}(x)]\end{matrix}}{\forall x.P[x]}\ \mathtt{ind}$$

When this rule is applied, the specified occurrences of the assumption $P[x]$ is discharged, and the rule may be applied only if the variable x does not occur free in any assumptions, other than $P[x]$, on which $P[\mathtt{s}(x)]$ depends. In Edinburgh LF, this rule can be defined by declaring the constant IND as follows:

$$\begin{aligned} \text{IND} \quad : \quad & \Pi\phi : \iota \to o.\ \mathtt{True}(\phi 0) \to \\ & \quad (\Pi x : \iota.\mathtt{True}(\phi x) \to \mathtt{True}(\phi(\mathtt{s}(x))) \to \\ & \quad \mathtt{True}(\forall(\lambda x : \iota.\phi x))). \end{aligned}$$

Thus, in LF, the induction rule is represented by a higher-order function that accepts proofs of the two premises $\mathtt{True}(\phi 0)$ and $\Pi x : \iota.\mathtt{True}(\phi x) \to \mathtt{True}(\phi(\mathtt{s}(x)))$, and returns a proof of the conclusion $\mathtt{True}(\forall(\lambda x : \iota.\phi x))$. The second premise is an LF encoding of a general judgment whose body is a hypothetical judgment.

In NF, the induction rule is given by the following rule-schema:

$$\frac{\boldsymbol{P}[\![\mathtt{0}]\!] \mathtt{ : True} \quad (x\mathtt{ : Nat)}\,[\boldsymbol{P}[\![x]\!] \mathtt{ : True} \Rightarrow \boldsymbol{P}[\![\mathtt{s}(x)]\!] \mathtt{ : True}]}{\forall(x\mathtt{ : Nat})\,[\boldsymbol{P}[\![x]\!]] \mathtt{ : True}}\ \mathtt{ind}(\boldsymbol{P})$$

---

[4] In [10], Martin-Löf introduces universal judgments, which he calls general judgments. Edinburgh LF can represent general judgments as dependent types.

In this rule-schema, $\boldsymbol{P}$ is a *schematic parameter* of the rule, and when we use the rule-schema, we must *instantiate* the rule-schema by assigning an abstract expression $e \in \mathsf{E}$ to $\boldsymbol{P}$. So, if we wish to prove, for instance, the judgment

$$\forall(x : \mathtt{Nat})\, \mathtt{[}x + x = 2 \cdot x\mathtt{]} : \mathtt{True}$$

by applying the `ind`-rule, then we must assign the abstract $(x)\,\mathtt{[}x + x = 2 \cdot x\mathtt{]}$ to $\boldsymbol{P}$. In NF, in order to express such rule-schemata formally, we extend the set E of expressions to the set S of *schemata* by adding *schematic variables* which will be used as parameters of rule-schemata and *application schemata* which are of the form $\boldsymbol{X}\,[\![S]\!]$ where $\boldsymbol{X}$ is a schematic variable and S is a schema. Although elementary rule-schemata can be represented by expressions in E we use schematic variables so that rule-schemata can be presented uniformly as schemata.

In addition to the mechanism of instantiating rule-schemata, we need a general mechanism to generate valid *derivations* of judgments under a given set of rule-schemata. We will introduce the notion of *derivation game*[5] which provides such a mechanism. A derivation game G is simply a finite list of rule-schemata and hence it is an element in S. Relative to a derivation game G, the set of derivations valid in G is determined as a recursively enumerable subset of E and it is always decidable whether a given expression is a valid derivation in G or not. We can use derivation games to define various formal systems in a completely formal way.

The paper is organized as follows. In §II, we introduce a theory of expressions and schemata. In §III, we introduce three forms of judgments, namely, categorical, hypothetical and universal judgments syntactically. The set of judgments becomes a primitive recursive subset of E. We then define derivation games, which can be used to assign meanings to these judgments. In this way, one and the same judgment can have different meanings in different derivation games. NF is neutral semantically in this sense, and can define both intuitionistic and classical mathematics equally well. All the definitions and proofs in §§II–III will be given in a finitistic fashion. Due to lack of space, we cannot give concrete examples and comparison with related works. For these, refer to the full paper which is available at:

`http://www.sato.kuis.kyoto-u/~masahiko/index-e.html`

The full paper will appear in a volume of the Lecture Notes in Aritificial Intelligence, published by the Springer Verlag.

II. THEORY OF EXPRESSIONS AND SCHEMATA

We define the set E of *expressions* as a structure which is equipped with the following two operations. The first operation is *pair formation.* If e and f are expressions then we can form an expression $\mathtt{<}e \,|\, f\mathtt{>}$ which is the pair of e and f. The second operation is *abstraction.* This operation abstracts away a specified variable x from a given expression e yielding an abstract of the form $(x)\,\mathtt{[}e\mathtt{]}$.

We assume that we have a countably infinite set of atomic symbols which consists of two mutually disjoint infinite sets of *variables* and *constants.* Variables will also be called *object variables* to distinguish them from other kinds of variables we introduce later in this section. Furthermore, we assume that the set of object variables is a union of two infinite and mutually disjoint sets of *general variables* and *derivation variables.* Object variables will be designated by letters x, y, z and derivation variables will be designated by X, Y, Z, and constants will be designated by the letter c. We will also assume that strings of characters in `typewriter` font are included among constants and they are distinct if they are spelled differently.

Definition 1 (Context) A sequence

$$x_1, \ldots, x_n$$

of distinct object variables is called a *context.*
We will say that a variable x is *declared* in a context Γ if x occurs in Γ.

Definition 2 (Expression) We define *expressions* relative to a context Γ as follows.

1. *Object variable.* If Γ is a context and x is declared in Γ, then x is an expression under Γ.
2. *Constant.* If Γ is a context and c is a constant, then c is an expression under Γ.
3. *Pair.* If Γ is a context and e, f are expressions under Γ, then $\mathtt{<}e \,|\, f\mathtt{>}$ is an expression under Γ.
4. *Abstract.* If Γ, x is a context and e is an expression under Γ, x, then $(x)\,\mathtt{[}e\mathtt{]}$ is an expression under Γ

In the 4th clause of the above definition, the variable x which was (globally) declared in Γ, x becomes locally declared in the expression $(x)\,\mathtt{[}e\mathtt{]}$. The syntax $(x)\,\mathtt{[}e\mathtt{]}$ may look heavy compared to more conventional notations like $x.\ e$ or $(x)e$. However, we prefer this notation, since it clearly shows that the variable x in (x) is a binding occurrence and the scope of the binder x is e.

We will identify two expressions e and f if they are defined in the same way except for the choices of variables, and we will write $e \equiv f$ in this case. Then, we can define the capture avoiding substitution operation on expressions as usual, and we will write $[x := d](e)$ for the result of substituting d for x in e. For an expression e, we can define the set $\mathrm{FV}(e)$ of *free variables*

---

[5] We use the term 'game' to reflect the formalistic view of mathematics according to which mathematical theories are presented by spcifying the *rules* of the games (i.e., theories) which determine the derivable judgments of the theories.

in e as usual and can verify that if e is an expression under Γ, then $\mathrm{FV}(e) \subseteq \Gamma$. An expression is *closed* if $\mathrm{FV}(e) = \emptyset$ or, equivalently, if it is an expression under the empty context.

As a notational convention, we will write

$$\texttt{<}e_1, e_2, \ldots, e_n\texttt{>} \ \text{ for } \ \texttt{<}e_1\,\texttt{|}\,\texttt{<}e_2\,\texttt{|}\cdots\texttt{<}e_n\,\texttt{|nil>}\cdots\texttt{>>}.$$

Such an expression is called a *list* and we will define concatenation of two lists by:

$$\texttt{<}e_1, \ldots, e_m\texttt{>}\oplus\texttt{<}f_1, \ldots, f_n\texttt{>} :\equiv \texttt{<}e_1, \ldots, e_m, f_1, \ldots, f_n\texttt{>}.$$

We will extend this notational convention to schemata we define below. We will also identify a context $x_1, \ldots, x_n$ with the list $\texttt{<}x_1, \ldots, x_n\texttt{>}$.

We now generalize the notion of an expression to that of a *schema*. We will write S for the set of schemata. We first extend the atomic symbols by adding a countably infinite set V_s of *schematic variables*. Schematic variables are classified into two mutually disjoint subsets of *expression variables*[6] and *abstract variables*[7]. Schematic variables will be written by bold italic letters such as $\boldsymbol{z}$, $\boldsymbol{X}$ etc.

Definition 3 (Schema) We define a *schema* relative to a context where, as before, a context is a sequence of distinct object variables.

1. *Expression variable.* If Γ is a context and $\boldsymbol{z}$ is an expression variable, then $\boldsymbol{z}$ is a schema under Γ.
2. *Object variable.* If Γ is a context and x is declared in Γ, then x is a schema under Γ.
3. *Constant.* If Γ is a context and c is a constant, then c is a schema under Γ.
4. *Pair.* If S and T are schemata under Γ, then $\texttt{<}S\,\texttt{|}\,T\texttt{>}$ is a schema under Γ.
5. *Abstract.* If S is a schema under $\Gamma \oplus \texttt{<}x\texttt{>}$, then $(x)\,\texttt{[}S\texttt{]}$ is a schema under Γ
6. *Application.* If $\boldsymbol{X}$ is an abstract variable and S is a schema under Γ, then $\boldsymbol{X}\,\llbracket S\rrbracket$ is a schema under Γ.

Any expression is a schema because of clauses 2 – 5. Here again we will identify two schemata S and T if they are defined in the same way except for the choices of binding variables in clause 5. In this case, we will write $S \equiv T$ and say that S and T are *definitionally equal.* It is clear that if $S \equiv T$ and S is an expression, then T is also an expression. A schema is *closed* if it is a schema under the empty environment. A closed schema may contain schematic variables but may not contain free object variables. Schematic variables will be used as *parameters* of rule-schemata and there is no need to abstract over schematic variables. This is the reason why we do not have a clause to bind schematic variables in the above definition. We now define a very important operation of *instantiation.*

Definition 4 (Environment and Instance) A function

$$\rho : \mathsf{V}_s \to \mathsf{E}$$

is an *environment* if $\rho(\boldsymbol{X})$ is an abstract for any abstract variable $\boldsymbol{X}$ and $\rho(\boldsymbol{z}) \equiv (x)\,\texttt{[}x\texttt{]}$ for all but finitely many schematic variables $\boldsymbol{z}$. An environment ρ can be extended to a function $\llbracket\ \cdot\ \rrbracket_\rho : \mathsf{S} \to \mathsf{E}$ as follows.

1. *Expression variable.* $\llbracket \boldsymbol{z}\rrbracket_\rho :\equiv \rho(\boldsymbol{z})$.
2. *Object variable.* $\llbracket x\rrbracket_\rho :\equiv x$.
3. *Constant.* $\llbracket c\rrbracket_\rho :\equiv c$.
4. *Pair.* $\llbracket \texttt{<}S\,\texttt{|}\,T\texttt{>}\rrbracket_\rho :\equiv \texttt{<}\llbracket S\rrbracket_\rho\,\texttt{|}\,\llbracket T\rrbracket_\rho\texttt{>}$.
5. *Abstract.* $\llbracket (x)\,\texttt{[}S\texttt{]}\rrbracket_\rho :\equiv (x)\,\texttt{[}\llbracket S\rrbracket_\rho\texttt{]}$. (We assume, without loss of generality, that for any $\boldsymbol{z}$ occurring in S, x is not free in $\rho(\boldsymbol{z})$.)
6. *Application.* $\llbracket \boldsymbol{X}\,\llbracket S\rrbracket\rrbracket_\rho \equiv [x := \llbracket S\rrbracket_\rho](e)$, where $\rho(\boldsymbol{X}) \equiv (x)\,\texttt{[}e\texttt{]}$.

The expression $\llbracket S\rrbracket_\rho$ is called the *instantiation of S by ρ* or an *instance of S.*
We see from clause 6 that an *application* $\boldsymbol{X}\,\llbracket S\rrbracket$ is instantiated by a call-by-value manner.

III. JUDGMENTS AND DERIVATION GAMES

Based on the theory of expressions and schemata, we can describe the syntax of *judgment* as follows. In this way, we can characterize the possible *form* a judgment can take. A *general variable context* is a context such that all the variables declared in it are general variables. In the following definition, Γ stands for an arbitrary general variable context.

Definition 5 (Judgment) *Judgments* are defined relative to general variable contexts and thereby classified into the following three types.

1. *Categorical Judgment.* If s and P are expressions under Γ, then
$$s : P$$
is a judgment under Γ.[8] In this case, s (P, resp.) is called the *object part* (*concept part*, resp.) of the categorical judgment.
2. *Hypothetical Judgment.* If H and J are judgments under Γ, then
$$H \Rightarrow J$$
is a judgment under Γ.[9]
3. *Universal Judgment.* If x is a general variable not declared in Γ, and J is an expression under $\Gamma \oplus \texttt{<}x\texttt{>}$, then
$$(x)\,\texttt{[}J\texttt{]}$$

---

[6] So called since they can be instantiated to arbitrary expressions.

[7] So called since they can be instantiated to arbitrary abstracts.

[8] This is an abbreviation of the expression: $\texttt{<:}, s, P\texttt{>}$.

[9] This is an abbreviation of the expression: $\texttt{<}\Rightarrow, H, J\texttt{>}$.

is a judgment under Γ.

One can verify that if J is a judgment under a general variable context Γ, then J is an expression under Γ. We can explain the meanings of judgments informally as follows. A categorical judgment $s : P$ means that the object s falls under the *concept* P. When P is a constant, it is possible to regard P as a *name* of the concept which is characterized by those s for which the judgment $s : P$ holds. For example, if `Nat` is a name given to the concept of the natural number, then `0 : Nat` means that `0` is a natural number. A hypothetical judgment $H \Rightarrow J$ means that we can derive the judgment J whenever H is derivable. A universal judgment of the form $(x)\,[J]$ means that we can derive the judgment $[x := e](J)$ for any expression e. Since we will often consider judgments of the form $(x)\,[x : P \Rightarrow J]$, we will abbreviate this form by $(x : P)\,[J]$.

Given a context Γ, we put J_Γ to be the set of judgments under Γ. It is easy to see that J_Γ is a primitive recursive subset of E.

Based on the above informal explanation of the intended meanings of judgments, we introduce the notion of derivation game.

Definition 6 (Rule-Schema) A closed schema of the form:

$$\text{<RS}, J, \text{<}c, z_1, \ldots, z_m\text{>}, \text{<}H_1, \ldots, H_n\text{>>}$$

is a *rule-schema* if c is a constant (called the *name* of the rule-schema) and $z_1, \ldots, z_m$ is a list (called the *parameter list* of the rule-schema) of distinct schematic variables exactly covering all the schematic variables in the schemata $J, H_1, \ldots, H_n$.

A rule-schema can be displayed as a figure of the following form:

$$\frac{H_1 \quad \cdots \quad H_n}{J}\ R(z_1, \ldots, z_m)$$

We will also use the following Prolog-like notation for the same rule-schema:

$$J \ \text{:-}_{R(z_1, \ldots, z_n)}\ H_1, \ldots, H_n.$$

We will often omit the rule name and its parameter list in these notations; or we keep the rule name only and omit the parameter list; or we keep the rule name and the list of abstract variables and omit the expression variables from the parameter list. If $J, H_1, \ldots, H_n$ are all of the form $s : c$ where $s \in \mathsf{S}$ and c is a constant, then the rule-schema is said to be an *elementary rule schema.*

Definition 7 (Derivation Game) A *derivation game* is a schema of the form $\text{<}R_1, \ldots, R_n\text{>}$ where each R_i is a rule-schema.

A derivation game is a closed schema since a rule-schema is closed. We write G for the set of derivation games. The set G is a primitive recursive subset of S. A derivation game is *elementary* if each rule-schema of the game is elementary.

A derivation game G determines a set $\mathsf{D}(G)$ of expressions called the set of G-*derivations*, which is the set of derivations derivable in G. To define $\mathsf{D}(G)$, we must first define the notion of a derivation context.

Definition 8 (Derivation Context) We define a *derivation context* Γ together with its *general variable part* $\mathrm{GV}(\Gamma)$ which is a context in the sense of Definition 1.

1. *Empty context.* The empty list <> is a derivation context and its general variable part is <>.
2. *General variable declaration.* If Γ is a derivation context, and x is a general variable not declared in Γ, then $\Gamma \oplus \text{<}x\text{>}$ is a derivation context and its general variable part is $\mathrm{GV}(\Gamma) \oplus \text{<}x\text{>}$.
3. *Derivation variable declaration.* If Γ is a derivation context, J is a judgment under $\mathrm{GV}(\Gamma)$, and X is a derivation variable not declared in Γ, then $\Gamma \oplus \text{<}X :: J^{10}\text{>}$ is a derivation context and its general variable part is $\mathrm{GV}(\Gamma)$.

Let Γ be a derivation context. An environment ρ is a Γ-*environment* if $\rho(z)$ is an expression under $\mathrm{GV}(\Gamma)$ for any schematic variable z.

Definition 9 (G-derivation) We define a G-*derivation* (also called a *derivation in* G) relative to a derivation context Γ as follows. We define its *conclusion* at the same time. In the following definition, Γ stands for an arbitrary derivation context. We can see from the definition below, that if D is a G-derivation under Γ, then its conclusion is an expression under $\mathrm{GV}(\Gamma)$.

1. *Derivation variable.* If X is a derivation variable and $X :: H$ is in Γ, then

   $$X$$

   is a G-derivation under Γ and its conclusion is H.
2. *Composition.* If $D_1, \ldots, D_n$ are G-derivations under Γ such that their conclusions are $H_1, \ldots, H_n$, respectively, and

   $$\frac{H_1 \quad \cdots \quad H_n}{J}\ R(e_1, \ldots, e_m)$$

   is an instantiation of a rule-schema in G by a Γ-environment and J is a categorical judgment under $\mathrm{GV}(\Gamma)$, then

   $$\frac{D_1 \quad \cdots \quad D_n}{J}\ R(e_1, \ldots, e_m)^{11}$$

   is a G-derivation and its conclusion is J.

[10] $X :: J$ is an abbreviation of $\text{<::}, X, J\text{>}$.

3. *Hypothetical derivation.* If D is a G-derivation under $\Gamma \oplus \texttt{<}X :: H\texttt{>}$ and its conclusion is J, then

   $$(X :: H)\,\texttt{[}D\texttt{]}$$ [12]

   is a G-derivation under Γ and its conclusion is $H \Rightarrow J$[13] .
4. *Universal derivation.* If D is a G-derivation under $\Gamma \oplus \texttt{<}x\texttt{>}$ and its conclusion is J, then

   $$(x)\,\texttt{[}D\texttt{]}$$

   is a G-derivation under Γ and its conclusion is $(x)\,\texttt{[}J\texttt{]}$.

We will write

$$\Gamma \vdash_G D :: J$$

if D is a derivation in G under Γ whose conclusion is J. For example, for the game Nat we gave in §I and for the derivation D given there, we have

$$\vdash_{\mathsf{Nat}} D :: \texttt{s(s(0))} : \texttt{Nat}.$$

NF provides another notation which is conveniently used to input and display derivations on a computer terminal. In this notation, instead of writing $\Gamma \vdash_G D :: J$ we write:

$$\Gamma \vdash J \ \texttt{in}\ G\ \texttt{since}\ D.$$

Also, when writing derivations in this notation, a derivation of the form

$$\frac{D_1 \quad \cdots \quad D_n}{J}\ R(e_1, \ldots, e_m)$$

will be written as:

$$J\ \texttt{by}\ R(e_1, \ldots, e_m)\ \texttt{\{}D_1\texttt{;}\ \ldots\ \texttt{;}D_n\texttt{\}}$$

Here is a complete derivation in Nat in this notation.

```
⊢ (x)[x:Nat ⇒ s(s(x)):Nat] in Nat
since
(x)[(X::x:Nat)[
    s(s(x)):Nat by succ(s(x)) {
        s(x):Nat by succ(x) {X}
    }
]]
```

The conclusion of the above derivation asserts that for any expression x, if x is a natural number, then so is $\texttt{s(s(}x\texttt{))}$, and the derivation shows us how to actually construct a derivation of $\texttt{s(s(}x\texttt{)):Nat}$ given a derivation X of $x\texttt{:Nat}$.

There are certain syntactic structures that are common among G-derivations for all derivation games G. To capture such structures, we introduce the notion of derivation.

Definition 10 (Derivation) We define a *derivation* and its *conclusion* relative to a derivation context as follows. In the following definition, Γ stands for an arbitrary derivation context.

1. *Derivation variable.* If X is a derivation variable and $X :: H$ is in Γ, then

   $$X$$

   is a derivation under Γ and its conclusion is H.
2. *Composition.* If $D_1, \ldots, D_n$ are derivations under Γ, R is a constant, $e_1, \ldots, e_m$ are expressions under Γ, and J is a categorical judgment under $\mathrm{GV}(\Gamma)$, then

   $$\frac{D_1 \quad \cdots \quad D_n}{J}\ R(e_1, \ldots, e_m)$$

   is a derivation under Γ and its conclusion is J.
3. *Hypothetical derivation.* If D is a derivation under $\Gamma \oplus \texttt{<}X :: H\texttt{>}$ and its conclusion is J, then

   $$(X :: H)\,\texttt{[}D\texttt{]}$$

   is a derivation under Γ and its conclusion is $H \Rightarrow J$.
4. *Universal derivation.* If D is a G-derivation under $\Gamma \oplus \texttt{<}x\texttt{>}$ and its conclusion is J, then

   $$(x)\,\texttt{[}D\texttt{]}$$

   is a derivation under Γ and its conclusion is $(x)\,\texttt{[}J\texttt{]}$.

We will write

$$\Gamma \vdash D :: J$$

if Γ is a derivation context and D is a derivation under Γ whose conclusion is J. We have the following theorem which characterizes derivations in terms of G-derivations.

Theorem 1 (Characterization of Derivations) For any expressions Γ, D and J, we have $\Gamma \vdash D :: J$ if and only if $\Gamma \vdash_G D :: J$ for some derivation game G.

Proof: Since *if-part* is trivial, we only show the *only-if-part* by induction on the construction of D. The only non-trivial case is the case where D is a derivation under Γ of the form:

$$\frac{D_1 \quad \cdots \quad D_n}{J}\ R(e_1, \ldots, e_m)$$

We let J_i be the conclusion of D_i for all i such that $1 \le i \le n$. In this case, we may assume by induction

---

[12] This is an abbreviation of the expression: $\texttt{<HD,}\,H\texttt{,}\,(X)\,\texttt{[}D\texttt{]>}$.

[13] We sometimes write the hypothetical judgment $H \Rightarrow J$ also by $(X :: H)\,\texttt{[}J\texttt{]}$ to emphasize the intended meaning of the judgment; namely, we can derive J if we have a derivation X of H. This notation also matches well with the notation for the hypothetical derivation we introduced here.

hypothesis, that we have derivation games G_i $(1 \leq i \leq n)$ such that $\Gamma \vdash_{G_i} D_i :: J_i$ $(1 \leq i \leq n)$. Then, we can obtain the derivation game with the required property by putting:

$$G :\equiv G_1 \oplus \cdots \oplus G_n \oplus \texttt{<<RS}, \texttt{<}R, e_1, \ldots, e_m\texttt{>}, \texttt{<}J_1, \ldots, J_n\texttt{>>>}.$$

Note that the last rule-schema of G is actually an expression and not a proper schema. ■

Given a context Γ, we let D_Γ be the set of derivations under Γ. For any $e \in \mathsf{E}$, we can decide whether $\Gamma \vdash e :: J$ for some J by induction on the construction of $e \in \mathsf{E}$. So, we see that D_Γ is a primitive recursive subset of E, and there is a primitive recursive function:

$$\mathrm{conc}_\Gamma : \mathsf{D}_\Gamma \to \mathsf{J}_\Gamma$$

such that $\Gamma \vdash_G D :: \mathrm{conc}_\Gamma(D)$ holds for all $D \in \mathsf{D}_\Gamma$. Then, by the characterization theorem of derivations, we see that if $\Gamma \vdash_G D :: J$, then $D \in \mathsf{D}_\Gamma$ and $J \equiv \mathrm{conc}_\Gamma(D) \in \mathsf{J}_\Gamma$.

Theorem 2 (Decidability) If $G \in \mathsf{S}$ and $\Gamma, D, J \in \mathsf{E}$, then it is primitive recursively decidable whether $\Gamma \vdash_G D :: J$ or not.

Proof: By induction on the construction of D as an element of E. ■

For a derivation game G, we let $\mathsf{D}(G)$ be the set of G-derivations under the empty context. Then, we have the following corollary which fulfills Kreisel's dictum.

Corollary 1: For any derivation game G, $\mathsf{D}(G)$ is a primitive recursive subset of E.

Proof: Suppose that an element $e \in \mathsf{E}$ is given. If $e \notin \mathsf{D}_{<>}$ then, by the characterization theorem, $e \notin \mathsf{D}(G)$. If $e \in \mathsf{D}_{<>}$, then by the Decidability Theorem 2, we can decide $e \in \mathsf{D}(G)$ or not by checking $<> \vdash_G e :: \mathrm{conc}_{<>}(e)$ or not. ■

We can also check the following properties of derivations.

Proposition 1: If $\Gamma \vdash D :: J$, then any free variable in J is declared in $\mathrm{GV}(\Gamma)$, and any free variable in D is declared in Γ.

Derivation games enjoy the following fundamental properties.

Theorem 3: The following properties hold for any derivation game G.

1. *Weakening.* If $\Gamma \vdash_G D :: J$ and $\Gamma \oplus \Gamma'$ is a context, then $\Gamma \oplus \Gamma' \vdash_G D :: J$.
2. *Strengthening for general variable.* If $\Gamma \oplus <x> \oplus \Gamma' \vdash_G D :: J$, and $x \notin \mathrm{FV}(\Gamma') \cup \mathrm{FV}(D) \cup \mathrm{FV}(J)$, then $\Gamma \oplus \Gamma' \vdash_G D :: J$.
3. *Strengthening for derivation variable.* If $\Gamma \oplus <X :: H> \oplus \Gamma' \vdash_G D :: J$, and $X \notin \mathrm{FV}(D)$, then $\Gamma \oplus \Gamma' \vdash_G D :: J$.
4. *Substitution for derivation variable.* If $\Gamma \oplus <X :: H> \oplus \Gamma' \vdash_G D :: J$ and $\Gamma \vdash_G D' :: H$, then $\Gamma \oplus \Gamma' \vdash_G [X := D'](D) :: J$.
5. *Substitution for general variable.* If $\Gamma \oplus <x> \oplus \Gamma' \vdash_G D :: J$, and $e \in \mathsf{E}$, then $\Gamma \oplus [x := e](\Gamma') \vdash_G [x := e](D) :: [x := e](J)$.
6. *Exchange.* If $\Gamma \oplus <e, f> \oplus \Gamma' \vdash_G D :: J$, and $\Gamma \oplus <f, e> \oplus \Gamma'$ is a derivation context, then $\Gamma \oplus <f, e> \oplus \Gamma' \vdash_G D :: J$.

These basic properties of derivations and G-derivations imply that it is possible to implement a system on a computer that can manipulate these symbolic expressions and decide the correctness of derivations. At Kyoto University we have been developing a computer environment called CAL (for Computation And Logic) [13] which realizes this idea.

REFERENCES

[1] Beeson, M.J., *Foundations of constructive mathematics*, Springer, 1980.

[2] Bell, D., *Frege's Theory of Judgement*, Clarendon Press, Oxford, 1979.

[3] Frege, G., Über Begriff und Gegenstand, *Vierteljahrsschrift für wissenschftliche Philosophie*, **16**, pp. 192–205, 1892.

[4] Harper, R., Honsell, F. and Plotkin, G., A framework for defining logics, *Journal of the Association for Computing Machinery*, **40**, pp. 143–184, 1993.

[5] Kreisel, G., Hilbert's programme, *Dialectica*, **12**, pp. 346–372, 1958.

[6] Kreisel, G., Foundations of Intuitionistic Logic, Nagel, Suppes, Tarski eds, *Logic, Methodology and Philosophy of Science*, Stanford University Press, pp. 198 – 210.

[7] Kreisel, G., Mathematical Logic, Saaty ed., *Lectures on Modern Mathematics, III*, Wiley, New York, pp. 95 – 195, 1965.

[8] Martin-Löf, P., An intuitionistic theory of types: Predicative part, *Logic Colloquium '73*, Rose, H.E. and Shepherdson, J.C., Eds, Studies in Logic and Foundation of Mathematics, **80**, North-Holland, pp. 73 – 118, 1975.

[9] Martin-Löf, P., *Intuitionistic Type Theory*, Bibliopolis, 1984.

[10] Martin-Löf, P., On the meanings of the logical constants and the justifications of the logical laws, *Nordic J. of Philosophical Logic*, **1**, pp. 11–60, 1996.

[11] Nordström, B., Petersson,K. and Smith, J.M., *Programming in Martin-Löf's Type Theory*, Oxford University Press, 200 pages, 1990.

[12] Okada, M., Ideal concepts, intuitions, and mathematical knowledge acquisitions: Husserl's and Hilbert's cases (a preliminary report), in this volume.

[13] Sato, M., Kameyama, Y., Takeuti, I., CAL: A computer assisted system for learning compuation and logic, in *EuroCAST 2001*, Lecture Notes in Computer Science,**2178**, 2001.

[14] Smorynski, C., The incompleteness theorems, Barwise, J. ed., *Handbook of mathematical logic*, Studies in logic and the foundations of mathematics, **90**, pp. 821 – 865, North-Holland, 1977.

[15] Sundholm, G., Constructions, proofs and the meaning of logical constants, *J. Phil. Logic*, **12**, pp. 151 – 172, 1983.

Evaluation of HTTP, E-mail and NNTP with Regard to Negotiation Requirements in the Context of Electronic Commerce

Safa Kaabi Chihi
The University of Manouba
National School of Computer
Science - Cristal Laboratory
2010 Tunisia
safa.kaabi@fst.rnu.tn

Hella Kaffel Ben Ayed
The University of Manouba
National School of Computer
Science - Cristal Laboratory
2010 Tunisia
hella.kaffel@fst.rnu.tn

Farouk Kamoun
The University of Manouba
National School of Computer
Science - Cristal Laboratory
2010 Tunisia
Frk.kamoun@planet.tn

Abstract-- **Recent developments of electronic commerce has led to explore the negotiation process in the electronic transactions. Several studies have focused on negotiation at the business and service levels. However, applications have been implemented in a proprietary way with regard to the communication level. This has resulted in a lack of interoperability between those applications. Our work focuses on the communication level underlying each negotiation application. First, we propose a description of the negotiation process in the contexts of B to C, B to B, and auctions and emphasize the characteristics of each context. We consider the negotiated terms, the result of the negotiation and the link between the negotiating parties. Then, we focus on the basic requirements of negotiation with regard to communication protocols: the connection mode, the security services, the message tracking, the notification service, the time of message validity, the fairness and the recoverability. We evaluate the following application protocols i.e. HTTP, E-mail and NNTP according to these requirements. This evaluation shows that HTTP is not adequate. E-mail seems to be the richest with regard to the specified requirements even though there is a lack of interactivity. NNTP seems particularly suitable for multiparty negotiation.**

Keywords-- communication protocol, negotiation, electronic commerce, requirements.

I. INTRODUCTION

ELECTRONIC commerce is divided roughly in two main aspects: marketing and contracting [11]. Electronic marketing focuses on promoting a firm's products and services to customers. Electronic contracting, focuses on negotiating the terms and the conditions of contracts. In some situations, this includes the monitoring of the contract performance [14]. Negotiation concerns both small and big value transactions and all categories of electronic commerce [15]. Negotiation in the context of electronic commerce is not a new practice. It has already been introduced in the EDI (*Electronic Data Interchange*) exchange process: trading partners had to proceed with the negotiation of an interchange agreement. This negotiation occurred offline and before the partners start the exchange of data related to their transactions [1][9].

Research in automated on-line negotiation is still in its Infancy [6]. Related studies have mainly focused on the negotiation at the business and the service levels [8] [28]. IT infrastructure and especially communication services have not received as much attention. Experimentations of on-line negotiation are based on several Internet applications such as e-mail [17], HTTP [4], chat [28], Whiteboard [28] and real-time collaborative tools[21][28]. However as these applications have not been designed for on-line negotiation, many missing services have been implemented in a proprietary manner. This has led to many non interoperable implementations. Hence, we believe that the specification of a suitable communication protocol would contribute to facilitate the deployment of interoperable negotiation applications.

In this paper we first propose a description of the negotiation process in the contexts of Business to Consumer (B to C), Business to Business (B to B), and auctions. Then, we define the basic requirements of a negotiation in the context of electronic commerce in terms of communication services. Finally, we evaluate the commonly used application protocols according to these requirements.

II. THE NEGOTIATION PROCESS

In [13], negotiation in electronic commerce is defined as:" the process by which two or more parties multilaterally bargain resources for mutual intended gain, using the tools and techniques of electronic commerce."

We may define the negotiation in the context of Electronic Commerce as an adaptation of an offer to a client's needs while considering the constraints of the merchant. It consists of a discussion that must end with an agreement or an abort.

Most studies who have focused on the description of the electronic transaction phases have considered a negotiation/agreement phase within the transaction [7][15][18].

The negotiation process describes the way of agreeing and settling business actions. It gives details on the way of exchanging information, the way of ordering and of settling the terms of the contract [13]. It starts with an offer, i.e. a bid, resulting from the previous phase (generally called information phase) and ends with a contract transmitted to the next phase (settlement phase). It is decomposed into three phases: a starting phase "start", a discussion phase "discussion" and a settlement phase "settlement"[13].

There are two forms of negotiation: a one-to-one i.e. bilateral negotiation, which involves two actors who decide about the conclusion of the negotiation, and a multilateral negotiation where many parties are involved simultaneously on a merchant offer [13]. Furthermore, negotiation can be carried out directly between negotiating parties, or through a third party which conducts the negotiation. This feature is called in what follows: "the link between negotiating parties".

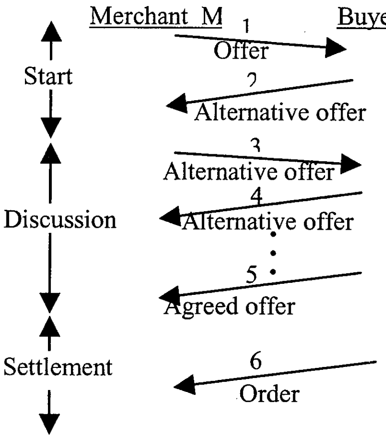
Our description of the negotiation process considers the contexts of B to C and B to B as they are the most deployed categories of electronic commerce. Although auctions may be carried in B to C and B to B electronic commerce, we consider auctions separately as they constitute a specific form of negotiation [12]. We analyze the negotiation process, the form of negotiation, the link between the negotiation parties, the terms negotiated and the contract resulting from the negotiation.

A. The Negotiation Process in the Context of B to C Electronic Commerce

This type of negotiation is growing with the expansion of electronic commerce on the Internet and the World Wide Web [17]. It is the result of several factors, such as the trend toward customization and the lowering of transaction's costs. Several public sites offer a negotiation service between distant parties. These sites are based on software agent systems[7], which negotiate on behalf of users such as AuctionBot [3] Kasbah [10] and Tete-a-Tete [19].

As shown in fig.1, the negotiation in a B to C transaction starts with an offer generated by the merchant site (stage1). The buyer gives an alternative offer corresponding to this offer (stage 2). The "discussion" begins when the merchant answers with an alternative offer and so on (stages 3,4,..). The discussion may concern different terms such as price, warranty, time of delivery, and terms of service. The goal of the consumer is to adapt these terms according to his preferences until reaching an agreement (stage 5). The fact to make an order (stage 6) represents a successful negotiation and leads to the establishment of a contract ("settlement").

A B to C negotiation is generally a one-to-one negotiation, which can be direct or indirect. In case it is direct, i.e. non-brokered commerce, consumers contact merchants directly to search suitable offers and/or to conduct a negotiation. This often happens in very small markets and occurs when the consumer knows the seller, and the items to buy [6]. However, the vast amount of commercial information available on the Internet has resulted in the raise of the number of electronic brokers, which provide secure, rapid and effective means of exchanging information and negotiation between consumers and providers [1]. This practice is becoming more and more appreciated in the electronic market.



1. Offer
2. Generation of an alternative offer by B
3. Approval of B's offer or creation of alternative offer by M
4. Creation of alternative offer by B

••• Possibility of iteration

5. The M's offer is agreed by B
6. Transformation of the agreed offer into order

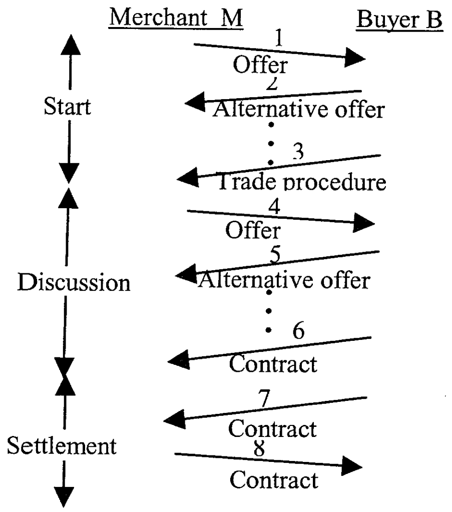
Fig.1 The Negotiation Process in the Context of B to C

B. The Negotiation Process in the Context of B to B Electronic Commerce

In the context of B to B, negotiation is an important phase in the electronic transaction. It is also called, as seen previously, agreement phase or electronic contracting phase. Negotiation within EDI is held within a closed and long-term relationship [1]. The more challenging case is open electronic contracting, which would allow the settlement of contracts among parties with no prior trading relationships. This represents a real need with the expansion of the Internet, as companies need more flexibility. Fig.2. illustrates the process of negotiation between two trading partners and shows the "start", the "discussion" and the "settlement" phases of the negotiation process:

- The "start" phase consists in a discussion of parameters to determine the way the negotiation process is expected to occur. These parameters may be related to the nature of documents that would be exchanged, the terminology, the security, the system of negotiation, the terms to discuss. Even actors that may be involved in specific cases such as a trusted third party would be defined in this phase. These parameters are carried within an offer and alternative offers exchanged between the merchant and the buyer (steps 1,2,. . .) until reaching an agreed trade procedure (step 3) [13].

-The discussion phase (steps 4,5, . . .) is similar to the B to C discussion phase and consists in the exchange of bids (offer/alternative offers) in order to reach an agreed contract (step 6).



1. Offer
2. Generation of an alternative offer by B

••• Possibility of iteration

3. Agreement on the trade procedure
4. Offer
5. Creation of alternative offer by B

••• Possibility of iteration

6. M's offer is agreed by B and is considered as a contract
7. Transformation of the agreed offer into contract
8. The M's contract signature

Fig.2 The Negotiation Process in the Context of B to B

-The settlement phase is focused on settling the contract commitments and eventually the contract signature (steps 7 and step 8).

Generally, a B-to-B negotiation is a one-to-one negotiation, which can be direct or indirect. However, negotiating parties usually negotiate and exchange documents through systems such as trusted third parties, electronic notaries, certification authorities, or any trustworthy environment who have to save all the discussion matters and the various commitments in particular the contract and the signature.

C. *The Negotiation Process in the Context of Auctions*

The Internet provides an infrastructure that enables the execution of auctions and bids in a much cheaper way than in traditional auctions. Furthermore, Internet auctions involve much more sellers and buyers [11]. Auctions constitute a form of multiparty negotiations, which obey to particular rules [4] [13]. An auction site (auctioneer) acts like a broker and offers services for sellers to post their products for sale and allows buyers to bid on those products. There are different types of auctions; most of them open with a starting bid [11].

We notice that auctions are usually organized by the auctioneer who provides the institutional setting of the auction. He also has to manage and save the different transaction phases of the trading process: information exchanges, the price determination, the trade execution and the settlement [12]. Generally, the only negotiated term is the price and the negotiation process is also called bargaining process [4][5][11]. The auction process may be decomposed in five basic stages [13]. In Fig.3, these stages are decomposed into steps to describe an auction process which involves a given buyer, the auctioneer, and the seller.

- stage 1: Bidder (i.e buyer) and Seller registrations (steps 1 and 2).
- stage 2: Setting up a particular auction event (step3, 4,...): This consists, in a discussion of several auction parameters such as the auction method and the schedule.
- Bidding (step 5): This is the process of transmitting bids from each bidder to the auctioneer.
- The evaluation of bids and notification of its result (step 6): There may be iterations between this step and the previous one.
- Notification of the auction result and closing the auction (step 7).

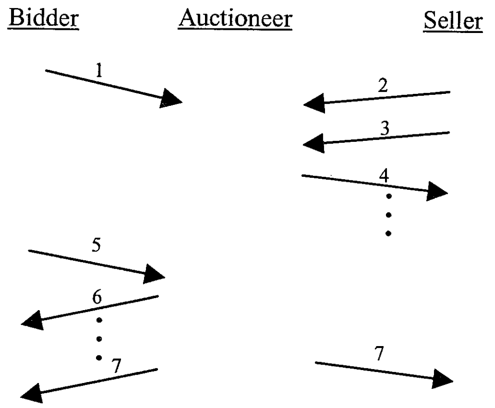


Fig.3 The Auction Process

Note that in this process, we can see that the negotiation is started between the auctioneer and each bidder after he registers. The discussion phase consists in the bidding process. The settlement phase corresponds to the notification of the auction's result to the bidders and/or the announcement of closing the bids.

III. NEGOTIATION REQUIREMENTS

We propose to formulate the following requirements: the connection mode, security services, message tracking, need for acknowledgment, time of a message validity, recoverability and fairness.

Connection mode:

In traditional transactions, negotiation is always held in the form of face-to-face meetings. When done between distant parties, the two parties communicate via a combination of telephone calls, Fax, e-mail, or videoconferencing or other communication medium. This combination results from the fact that telephone calls and videoconferences enable on line interaction but do not support working on detailed documents. Fax, e-mail and file transfer can be used to exchange documents but do not enable real time interaction between the negotiating parties [28]. Hence, we can say that traditionally, negotiation requires both on-line interaction and simultaneous exchange of documents. On the Internet, negotiation can be performed in connected or non-connected mode. A connected mode implies a synchronous connection between the parties engaged in the negotiation. A non connected mode is based on an asynchronous communication.

A synchronous negotiation might be based on the opening of a negotiation session and a real time exchange of offers and responses. It has the advantage of interactivity, which increases the rapidity of the decision process. Many systems supporting negotiation have been experimented over synchronous communication such as INSS (*Internet Negotiation Support System*) [28], CBSS (*Collective Bargaining Support System*) [28] and INSPIRE [8].

An asynchronous negotiation allows a non interactive negotiation in a batched scenario. It is usually based on electronic mail. This approach has the drawback of the lack of interactivity. This may result in slowing down the negotiation process. Hence, it should be adopted only in the case of impossibility of performing a negotiation in a connected mode [17]. This may result from a deficiency in the communication infrastructure or limitations at the merchant side. We think that this practice should be considered as a temporary solution. Negotiation should be performed over a connected mode protocol and a new concept of a "negotiation session" has to be considered and studied.

Security services:

Guaranteed security is an essential component of the communication protocol that will enable and support online negotiation. Involved companies need to be sure of :

- the party's identity they believe they are negotiating with. This may be provided by authentication techniques. In the context of B to C, authentication at the starting phase of the negotiation session may be needed; for example, when a particular client negotiates with a merchant site with which he has specific relationships. Furthermore, in non connected mode or in case of restarting a negotiation, it might be necessary to authenticate the client. In the context of auctions, this service might be required during the initial buyer and seller registration to prevent from unauthorized posting and alteration or violation of the auction rules [11].
- the secrecy and the protection of negotiated offers, alternative offers and contracts from a concurrent access. In a business environment, the consequence of a such access can be disastrous. Confidentiality mechanisms can ensure that information exchanged during the negotiation cannot be read, copied or modified by an unauthorized party.
- The protection against denial of service attacks.

Message tracking:

To enhance the non-repudiation service, message tracking might be necessary for B to B, B to C, and auctions contexts. In fact, both buyers and merchants may need to keep a trace that could serve as a reference or a proof in case of a conflict. Two main parameters have to be considered: the period of validity of the trace and the number of messages that have to be saved. These parameters depend on the type of the business and its importance. The trace may be kept at each negotiating parties' system. In this case, we have to consider the problem of the integrity of the trace. The trace may also be kept at a third party's system, which could be the auctioneer in case of auctions, or a server or a broker in case of B to C. In case of B to B, trusted third parties might provide additional services such as documentary storage with eventually time stamping and an archive service useful in case of dispute [15].

Notification:

A notification or an acknowledgment consists in confirming a message reception. We consider that a notification may be needed in the three studied contexts. When a negotiating party sends an offer, an alternative offer may be considered as an acknowledgment (fig.4). However, in the case of negotiation abort or the impossibility of an "immediate" response, an acknowledgment should be sent to confirm the offer's reception (fig.5).

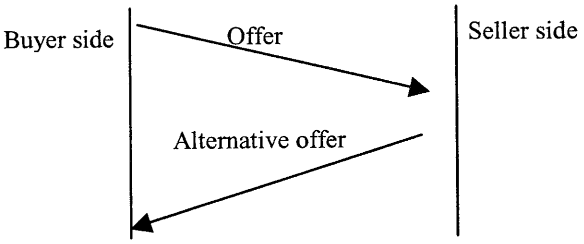


Fig.4 Case of alternative offer

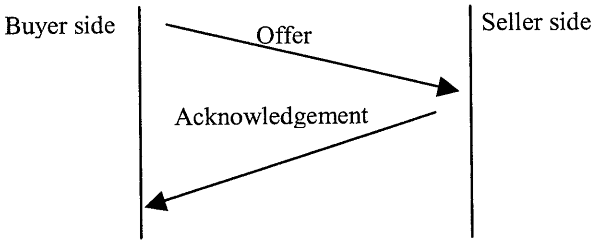


Fig.5 Case of notification

Time of message validity:

In some situations, the negotiating parties might need to take time to think about an offer. But usually an offer is valid for a certain time and then becomes obsolete. So, we can consider that offers and alternative offers are time sensitive information. As a result, a new concept, called time of message validity, should be taken into consideration within the communication protocol. This would provide a waiting period while an offer is still valid. Furthermore, the communication protocol should be able to manage situations resulting for instance from the sending of a counter offer related to an obsolete offer. This should be processed transparently to the negotiating parties.

Recoverability:

During a negotiation session, some particular conditions may cause a communication failure within the underlying layers of the TCP/IP protocol stack. When this happens it is recommended that the session restarts transparently to the negotiating parties. The problem of restarting a negotiation session has to be considered. Checkpoints might be defined to allow a restarted negotiation to begin from the last checkpoint before the failure.

Fairness:

Fairness expresses the fact that all actors involved in a negotiation process should have the same privileges as regards the communication i.e. in sending or receiving offers and counter offers [2]. This is particularly recommended in the context of auctions, where bidders need to compete with their bids without considering any obstacles related to the communication protocol. Their bids should be taken into account fairly by the auctioneer. In addition, notifications should be sent to all bidders similarly and simultaneously.

IV. EVALUATION OF SOME INTERNET PROTOCOLS WITH REGARD TO NEGOTIATION REQUIREMENTS:

As mentioned before, many Internet applications (e-mail, HTTP, chat, Whiteboard) have been used for on-line negotiation. Furthermore, we can note that both collaborative work (or groupware) and negotiation are characterized by the sharing of information and the building of a consensus over time [21]. We evaluate the relevant Internet protocols i.e. HTTP, E-mail and NNTP with regard to the previously defined requirements.

A. HTTP (HyperText Transfer Protocol)

HTTP [22] is a connection oriented protocol that allows the exchange of multimedia documents and may be enforced with security services provided by S/HTTP *(Secure HyperText Transfer Protocol)* protocol and SSL (*Secure Socket Layer*) protocol. The HTTP protocol is based on a request/response paradigm. It does not deal with any other negotiation requirements. Hence we may state that HTTP protocol is inefficient and poorly suited to on-line negotiation. An upper layer protocol has to be specified above HTTP to implement the missing services.

B. Internet Mail

E-mail is a technology used successfully in some situations for negotiation between different parties in a non connected mode [17]. It may provide security services by the use of S/MIME *(Secure/ Multipurpose Internet Mail Extensions)* [25] or PGP/MIME *(MIME Security with Open Pretty Good Privacy)* (RFC3156)[24]. These RFCs specify security functions such as integrity, confidentiality, authentication and non-repudiation of the origin. The use of MIME [23] allows the exchange of multimedia documents with the possibility to keep a trace of each transfer. Moreover, the expiry date field may be exploited to satisfy the time of message validity requirement. Furthermore, the MDN (*Message Disposition Notification*) concept (rfc2298) [27] can be used to implement the notification requirement. However, there is no mechanism for recoverability because e-mail is a non connected application.

TABLE 1.
EVALUATION OF HTTP, INTERNET MAIL AND NNTP ACCORDING TO NEGOTIATION REQUIREMENTS

Requirements	**http**	**Internet Mail**	**NNTP**
Connection mode	Yes	No	Yes
Security services	Yes	Yes	Yes
Message tracking	No	Yes	Yes
Notification	No	Yes	Yes
Time of message validity	No	Yes	Yes
Recoverability	No	Not specified	Not specified
Fairness	Not specified	Not specified	Not specified

C. NNTP (News Network Transfer Protocol)

NNTP defines *a protocol for the distribution, inquiry, retrieval and posting of news, articles using a reliable stream-based transmission of news* (RFC977) [26]. NNTP has been used for group communications (videoconferencing,..). It is based on a flood broadcast mechanism thus allowing information distribution to all client sites. Thanks to this mechanism, NNTP protocol seems particularly suitable for multi-party negotiation. It provides real-time interaction. Besides, NNTP is based on storing news in a Netnews server and managing articles. The management of articles includes expiration of aged messages [26]. So, NNTP can provide both time of message validity and tracking requirements. Moreover it supports the transfer of multimedia documents. As it is client-server oriented, it is able to provide fairness to all connected clients. However, the RFC977 does not deal with recoverability.

D. Conclusion

Table1 summarizes the results of our evaluation. It shows that HTTP is not sufficient for on-line negotiation. E-mail seems to satisfy all the requirements except the connection mode and the recoverability. Hence, we can say that e-mail services are suitable for one to one as well as multiparty negotiation. NNTP is also a suitable protocol for negotiation. Its broadcast mechanism makes it particularly interesting in the context of multiparty negotiation such as auctions. However, the recoverability and fairness aspects have to be studied for these protocols.

V. SUMMARY AND OUTLOOK

The growth of electronic commerce brought up a wide variety of systems that try to facilitate business transactions. Development of mechanisms for negotiation and contract execution has become an important area for research since negotiation affects all categories of electronic commerce with high or low value items.

In this paper we have described the negotiation process in the contexts of B to C, B to B and auctions. Three phases have been outlined the "start", "discussion", and "settlement" phases.

We have considered some basic requirements of a negotiation with regard to communication protocols: the connection mode, security services, message tracking and notification service, time of message validity, recoverability, and fairness.

In the last section, we have considered the three Internet application protocols that have been mostly used to experiment on line negotiation i.e. HTTP, e-mail, and NNTP. We have shown that HTTP is a poor protocol with regard to the defined requirements. Email seems to be the richest protocol. However the non connected mode results in a lack of interactivity which constitutes a major drawback for on-line negotiation. NNTP is a suitable protocol especially for multiparty negotiation.

This work constitutes a first step in the specification of the requirements of negotiation in term of communication services. In the near future, the needs of each context should be studied separately to identify a set of requirements specific to each category. E-mail and NNTP have to be better studied with regard to recoverability.

APPENDIX

This paper has been accepted for publication in ICECR4, Dallas, Nov.2001.

REFERENCES

[1] V. Almeida, W. Meira, V. Riberiro, N. Ziviani, "Efficiency Analysis of Brokers in the Electronic Marketplace", *Computers Networks,* 31, 1999.

[2] N. Asokan, "Fairness in Electronic Commerce", *IBM research division,* Mar. 1997.

[3] AuctionBot, URL: <http:// auction.eecs.umich.edu>

[4] C. Beam, A. Segev, "Automated Negotiations: A survey of the State of the Art", in WI 39, pp. 263-268, 1997.

[5] M. Bichler, A. Segev, "A brokerage Framework for Internet Commerce", *Fisher Center for Management and Information Technology, CMIT Working Paper* 98-WP-1031.

[6] M. Bichler, A. Segev, C. Beam, "Services of a Broker in Electronic Commerce Transactions", *EM-Electronic Markets,* vol.8, n°1, Mai 1998.

[7] R. H. Guttman, A.G. Moukas, P. Maes, "Agents as Mediators in Electronic Commerce", *EM-Electronic Markets,* vol.8, n°1, Mai 1998.

[8] INSPIRE de Carleton University <http://www.business.carleton.ca/inspire/>

[9] H. K. Ben Ayed, "The Impact of Electronic Commerce on EDI", *Proc. Third International Conference on Telecommunications and Electronic Commerce ICTEC'3.* Dallas, Nov.2000.

[10] Kasbah, URL: <http:// kasbah. media. mit.edu>

[11] S. Klein, "Auctions and Bidding on the Internet: An Assessment", *EM-Electronic Auctions. EM-Electronic Markets,* vol.7, n°4, Dec.1997.

[12] S. Klein, "Introduction to Electronic Auctions", *EM-Electronic Auctions. EM-Electronic Markets,* vol.7, n°4, Dec.1997.

[13] M. Kumar, S. Feldman, "Business Negotiations on the Internet", *IBM Institute for advanced commerce, Yorktown Heights,98.*

[14] M. Lee Ronald, "Towards Open Electronic Contracting", *International Journal of Electronic Commerce,* vol.8, n°3, 1998.

[15] A. L. Markus, R. Alexander, " Non Repudiation within the Electronic Contracting Phase of Electronic Commerce Transactions", http://www.netacademy.org

[16] A. L. Markus,A. Runge, "Electronic Contracting within the Reference Model for Electronic Markets", *Proc. of The 6th European Conference on Information Systems, ECIS'98.*

[17] L. Ook, "Negotiation and Contract through E-mail", *International Journal of Electronic Commerce,* vol.8, n°3, 1998.

[18] NSF Workshop, IC2 Institute, the University of Texas at Austin, "Research Priorities in Electronic Commerce", Sep. 1998. http://cism.bus.utexas.edu/workshop/ecdraft.htm. (workshop report, Jan. 1999).

[19] Schmid, B. Lindermann, "Elements of Reference Model for Electronic Markets", *Proc. of the 31st Annual HICSS,* IV,1998

[20] Tete-a-Tete, URL: < http:// ecommerce.media.mit.edu/tete-a-tete/>

[21] http:// www8.zdnet.com/pcmag/features/groupware/gpws5.htm

[22] www.ietf.org RFC1945

[23] www.ietf.org RFC1767

[24] www.ietf.org RFC3156

[25] www.ietf.org RFC2311

[26] www.ietf.org RFC 977

[27] ftp://ftp.isi.edu/in-notes/rfc2298.txt

[28] Y. Yuan, J. B. Rose, A. Norm and Suarga, "A Web-Based Negotiation Support System", *International Journal of Electronic Commerce,* vol.8, n°3, 1998.

A Tabu Search Heuristic for a Real Life Vehicle Routing Problem

Khaled MELLOULI
Institut des Hautes Etudes Commerciales
Tunis Carthage
Khaled Mellouli@ihec.rnu.tn

Jaber JEMAI
Institut Superieur de Gestion de Tunis
41 rue de la liberté cité Bouchoucha
2000 Bardo Tunis Tunisie
Jaber.Jemai@isg.rnu.tn

Abstract-- **In this paper, we present and develop an algorithm for the SNDP commercial assistants problem. The SNDP has a network of stations, each with a number of visits to be performed monthly and to each visit corresponds a service time. Commercial assistants hosted at multiple attachment points have to visit stations with respect to their capacity and the fact that two visits of the same station must be separated by at least one week.**

The real problem was modelled as a Vehicle Routing Problem VRP, after applying a transformation that flat the natural network to achieve the Extended Artificial Network EAN. Then, we develop a Tabu Search heuristic to solve the obtained problem.

Keywords-- Real Life Problem, Vehicle Routing Problem, Transformation, Tabu Search .

I. THE PROBLEM

THE SNDP is the national petroleum distribution company operating in Tunisia. It has a stations network covering the Tunisian territory distributing SNDP products like fuel, gazoil,... The SNDP has also a number of commercial assistants attached to several attachment points, their function is to periodically visit stations to control and assist commercial transactions. The frequencies of visits are not the same for all stations, each has a number of visits 1 or 2 or 3 calculated from its volume of sales concluded during the previous year. Fixed times service are associated to each visit.

On accomplishing their jobs, commercial assistants travel throw valuated arcs joining stations, also they spend, every working day, additional costs of breakfast, lunch, diner and night if they pass the night in a hotel.

The problem is to design a set of feasible routes for assistants to cover the network demand in visits while minimizing the total traveled time and the total assistants expenses.

A. Problem Data

The problem could be summarized as follow:

1. ***G(X,U)*** : be the stations graph where:
    - X is the station set.
    - $U=\{(i,j) / i \in X, j \in X$ and j is directly reachable from $i\}$ be the arcs set.
2. For all stations $i \in X$ a frequency function $f(i)=\{1,2,3\}$ calculated from the volume of the sales of the previous year.
3. For all stations $i \in X$ and during a visit $v=\{1,2,3\}$ a delivery time t_{iv} is associated.
4. For all arcs $(i,j) \in U$ a travelling time t_{ij} is associated.
5. ***ND*** the number of depots.
6. ***NV*** the number of commercial assistants.
7. The maximal route duration is fixed to be ***DurMax***.
8. The cost of a travelled kilometer is ***ck***.
9. The daily expense of a commercial assistant are defined as follow:
    - ***b*** for the breakfast,
    - ***l*** for the lunch,
    - ***dn*** is the diner and the night cost.
10. The average speed is ***AS***.

B. Problem Constraints

We find the following two types of constraints:

1. Technical constraints that guarantee the following facts:
    - route continuity,
    - every station must be visited a number of times equal to it's frequency function,
    - every commercial assistant must start and finish his tour at his depot.
2. Specific constraints related to the problem which are:
    - a route length doesn't exceed the prescribed bound ***DurMax***,
    - Two successive visits of the same station must be separated at least by a week.

II. THE EXTENDED ARTIFICIAL NETWORK

As presented, the SNDP commercial assistants problem shows a spatial nature with the associated frequencies. That's and in order to manipulate the difference in number of visits between stations and to meet the constraint of the interval between two successive visits of the same station. We choose to flat the real network by duplicating every station the value of it's frequency function times to obtain the Extended Artificial Network (**EAN**). The transformation employed to obtain the **EAN** is defined as follow:

Let ***G'(X',U')*** be the artificial graph obtained from the initial graph ***G(X,U)*** where:

1. $X'=\{i' / i'$ is a duplicate of i and $i \in X\}$.
2. $U'=\{(i',j') / i'$ and j' respectively duplicates of i and j and $(i,j) \in U\}$.

3. For all (i,j) ∈ *U*, for all i' and j' duplicates of i and j, we have, $t_{i'j'} = t_{ij}$.

We denote by :

- ***S*** the number of stations in the initial network,
- X_1 the set of stations to be visited once,
- X_2 the set of stations to be visited twice i.e. $f(i)=2$ and $|X_2| = M$,
- X_3 the set of stations to be visited three times i.e. $f(i)=3$ and $|X_3| = L$,
- ***PA*** are attachment points.

Then the number of vertices in the Extended Artificial Network will be $n=ND+S+(2 * L)+M$.

The elements of ***X*** must be sorted in decreasing order following the frequency function *f*. The structure of ***X*** will be:

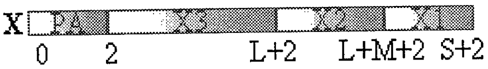


Fig. 1. Structure of the Set *X*.

The structure of the set ***X'*** will be:

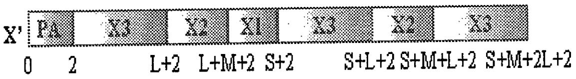


Fig. 2. Structure of the Set *X'*.

This structure permits also the identification of the duplicates of a given station i. Let i ∈ ***X*** and r be it's rank in ***X*** then we have:

1. If $f(i)=1$ then it's duplicate is r in ***X'***.
2. If $f(i)=2$ then it's duplicates are r and r + ***S*** in ***X'***.
3. If $f(i)=3$ then it's duplicates are r, r+***S*** and r+***S***+***L***+***M*** in ***X'***.

III. THE ALGORITHM

The transformed SNDP problem could be "naturally" viewed as a Vehicle Routing Problem following this analogy: commercial assistants are vehicles, stations in the Extended Artificial Network are cities or nodes to be visited and attachment points are depots. The route length is limited to ***DurMax*** and multiple depots are allowed. Then, the SNDP commercial assistants problem is modelled as a Multi-**D**epots **C**apacitated **V**ehicle **R**outing Problem (**DCVRP**).

The Vehicle Routing Problem is a real world problem that has been widely studied [1,2,4]. The standard VRP could be stated as follow: Given demands for goods or services at various geographically dispersed points in a transportation network and a fleet of vehicles stationed in a central depot, decisions concerning the spatial configuration of vehicles movements to supply points demand in such a way to minimize some objective are classified as "Vehicle Routing Problem".

Solving strategies for the **VRP** include three major classes. The first class contains exact procedures that attempt to find the optimal solution for the **VRP**. The second class is composed by classical heuristics which are approximate procedures that compromise between the quality of the founded solution and the needed running time for practical and large size **VRP** instances. Classical heuristics could be classified into three categories: constructive heuristics, two phase heuristics and improvement heuristics. The third class are metaheuristics which are optimization technics that give approximate and near-optimal solution for a variety of hard and very difficult problems where classical heuristics fail. Among metaheuristics, we find the Tabu Search have been widely applied to solve **VRP** instances.

Tabu Search is metaheuristic initially presented by Glover (1970) and then, developed by Glover (1986) and Hensen (1986) to be an evolved optimization technic. Tabu Search is a Local Search[a] heuristic that perform iteratively a sequence of modification on an initial solution to reach a local optima. Tabu Search escape local minima and cycling by employing a deterministic acceptance criterion using the memory concept. The memory i.e. Tabu List is structure that penalizes given moves that would return to recently visited solutions. That's why TS was called **Variable neighborhood method** [14] or **Dynamic neighborhood method** [17]. Tabu List Size is an important parameter to be adjusted. During the search process some tabu solution could appear attractive that's the aspiration criteria defines if the tabu status of a solution will be omitted and allow it's acceptance. Taillard [12] presents the intensification and diversification technics that improve the quality of a **TS** implementation. To succeed a **TS** implementation Glover [14] presents a set of tactical, technical and computational improvements that enhance the quality of the solution and the consumed processor time.

Recently, some powerful and promising concept has been proposed in tabu search including: the adaptive memory of Rochat and Taillard (1995) [20], the Granular Tabu Search (**GTS**) [23] and the Reactive Search [15]. Tabu Search have been widely applied for solving the VRP: Osman's algorithm [13], Tabouroute of Gendreau Hertz and Laporte[16], Taillard's algorithm [13], Xu and Kelly algorithm [22] and Rego and Roucairol algorithm[31]. Theses algorithm show a comparative advantage compared to other solving procedures.

IV. TABU SEARCH IMPLEMENTATIONN

TABU Search is a local search heuristic that improve a given initial solution by applying a sequence of transformations. Our initial solution generation algorithm

---

[a] They are also known by "Descent methods" [30], "Improving methods" [18] or simply "Iterative search" [22].

is an implementation of the Cristofides, Mingozzi and Toth algorithm which is basically designed for **CVRP** and **DVRP** instances like our case. The basic employed transformation is the (1,0) move type consisting of transferring a vertex from a route to another. To evaluate the visited solutions, we compute only the elements changed by the performed move instead of evaluating the complete new solution. In adding to this basic move, we implement a diversification scheme that uses a particular 2-Opt move. This move operates on two routes and must eliminate an existing route or create a new one while maintaining feasibility. Intensification technic has been applied by storing best founded moves according to their improvement. To avoid cycling, we implement a tabu list structure that records attributes of the moves recently performed. Selected attributes to be stored are the vertex actually transferred, its predecessor and its successor. This type of memory defines solutions with recorded attributes to have a tabu status. The size of the tabu list is deterministic and it have been determined by observing solution quality while varying the TL Size parameter. The aspiration criteria consisting of omitting the tabu status of a move that gives a solution better than the best founded so far is implemented within our heuristic. The search process stops if it can't improve the best founded solution.

A. Algorithm

The algorithm is composed by the following 5 phases:

1. Generation of the Extended Artificial Network,
2. Generation of the initial solution,
3. Iterative process: consisting of improving the given solution using the (1,0) move type,
4. Diversification,
5. Intensification.

Notation

- k iteration counter,
- λ the Cristofides, Mingozzi and Toth algorithm parameter,
- TL the tabu list,
- α Intensification threshold,
- i_k current solution,
- i^* the current best solution,
- $N(i,k)$ neighborhood of solution i at iteration k,
- $D(i_k)$ the set containing solution obtained by applying the diversifying move to solution i_k,
- $I(i_k)$ the set containing solution obtained by intensifying the solution i_k,

Algorithm Initialization

- k=0
- λ=1
- $|TL|$ =0 (Tabu List initialization)
- initialize TLSize=52
- initialize α=13

Main

```
Phase 1 : Generate the Extended Artificial Network.
Phase 2 : Generate the initial solution $i_0$.
Phase 3 : (Iterative Process)
    Step 1 : $k=k+1$, Generate a subset $V^*$ of solutions
             in $N(i,k)$.
    Step 2 : Choose a solution $j \in V^*$ such that:
        * $\Delta(j) > \Delta(s)$ for all $s \in V^*$.
        * $j$ is not a tabu solution or $Cost(j) < Cost(i^*)$
          (aspiration criteria)
      * set $i_{k+1} = j$.
    Step 3 : If $Cost(i_{k+1}) < Cost(i^*)$ then $i^* = i_{k+1}$, go to
             Phase 3 Else go to Phase 4.
Phase 4 : (Diversification)
    Step 1 : Generate $D(i_k)$ and select the best $j \in D(i_k)$
             Set $i_{k+1} = j$.
    Step 2 : If $Cost(i_{k+1}) < Cost(i^*)$ then $i^* = i_{k+1}$, go
             to Phase 3 Else goto Phase 5.
Phase 5 : (Intensification)
    Step 1 : Generate $I(i_k)$ and select the best $j \in I(i_k)$
             Set $i_{k+1} = j$.
    Step 2 : If $Cost(i_{k+1}) < Cost(i^*)$ then $i^* = i_{k+1}$, go
             to Phase 3 Else Stop.
```

V. THE SOFTWARE

The developed software for solving the SNDP commercial assistants problem, is composed by two parts: the first component deals with saving problem data and presenting the obtained solutions. This part calls the second one which is the optimizer by giving real data and problem parameters to solve.

The first part contains the database where all problem data and parameters are stored. The database is implemented using the ***ACCESS*** software. The interface is composed by all windows ensuring the communication between the user and the system. This first part was developed using the ***Visual Basic 6.0*** programming language.

The second component, the optimizer, is developed using the ***C*** programming language. It communicates with the first part using binary files to get problem parameters and to give the obtained solution. It's composed by four modules, which are:

1. Communication module: it read data and parameters from the input file and write the solution in the output file.
2. Initial solution generation module: it constitutes our implementation of Cristofides Mingozzi and Toth algorithm.
3. Moves module: it contains the implementation of the used move types.
4. Improvement module: it's the main Tabu Search algorithm.

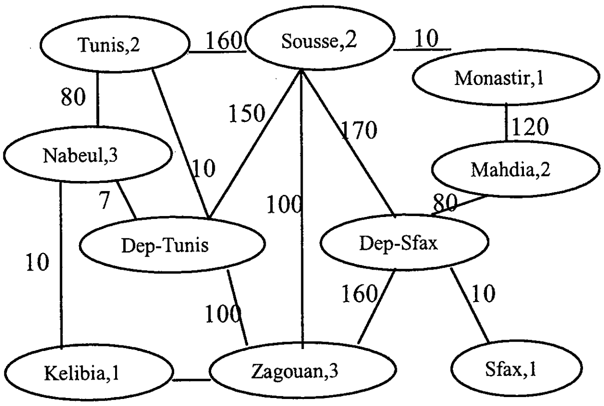


Fig. 3 : Example, real network

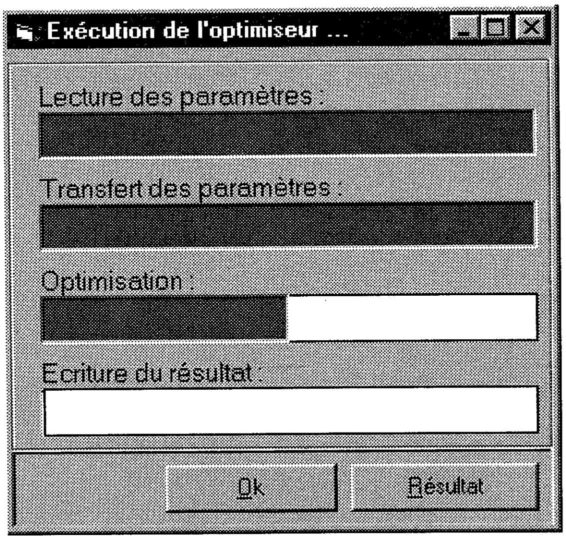
VI. EXAMPLE

Let's consider the real network described in Fig. 3 with two commercial assistants (***NV***=2) each hosted in a depot (***ND***=2). Their mission is to visit the 10 stations (***S***=8), noted by (Name,*f*) where Name is the station name and f is the associated frequency. Travelling time is associated to each arc. The problem is to find least cost feasible routes for assistants to visit stations.

Data details:

- ***DurMax***= 40 hours.
- ***Ck***= 0.1^{d}.
- ***b***=0.7^{d},
- ***l*** =3.5^{d},
- ***dn*** =30^{d}.
- ***AS*** = 70 km/h.

We have: $S = 8$, $L = 2$ and $M = 2$ then the number of nodes in the generated Extended Artificial Network is n=16.


Exécution de l'optimiseur ...

Lecture des paramètres :

Transfert des paramètres :

Optimisation :

Ecriture du résultat :

Fig. 4: Optimizer running


Edition d'une tournée ...

Tour :3

Attaché : CHAKER

Semaine : 3

Point d'attache : TUNIS

Séquence des stations visitées :
ZAGOUAN
MONASTIR
SOUSSE
NABEUL

Fig. 5: Route edition

Figure 4 and Figure 5 show some interface windows and the given solution has a cost of 31^{d} with the following routes:

Route-1 (Assist-1,Sem-1) : Dep-Tunis -> Zagouan ->
Sousse -> Nabeul ->
Tunis - > Dep-Tunis.
Route-2 (Assist-1,Sem-2) : Dep-Tunis -> Zagouan ->
Kelibia -> Nabeul ->
Tunis - > Dep-Tunis.
Route-3 (Assist-1,Sem-3) : Dep-Tunis -> Zagouan ->
Monastir->Sousse-> Nabeul
-> Dep-Tunis.
Route-4 (Assist-2,Sem-1) : Dep-Sfax -> Mahdia ->
Sfax - > Dep-Sfax.

VII. CONCLUSION

In this work, we address a real life problem which is the SNDP Commercial Assistant Problem, where a set of commercial assistants have to visit SNDP stations. The proposed problem have particularities in terms of the difference in the number of visits to be performed for each station and also in terms of it's specific constraints. The SNDP problem has been first transformed to achieve the Extended Artificial Network and then modelled as a Multi-Depots Capacitated Vehicle Routing Problem. A Tabu Search based heuristic with intensification and diversification has been developed to solve the given problem. The obtained results are satisfactory.

REFERENCES

[1] N. Chrisofides, "The Vehicle Routing Problem", RAIRO 10: 55-70 (1976).

[2] L.D. Bodin, B.L. Golden, A.A. Assad and M.O. Ball, "Routing and Scheduling of Vehicles and Crews: the State of the Art" , Computers and Operations Research 10: 69-211 (1983).

[3] J. Beasley, "Route First - Cluster Second Methods for Vehicle Routing" , OMEGA 11: 403-408 (1983).

[4] N. Chrisofides, "Vehicle Routing" , in E.L. Lawer, J.K.Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, The Traveling Salesman Problem. A guided of Combinatorial Optimization, Jhon Wiley, p. 431-448 (1983).

[5] G. Laporte, "Exact Algorithms for the Vehicle Routing Problem", Annals of Descrete Mathematics 31: 147-184 (1987).

[6] F. Glover, "Tabu Search -Part I", ORSA Journal of Computing 1: 190-206 (1989).

[7] F. Glover, "Tabu Search -Part II", ORSA Journal of Computing 2: 4-32 (1990).

[8] M. Haouari, P. Dejax and M. Desrochers, "Les Problèmes de Tournées de Vehicules avec Fenêtres de Temps: l'Etat de l'Art, Recherche Opérationnelle/ Operations Research 24: 217-244 (1990).

[9] G. Laporte, "The Vehicle Routing Problem: an Overview of Exact and Approximate Algorithms", European Journal of Operational Research 59: 345-358 (1992).

[10] I.H. Osman, "Metstrategy Simulated Annealing and Tabu Search Algorithm for the Vehicle Routing Problem", Annals of Operations Research 41: 421-451(1993).

[11] G. Laporte, "The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms", European Journal of Operational Research 59: 231-247 (1992).

[12] E. Taillard, "Recherches Itératives Dirigées Parallèles", Thèse n°1153 Département de Mathématiques , Ecole Polytechnique Fédérale de Lausanne (1993).

[13] E. Taillard, "Parallel Iterative Search Methods for Vehicle Routing Problem", Networks 23: 661-673 (1993).

[14] F. Glover, E. Taillard and D. De Werra, "A User's Guide to Tabu Search", Annals of Operations Research 41: 3-28 (1993).

[15] R. Battiti and G. Tecchiolli, "The Reactive Tabu Search", ORSA Journal of Computing 6(2): 126-140(1994).

[16] M. Gendreau, A. Hertz and G. Laporte, "A Tabu Search Heuritic for the Vehicle Routing Problem", Management Science 40: 1276-1290(1994).

[17] A. Hertz, E. Taillard and D. de Werra, "A Tutorial On Tabu Search", http//www.idsia.ch/~eric/articles.dir (1994).

[18] A.H. Eiselt, M. Gendreau and G. Laporte, "Arc Routing Problems, Part I: The Chinese Postman Problem", Operations Research 43: 231-242 (1995):.

[19] A.H. Eiselt,M. Gendreau and G. Laporte, "Arc Routing Problems, Part II: The Rural Postman Problem", Operations Research 43: 399-413 (1995).

[20] Y. Rochat and E. Taillard, "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing", Journal of Heuristics 1: 147-167(1995).

[21] I.H.Osman and G. Laporte, "Metaheuristics: A Bibliography", Annals of Operations Research 63: 513-628(1996).

[22] J. Xu and J.P Kelly, "A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem", Transportation Science 30: 379-393(1996).

[23] P. Toth and D. Vigo, "The Granular Tabu Search (and it's Application to the Vehicle Routing Problem)", Working paper, DEIS, University of Bologna (1998).

[24] C. Rego, "A Subpath Ejection Method for the Vehicle Routing Problem", Management Science 44: 1447-1459 (1998).

[25] G. Laporte and F. Semet, "Classical Heuristics for the Vehicle Routing Problem", Les cahiers de GERAD: G-98-54 (1999).

[26] M. Dorigo, G. Di Caro and M. Gambardella, "Ant Algorithms for Descrete Optimization", To appear in Artificial Intelligence Life MIT Press (1999).

[27] E. Taillard, "Ant Systems, Technical Report", IDSIA-05-99 (1999).

[28] F. Glover and S. Hanafi, "Tabu Search and Finite Convergence", Hearn Center of Enterprise Science, Report number: HCES-04-99(1999).

[29] M. Gendreau, G. Laporte and J.Y. Potvin, "Metaheuristics for the Vehicle Routing Problem", Les Cahiers de GERAD: G-98-52 (1999).

[30] Van Breedam, "Comparing Descent Heuristics and Metaheuristics for the Vehicle Routing Problem", Computers and Operations Research 28: 289-315 (2001).

[31] C. Rego and C. Roucairol : A parallel tabu search algorithm using ejection chains for the Vehicle Routing Problem, In Osman I.H. Kelly J.P., Editors Meta-Heuristics: Theory and Applications, pages: 661-675. Kluwer, Boston.

[32] Sam R. Tangiah: Vehicle Routing Problem with Time Windows using Genetic Algorithms, submitted to the book "Applications Handbook of Genetic Algorithms: New frontiers"

[33] Sam R. Tangiah, I. H. Osman and T. Sun: Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Methods for Vehicle Routing Problem with Time Windows, http://www.uwasa.fi.

Control Structures in Type Theory

Yukiyoshi Kameyama
Institute of Information Sciences & Electronics
University of Tsukuba
Tsukuba, 305-8573, JAPAN
kam@is.tsukuba.ac.jp

Abstract— **This paper studies control structures in functional programming languages. We focus on the static and dynamic features of control operators. Wihle the static control operators such as the first-class continuation operator have been studied by many researchers, the dynamic ones, such as catch/throw in Common Lisp, have not attracted much research interests so far. We show that dynamic control operators can be also formulated in type theories, and by giving an extension of the standard CPS-translation, we can show the CPS-transform preserves typing and reductions. The trick here is that the extended CPS-translation carries multiple continuation parameters throught the translation.**

Keywords— **Control operators, Type system, Classical logic, CPS-translation.**

I. INTRODUCTION

Control operators and data types are the two most important areas in the study of programming languages. Type theory is no double the important foundation for data types, but it can also serve as a foundational theory for control operators. Since Griffin observed in [3] that the Curry-Howard isomorphism can be extended to connect the type system for the call/cc operator and classical logic, there have been much research on connecting type systems for sequential control operators and classical logic [9], [4], [10], [12], [8]. Parigot's $\lambda\mu$-calculus [11] is one of the most studied logic, which does correspond to the call/cc-like control operators.

However, the studies so far done were mainly on the static control operators. Several existing control operators such as catch/throw in Common Lisp, exception in ML, and try-catch in JAVA have dynamic features, and even shift/reset, recently invented control operators, are dynamic. There remains much room for the study on these real control operators.

This paper tries to give a similar picture for dynamic control operators as one for static control operators, thus makes more uniform understanding to control operators. Our basic tool here is a simple extension of the standard CPS-transform in which several continuation parameters, rather than just one continuation parameter, are passed through the transform. We shall show that some formulation of a natural deduction style logic with multiple consequences nicely represents the dynamic control operators, and that the logic reflects the multiple continuation paramters in our extended CPS-tranform. In particular, we prove the type-preservation and reduction-preservation theorems hold for our extended CPS-tranform under certain conditions.

We examine this idea in formulating two important control operators: the first example is the exception mechanism, that is, catch/throw in Common Lisp, exception in ML, and try/catch in JAVA.

The second one is Danvy-Filinski's shift/reset operators. For the last decade, these operators are used by many researchers on partial evaluation and CPS-tranform, yet, there is no logic to reason about their properties. The only semantics is given through a CPS-translation but it does not preserve typing in a usual sense. In the previous work, we formulated a static variant of shift/reset in a type system which admits the Curry-Howard isomorphism. However, there is a demerit for the static variant, that is, it does not have a simple CPS-transform. In this paper, we give a standard type theory and also give an extended CPS-transform which preserves typing and reduction in the usual sense.

Thielecke also proposed an extension of CPS-transform with multiple continuations [13]. His CPS-transform carries two continuations to uniformly represent static controls, dynamic controls, and returning controls. A big problem in his approach is that his type system does not enjoy the subject reduction property. Our extended CPS-transform can be regarded as a fix for this problem. However, we focus on dynamic controls rather than comparing static/dynamic/ returning continuations, since static/returning continuations are already studied intensively.

This paper is organized as follows: Section II intoduces our terminology of static/dynamic control operators, and Section III introduces Thieleke's double-barrelled CPS-tranform. In Section IV we analyze the dynamic exception mechanism. We first give the type system and the reduction rules, and then give finitely many CPS-transform for the source language. We finally prove some theoretical results such as type-preservation under a certain conditions. In Section V we analyze Danvy and Filinski's shift/reset operators in a similar procedure as Section IV. Finally, Section VI gives concluding remarks.

II. STATIC VS DYNAMIC OPERATORS

We first make clear the difference of static and dynamic control operators.

Since control operators are functional analogue of the "go-to" statement in procedural languages, there should be two control points; the taking-off point (corresponding to "go") and the landing point (corresponding to "label"). In some language, the binding of go's and label's is determined lexically (when a program is written), and in another language, it is determined in a dynamic way. The latter case splits into two cases; the binding is determined at the time "label" is evaluated, or at the time "go" is evaluated. So there are (at least) three cases of binding. We distinguish these three cases, and call the first one *static* (sometimes called *lexical*) and the third one *dynamic* in this paper. We do not treat the second one in this paper.

For instance, the first-class continuation (**call/cc**) is static; in a program $\mathbf{call/cc}\lambda f.C[fM]$, the label is $\mathbf{call/cc}\lambda f.$, and the "go" is an occurrence of the variable f. Since the binding is represented by the variable-binding mechanism, it ought to be lexical. The μ-construct in Parigot's $\lambda\mu$-calculus is also static. Another example of static operators is the static catch/throw mechanism studied by Nakano [10], Sato [12], and myself [8].

On the other hand, the real catch/throw mechanism in Common Lisp is dynamic; when a `throw`-expression is evaluated, it is caught by the closest `catch` with the same tag. The exception mechanism in ML and the try-catch mechanism in JAVA are also dynamic.

The difference of static/dynamic is illustrated by the following example using Lisp-style catch/throw where α is a tag for these control operators:

(Static Catch/Throw Operators)

$$\begin{array}{ll} & \mathtt{catch}_\alpha((\lambda f.1+\mathtt{catch}_\alpha(f0))(\lambda y.\mathtt{throw}_\alpha y)) \\ \to & \mathtt{catch}_\alpha(1+\mathtt{catch}_\beta((\lambda y.\mathtt{throw}_\alpha y)0)) \\ \to & \mathtt{catch}_\alpha(1+\mathtt{catch}_\beta(\mathtt{throw}_\alpha 0)) \\ \to & \mathtt{catch}_\alpha(\mathtt{throw}_\alpha 0) \\ \to & 0 \end{array}$$

Note that, in the static calculus the tag variable α was renamed to β to avoid the unwanted capture of the free tag variable. The corresponding `catch`-construct for `throw` is the outer one.

(Dynamic Catch/Throw Operators)

$$\begin{array}{ll} & \mathtt{catch}_\alpha((\lambda f.1+\mathtt{catch}_\alpha(f0))(\lambda y.\mathtt{throw}_\alpha y)) \\ \to & \mathtt{catch}_\alpha(1+\mathtt{catch}_\alpha((\lambda y.\mathtt{throw}_\alpha y)0)) \\ \to & \mathtt{catch}_\alpha(1+\mathtt{catch}_\alpha(\mathtt{throw}_\alpha 0)) \\ \to & \mathtt{catch}_\alpha(1+0) \\ \overset{*}{\to} & 1 \end{array}$$

In this case, the `throw`-expression is captured by the inner `catch`-construct.

Note that Danvy and Filinski used the word "static" differently. In our sense, their shift/reset operators are dynamic as we shall see later.

There are many results which connect type systems for static control operators to classical logic. Besides Griffin's pioneering work for the call/cc operator, Nakano, Sato, and the author studied static variant of the catch/throw operators. Independently, de Groote [4] studied a variant of the exception mechanism in ML. For all these theories, the type system enjoys many nice properties such as the subject reduction, Church-Rosser, and strong normalizability, and does correspond to classical logic. There is also a type-preserving CPS-transform which does correspond to Kolmogorov-embedding from classical logic to intuitionistic logic. However, if we restrict the catch/throw mechanism to be static, it can be represented by the call/cc operator, or the call-by-value variant of Parigot's $\lambda\mu$-calculus.[1]

Compared to the static control operators, dynamic ones are less studied. An obvious reason is that, dynamic ones do not fit the Curry-Howard correspondence and we cannot have a simple logic for them. This situation is quite unsatisfactory since we see many examples of dynamic control operators in real programming languages; catch/throw in Common Lisp. exception in ML, and try/catch in JAVA. The purpose of the present paper is to exploit the dynamic control operators and to give a theoretical foundation similar to that for static control operators.

III. THIELECKE'S DOUBLE-BARRELLED CPS-TRANSFORM

Thielecke [13] introduced "Double-Barrelled CPS-transform" which passed two continuations through the transform, and showed that various control operators can be uniformly formulated using this setting.

His language contains two simple control operators `here` and `go`. These are similar to `catch` and `throw`, respectively. For example, $\mathtt{here}(E[\mathrm{go}V])$ evaluates to V where $E[\,]$ is an evaluation context and V is a value. To be more precise, these control operators are closer to the first class continuations rather than the simple catch/throw, since the effect of go remains even after the computation of the whole `here`-expression finishes.

---

[1] To be precise, the static catch/throw calculi have non-deterministic reduction rules, so they cannot be represented by the call/cc-operator. We can encode the catch/throw operators by the corresponding first-class continuation operators if we fix an evaluation strategy.

He introduced the double-barrrelled CPS-transform which passes two continuation parameters k and q. The extra parameter q represents the escaping continuation which is saved when `here` is evaluated. The whole definition of his CPS-transform is shown below:

$$\begin{array}{rcl}
[\![x]\!] & \equiv & \lambda kq.kx \\
[\![\lambda x.M]\!] & \equiv & \lambda ks.k(\lambda xrd.[\![M]\!]r\boxed{?}) \\
[\![MN]\!] & \equiv & \lambda kq.[\![M]\!](\lambda m.[\![N]\!](\lambda n.mnkq)q)q \\
[\![\mathtt{here}M]\!] & \equiv & \lambda kq.[\![M]\!]kk \\
[\![\mathtt{go}M]\!] & \equiv & \lambda kq.[\![M]\!]qq
\end{array}$$

where $\boxed{?}$ in the second clause is one of s, d, or r, depending on the choice of static, dynamic, or returning control:

- If s is chosen as $\boxed{?}$, then `here` and `go` are static control operators similar to **call/cc**.
- If d is chosen, then `here` and `go` are dynamic, and can be considered as `catch` and `throw` in Common Lisp.
- If r is chosen, then `here` and `go` represent the returning continuation, namely, evaluating a go-term jumps to the latest function call.

He also introduced type systems shown in Figure 1 for each form of control operators and proved that the typability is preserved through the above CPS-transform.

$$\frac{}{\Gamma, x : A \vdash x : A \ ; \ C} \qquad \frac{\Gamma, x : A \vdash M : B \ ; \ C}{\Gamma \vdash \lambda x.M : A \to B \ ; \ C}$$

$$\frac{\Gamma \vdash M : A \to B \ ; \ C \quad \Gamma \vdash N : A \ ; \ C}{\Gamma \vdash MN : B \ ; \ C}$$

$$\frac{\Gamma \vdash M : B \ ; \ B}{\Gamma \vdash \mathtt{here}M : B \ ; \ C} \qquad \frac{\Gamma \vdash M : B \ ; \ B}{\Gamma \vdash \mathtt{go}M : C \ ; \ B}$$

Fig. 1. Thielecke's Type System

He allowed only one type after the semicolon, which reflects that he has only one extra continuation parameter other than the standard continuation parameter in his CPS-transform.

An apparent problem of this decision is that, the subject reduction property does not hold as shown below. Let t be $(\lambda x.\mathtt{here}(yx))(\lambda w.\mathtt{go}z)$. Then, we can deduce $y : (A \to B) \to D, z : C \vdash t : D \ ; \ C$ by these inference rules. Under the call-by-value β-reduction, t reduces to $\mathtt{here}(y(\lambda w.\mathtt{go}z))$ which is not typable unless $C = D$.

The obvious reason of the failure of subject reduction is that we can have only one type after the semicolon while two (C and D) are needed in this example. So we can fix this failure by introducing multiple occurrences of types after the semicolon, that is, $\Gamma \vdash t : A \ ; \ B_1, \cdots B_n$ for $n \geq 0$. Such a type system was already studied by Nakano and Parigot to give the logical interpretation for the static control operators. Here we use (almost) the same type system for dynamic control operators.

IV. DYNAMIC EXCEPTION OPERATORS

As our first example, we study the dynamic exception operators in this section.

A. The Source Language

Our source language is an extension of simply typed lambda calculus $\lambda^{\to}$ in two ways. For the types, the function type $A \to B$ is changed to $A \overset{\Delta}{\to} B$ where Δ is a finite set of types. Intuitive meaning of this type is that, given an argument of type A, it returns a value of type B or it raises an exception of type $B \in \Delta$. For the terms, two control operators `catch` and `throw` are added to term-constructs. In this simple language, the `catch`/`throw`-expressions do not have tags unlike those in Common Lisp. Instead, we use types to select the matching `catch`; when a `throw`-expression is evaluated, it will be caught by the closest `catch` which has the same type of the thrown value. Here the "closest" one is determined in a dynamic way. For instance, `catch`(0 + `throw`1) evaluates to 1. To increase readability we may sometimes write types of `catch`/`throw` expressions like $\mathtt{catch}_A M$.

The type system of this language is shown in Figure 2. Each judgement has the form $\Gamma \vdash M : A \ ; \ \Delta$ where Γ is a finite set of variable-type declarations, M is a term, A is a type, and Δ is a finite set of types. Since we do not have to distinguish catch/throw-expressions of the same type, Δ is simply a set of types, rather than a set of types with indexes as in $\lambda\mu$-calculus.

$$\frac{}{\Gamma, x : A \vdash x : A \ ; \ \{\}} \qquad \frac{\Gamma, x : A \vdash M : B \ ; \ \Delta}{\Gamma \vdash \lambda x.M : A \overset{\Delta}{\to} B \ ; \ \{\}}$$

$$\frac{\Gamma \vdash M : A \overset{\Delta}{\to} B \ ; \ \Delta' \quad \Gamma \vdash N : A \ ; \ \Delta''}{\Gamma \vdash MN : B \ ; \ \Delta \cup \Delta' \cup \Delta''}$$

$$\frac{\Gamma \vdash M : A \ ; \ \Delta}{\Gamma \vdash \mathtt{throw}_A M : B \ ; \ \Delta \cup \{A\}}$$

$$\frac{\Gamma \vdash M : A \ ; \ \Delta}{\Gamma \vdash \mathtt{catch}_A M : A \ ; \ \Delta - \{A\}}$$

Fig. 2. Type System for Exception

We then define reduction rules to give an operational semantics of the language.

Noting that we are working in a call-by-value evaluation order, values and evaluation contexts are defined as usual:

$$V \quad ::= \quad x \mid \lambda x.M$$

$$E \quad ::= \quad [\,] \mid EM \mid VE \mid \mathtt{throw}_A E \mid \mathtt{catch}_A E$$

As usual, $E[M]$ denotes the term obtained by replacing the hole $[\,]$ in E by M.

The 1-step reduction is defined as the union of the following relations:

$$\begin{array}{rcl} E[(\lambda x.M)V] & \to & E[M\{x := V\}] \\ E[\mathtt{catch}_A V] & \to & E[V] \\ E[\mathtt{catch}_A E'[\mathtt{throw}_A V]] & \to & E[V] \end{array}$$

where in the last rule, E' is an evaluation context where $E' \equiv F[\mathtt{catch}_A F']$ does not hold for any evaluation contexts F and F'. Note that, the corresponding `catch` is selected at the time the `throw`-expression is evaluated, i.e., in a dynamic way. The equality between two terms are the least equivalence relation which contains the 1-step reduction.

This system does not enjoy the subject reduction property, if we allow functional types for `catch`. For instance, if we evaluate $(\mathtt{catch}_{A \to B}(\lambda x.\mathtt{throw}_{A \to B} V))V'$, then it raises an uncaught exception. On the contrary, if all the exception types are atomic, the subject reduction property holds.

B. Finitely Many Continuation Passing Transform

We then introduce an extension of the standard call-by-value CPS-transform, where finitely many continuations, rather than a single continuation, are passed through the transform. The extra continuations represent dynamic controls such as the "catch"-point (labels).

For technical reasons, we cannot neatly define a CPS-transform which passes infinitely many continuation parameters, we give a finitely many CPS here. We call this CPS *finitely many continuation passing transform* (FMCPS in short).

Let $B_1, \cdots, B_n$ be enumeration of types of the `catch/throw` expressions used in source terms, namely, we assume that, for any implicational type $A \overset{\Delta}{\to} B$ appearing in the type derivations, Δ is a subset of $\{B_1, \cdots, B_n\}$. Relative to this choice, we shall give a CPS-transform of terms which may not contain `catch/throw` expressions of other types. The type-part of FMCPS is given as follows where $*$ is the answer type:

$$\begin{array}{rcl} [\![A]\!] & \equiv & (A^* \to *) \to \\ & & (B_1^* \to *) \to \\ & & \cdots \\ & & (B_n^* \to *) \to * \\ A^* & \equiv & A^* \ \text{if } A \text{ is atomic} \\ (A \overset{\Delta}{\to} B)^* & \equiv & A^* \to [\![B]\!] \end{array}$$

Note that this constitutes a definition only when all B_i's are atomic. Otherwise, the above clauses generate a recursive equation. For instance, let us assume $n = 1$ and $B_1 = A \to C$ with A and C being atomic. Then we have:

$$B_1^* = A^* \to (C^* \to *) \to (B_1^* \to *) \to *$$

To define B_1^* appropriately, we need recursive types.

The term-part of FMCPS is given as follows:

$$\begin{array}{rcl} [\![x]\!] & \equiv & \lambda k\overline{p}.kx \\ [\![\lambda x.M]\!] & \equiv & \lambda k\overline{p}.k(\lambda x k'\overline{p'}.[\![M]\!]k'\overline{p'}) \\ [\![MN]\!] & \equiv & \lambda k\overline{p}.[\![M]\!](\lambda m.[\![N]\!](\lambda n.mnk\overline{p})\overline{p})\overline{p} \\ [\![\mathtt{catch}_{B_j} M]\!] & \equiv & \lambda k\overline{p}.[\![M]\!]k\overline{p}\{p_j := k\} \\ [\![\mathtt{throw}_{B_j} M]\!] & \equiv & \lambda k\overline{p}.[\![M]\!]p_j\overline{p} \end{array}$$

where $\overline{p}$ represents $p_1, \cdots, p_n$. For instance, $\lambda k\overline{p}.M\overline{p}$ is an abbreviation of $\lambda k p_1 \cdots p_n.M p_1 \cdots p_n$. Also the notation $\overline{p}\{p_j := k\}$ represents the sequence $p_1 \ldots p_{j-1} k p_{j+1} \cdots p_n$.

We then state the results about FMCPS-transform.

If all types in $B_1, \cdots, B_n$ are atomic, then we can take the target language of FMCPS-transform as $\lambda^{\to}$.

Theorem 1: If all types in $B_1, \cdots, B_n$ are atomic, then the FMCPS-transform preserves typing and equality.

Namely, if $\Gamma \vdash M : A\ ;\ \Delta$ is derivable in the source language, then $\Gamma^* \vdash [\![M]\!] : [\![A]\!]$ is derivable in $\lambda^{\to}$. Moreover, if $M = N$ in the source language, then $[\![M]\!] = [\![N]\!]$ in the call-by-value semantics of $\lambda^{\to}$. (*End of Theorem*)

In the statement of the theorem, Γ^* means $\{x_1 : A_1^*, \cdots, x_k : A_k^*\}$ if $\Gamma \equiv \{x_1 : A_1, \cdots, x_k : A_k\}$.

We could refine this theorem by replacing the CPS-transform by a more sophisticated one so that not only the equality but also the reduction (with the direction) is preserved. From this fact, we have the strong normalizability of the source language, since 1-step reduction in the source is translated into 1 or more step reduction in the target.

In the case where there are non-atomic types for thrown values, we need to extend the type structure of the target language to include recursive types in order to achieve type preservation. First, pseudo-types are defined to be: type variables $X_1, \cdots, X_l$, atomic (constant) types, funtional type $A \to B$, and recursive type $\mu X_i.A$. The μ-construct binds the type variable, and if a pseudo-type does not have free type variables then it is called a type. For instance, $A \to \mu X.(X \to B)$ is a type. We call $\lambda^{\to}$ with arbitrary recursive types $\lambda^{\to,\mathbf{rec}}$.

Theorem 2: If $\Gamma \vdash M : A \; ; \; \Delta$ is derivable in the source language, then $\Gamma^* \vdash [\![M]\!] : [\![A]\!]$ is derivable in $\lambda^{\to,\mathbf{rec}}$.

Moreover, if $M = N$ in the source language, then $[\![M]\!] = [\![N]\!]$ in the call-by-value semantics of $\lambda^{\to,\mathbf{rec}}$. (*End of Theorem*)

These theorems establish that, we can treat the dynamic exception mechanism in a similiar way as the static one. However, there are deficiencies in dynamic one if we allow non-atomic exception types, for instance, the subject reduction property does not hold.

Note also that, even if we adopt a more elaborate CPS-tranform, we cannot show the strong normalizability of the source language in this case, since the target language is not strongly normalizing. In fact, there are non-terminating terms in the source language if we allow non-atomic exception types.

C. JAVA's Try/Catch

As an example of formulating control operators in real programming languages, we see how the try/catch-mechanism in JAVA can be formulated in our setting.

The programming language JAVA has an involved mechanism for user-invoked exception, called *checked exception*. A typical program using the checked exception is as follows where we intentionally omitted the `finally` clause, because in the absence of side-effects, the `finally` clause cannot have much meaning.

```
try {
   M
} catch (ExcName1 e) {
   N1
} catch (ExcName2 e) {
   N2
}
```

In ML this expression is written as:

```
M handle ExcName1(e) => N1
       |     ExcName2(e) => N2
```

In JAVA, the two identifiers `ExcName1` and `ExcName2` are classes for exceptions, and if `throw`V expression is evaluated during the evaluation of M and V is an instance of class `ExcName1`, then the computation of M is aborted and $N1$ is evaluated with the variable e bound to V. If the evaluation of M terminates normally, then the value of M becomes the result of the whole expression.

We can formulate the try/catch mechanism in our setting. We assume that there is only one catch-clause in a try-expression each exception has a single argument, and the exception type is uniquely determined from its argument type. All these restrictions are superflous and we can obviously extend our treatment to the general case. Let $\neg A$ be the exception type whose argument type is A. Then we have the following two typing rules:

$$\frac{\Gamma \vdash M : B \; ; \; \Delta \quad \Gamma', e : A \vdash N : B \; ; \; \Delta'}{\Gamma \cup \Gamma' \vdash \mathtt{try}\ M\ \mathtt{catch}_{\neg A} \lambda e.N : B \; ; \; (\Delta - \{A\}) \cup \Delta'}$$

$$\frac{\Gamma \vdash M : A' \; ; \; \Delta}{\Gamma \vdash \mathtt{throw}_{\neg A} M : B \; : \; \Delta \cup \{A\}}$$

The FMCPS-tranform for these operators are obvious, and we again have the theorems. Since we may consider the exception types in JAVA are non-functional, we do not have type-errors (failure of subject reduction) in the evaluation of try/catch in the above form, and also the target language of FMCPS-transform is simply typed lambda calculus.

The exception mechanism of ML can be formulated in almost the same way as the `try`/`catch`-mechanism in JAVA, except that, we have functional types for exception types in ML, so we need to extend the type system of the target language of FMCPS-transform to the recursive types. In fact, there exist non-terminating programs in ML using only the exception mechanism.

Similarly, the catch/throw mechanism of Common Lisp can be formulated if we restrict the tags of catch/throw to be constants and there exist only finitely many tags. Unfortunately, the general case, in which arbitrary expressions are allowed for tags, is beyond our formulation.

V. THE SHIFT/RESET OPERATORS

Danvy and Filinski's shift/reset operators [1], [2] are probably the most widely used control operators for partial (delimited) continuations. The reset operator (the delimiter) is denoted as $\mathtt{reset}M$ (or $\langle M \rangle$ in their original notation), and the shift operator is $\xi c.M$ where the variable c is bound by this ξ. Although the operational semantics of shift/reset is defined via the CPS-transformation, we may consider the following reduction rules for shift/reset:

$$E[\mathtt{reset}(E'[\xi c.M])] \to E[\mathtt{reset}M\{c := \lambda u.\mathtt{reset}(E'[u])\}]$$

The continuation up to the closest `reset` is captured as a functional object $\lambda u.\mathtt{reset}(E[u])$ and can be used in the subsequent computation. Note that, unlike the standard (full) continuation generated by `call/cc`, the created continuation is a partial rest of computation (it is not the whole rest of computation like $\lambda u.E[E'[u]]$). Another important difference between delimited/full continuations is that the full continuation is abortive (not composable) while the delimited continuation can be composed. It is illustrated by the following example:

$$\begin{array}{rl} & \mathtt{reset}(1 + (\xi c.c(c(0)))) \\ \to & \mathtt{reset}(c(c(0)))\{c := \lambda u.\mathtt{reset}(1 + u)\} \\ \overset{*}{\to} & 1 + 1 + 0 \\ \overset{*}{\to} & 2 \end{array}$$

On the contrary, the term $1 + \mathbf{call/cc}\lambda k.k(k(k(0)))$ evaluates to 1, since the current continuation is aborted when k is applied to 0, and $k(k(k(0)))$ and $k(0)$ result in the same answer.

Note that, the shift/reset operators are *dynamic* in our sense, since $(\lambda x.\mathtt{reset}(xy))(\lambda z.\xi c.M)$ reduces to $\mathtt{reset}((\lambda z.\xi c.M)y)$, and $\mathtt{reset}((\mathtt{reset}\lambda x.\xi c.M)y)$ reduces to $\mathtt{reset}((\lambda x.\xi c.M)y)$. In both cases, the reset operator corresponding to the shift operator is different from the lexically determined one.

Remark In [1], the shift/reset operators are said to be *static* in comparison with Felleisen's *dynamic* operators. The functional object generated by an invocation of the shift operator is $\lambda u.\mathtt{reset}(E[u])$, while Felleisen's operator generates a functional object like $\lambda u.E[u]$. Suppose E contains another occurrence of shift. Then it is delimited by a fixed delimiter in the former case, and by an unknown outer delimiter in the latter case. This difference is the reason why Danvy and Filinski's operator enjoy a simple, elegant CPS-transformation.

A. Type System

We then try to give types to the shift/reset operators. In order to avoid the complications, we consider the simplest case only, namely, we assume that the types of the delimiters is unique. Let B_1 be this type. The type inference rules for shift/reset are given in Figure 3.

$$\frac{\Gamma, c : A \xrightarrow{\Delta'} B_1 \vdash M : B_1 \;;\; \Delta}{\Gamma \vdash \xi\ c.M : A \;;\; \Delta \cup \{B_1\}}$$

$$\frac{\Gamma \vdash M : B_1 \;;\; \Delta}{\Gamma \vdash \mathtt{reset}\ M : B_1 \;;\; \Delta - \{B_1\}}$$

Fig. 3. Type System for Shift/Reset

Note that, the use of the shift rule (the first rule) is restricted to type B_1. Hence, Δ and Δ' are either an empty set or the singleton $\{B_1\}$.

Murthy [9] introduced a rather heavy type system for the hierarchy of shift/reset's. He changed the function type $A \to B$ to $((A \to B) \to T_i) \to T_i$ where T_i is the type of i-th delimiter, that is, he annotated every function type by T_i, which complicates the type structures a bit, and his type system is not at all a simple extension of the simply typed lambda calculus.

We then have a reduction rule for shift/reset. Values and evaluation contexts are defined as usual.

$$E[\mathtt{reset}(E'[\xi c.M])] \quad \to \quad E[M\{c := \lambda u.\mathtt{reset}(E'[u])\}]$$

where $E' = F[\mathtt{reset}(F')]$ for no evaluation contexts F and F'.

B. CPS-transform

Danvy and Filinski already gave a CPS-transform for shift/reset as follows:

$$\begin{array}{rcl} [\![\xi c.M]\!] & \equiv & \lambda k.[\![M]\!]\{c := \lambda x k'.k'(kx)\}I \\ [\![\mathtt{reset}M]\!] & \equiv & \lambda k.k([\![M]\!]I) \end{array}$$

In fact, this CPS-transform is the only rigid semantics for the present shift/reset. At first look, this CPS-transform works well. However, the typability is not preserved at the Consider the term $\xi c.M$. If the types of c and M are $A \to B$ and B, then the whole term has type A. The type of c^* is $A \to B^*$, that is, $A^* \to (B^* \to *) \to *$. It follows that the bound variables x and k' in RHS of the CPS-transform of the `shift`-term have types A^* and $B^* \to *$. Since the type of k and k' in the RHS is $A^* \to *$, the type of kx has type $*$. But the term $k'(kx)$ does not typecheck.

On the contrary, our FMCPS can be defined for shift/reset so that the typing and the reduction are preserved. For the type-part, the transform is defined as follows:

$$\begin{array}{rcl} [\![A]\!] & \equiv & (A^* \to *) \to \\ & & (B_1^* \to *) \to \\ & & (A^* \to B_1^*) \to * \\ A^* & \equiv & A^* \ \text{ if } A \text{ is atomic} \\ (A \to B)^* & \equiv & A^* \to [\![B]\!] \end{array}$$

Note that, in this case, we have one more extra continuation parameter of type $A^* \to B_1^*$ which represents the partial continuation up to the closest delimiter.

For the term-part, the transform is defined as follows:

$$[\![x]\!] \;\equiv\; \lambda kpq.kx$$

$$\begin{aligned}
\llbracket \lambda x.M \rrbracket &\equiv \lambda kpq.k(\lambda xrd.\llbracket M \rrbracket r)\\
\llbracket MN \rrbracket &\equiv \lambda kpq.\llbracket M \rrbracket (\lambda m.\llbracket N \rrbracket (\lambda n.mnkpq)pq)pq\\
\llbracket \xi c.M \rrbracket &\equiv \lambda kpq.\llbracket M \rrbracket \{c := \lambda ak'p'q'.k'(qa)\}ppI\\
\llbracket \mathtt{reset} M \rrbracket &\equiv \lambda kpq.\llbracket M \rrbracket kkI
\end{aligned}$$

As is seen from the above definition, our FMCPS-tranform is a slight extension of Danvy-Filinski's CPS-transform. However, to preserve the typability we split the continuation paramters into three parts, and carefully combine these parameters.

We have the following theorem as desired:

Theorem 3: If $\Gamma \vdash M : A\ ;\ \Delta$ is derivable in the source language, then $\Gamma^* \vdash \llbracket M \rrbracket : \llbracket A \rrbracket$ is derivable in $\lambda^{\rightarrow}$ in case B_1 is atomic, and it is derivable in $\lambda^{\rightarrow,\mathbf{rec}}$ in a general case.

Moreover, if $M = N$ in the source language, then $\llbracket M \rrbracket = \llbracket N \rrbracket$ in the call-by-value semantics. *(End of Theorem)*

VI. CONCLUSION

In summary, we tried to fill the following two columns:

	Static	Dynamic
Exception	Nakano,Sato	This paper
Continuation	Griffin,Murthy	-
Partial Cont.	Kameyama	This paper

We give a type system, reduction rules, and an extended CPS-tranform. Although dynamic control operators have some bad behaviour in the most general case we can construct a good theoretical foundation if we have a certain restriction such as atomic exception types.

This paper mainly considers the type-theoretic analysis on dynamic control operators. A missing point of this paper is to study the logical significance of our type system (if any). As Thielecke showed in [13], it seems that the type system for dynamic control does not correspond to classical logic, but the detailed analysis is left for future work.

ACKNOWLEDGMENTS

The author would like to thank Olivier Danvy, Andrzej Filinski, Hayo Thielecke, and Amr Sabry for constructive comments. This work is supported in part by the Grant-in-Aid for Scientific Research by Ministry of Education, Japan, No. 13680411.

REFERENCES

[1] O. Danvy and A. Filinski, "Abstracting Control", *Proc. 1990 ACM Conference on Lisp and Functional Programming*, pp. 151-160, 1990.

[2] O. Danvy and A. Filinski, "Representing Control: a Study of the CPS Transformation", *Mathematical Structures in Computer Science* 2(4), pp. 361-391, 1992.

[3] T. Griffin, "A Formula-as-Types Notion of Control", *Conf. Record of 17th ACM Sympo. on Principles of Programming Languages*, pp. 47-58, 1990.

[4] P. de Groote, "A CPS-Translation of the $\lambda\mu$-Calculus", *Lecture Notes in Computer Science* **787**, pp. 85-99, 1994.

[5] M. Felleisen, "The Theory and Practice of First-Class Prompts", *Proc. 15th ACM Symp. on Principles of Programming Languages*, pp. 180-190, 1988.

[6] R. Hieb, R. Dybvig, and C. W. Anderson, "Subcontinuations", *Lisp and Symbolic Computation* 6, pp. 453–484, 1993.

[7] Y. Kameyama, "A Type-theoretic Study on Partial Continuations", Proc. IFIP International Conference on Theoretical Computer Science (IFIP TCS2000), Sendai, Japan, *Lecture Notes in Computer Science* **1872**, pp.489-504, 2000.

[8] Y. Kameyama and M. Sato, "Strong Normalizability of the Non-deterministic Catch/Throw Calculi", *Theoretical Computer Science*, to appear.

[9] C. Murthy, "Control Operators, Hierarchies, and Pseudo-Classical Type Systems: A-Translation at Work", *Proc. ACM Workshop on Continuations*, pp. 49–71, 1992.

[10] H. Nakano, "A Constructive Formalization of the Catch and Throw Mechanism", *Conf. Rec. IEEE Symposium on Logic in Computer Science*, pp. 82–89, 1992.

[11] M. Parigot, "$\lambda\mu$-Calculus: An Algorithmic Interpretation of Classical Natural Deduction", Proc. Intl Conf. on Logic Programming and Automated Reasoning, *Lecture Notes in Computer Science* **624** pp. 199–201, 1992.

[12] M. Sato, "Intuitionistic and Classical Natural Deduction Systems with the Catch and the Throw Rules", *Theoretical Computer Science* 175 (1), pp. 75-92, 1997.

[13] H. Thielecke, "Comparing Control Constructs by Typing Double-Barrelled CPS Transforms", *Proc. Third ACM SIGPLAN Workshop on Continuations*, pp. 47-54, 2001.

[14] P. Thiemann, "Cogen in Six Lines", *International Conference on Functional Programming*, pp. 180-189, 1995.

Towards a Systematic Reuse Based on both General Purpose and Domain-Specific Approaches

H. Hadjami Ben Ghezala
National School Of Computer Science (ENSI)
Campus Universitaire - La Manouba 2010 Tunisia
henda.BG@cck.rnu.tn

R. Beltaifa Hajri
National School Of Computer Science (ENSI)
Campus Universitaire - La Manouba 2010 Tunisia
raoudha.beltaifa@ensi.rnu.tn

Abstract-- **Several advances towards systematic software reuse have recently been reported with library systems, classification techniques, the creation and distribution of reusable components, reuse support environments, and corporate reuse programs. Despite these efforts, we argue that reuse has not delivered yet on its promise to significantly increase productivity and quality. We therefore suggest a systematic reuse approach, which supports more formal and disciplined practice of reuse. The proposed approach is based on: (1) a domain meta-model incorporating the necessary knowledge to carry out reuse in specific domains; (2) a component meta-model providing the components with abstract descriptions to facilitate their reuse; (3) a search process to support a strategic and intentional component retrieval; and (4) a set of certified reuse libraries accessible through Internet and Intranet networks, from reusable components could be retrieved. We also report a formalisation of this approach using the RAISE specification language, with C++ source code generated from formal specifications.**

Keywords—Systematic reuse, General purpose approaches, Domain-specific approaches.

I. INTRODUCTION

Software reuse has been actively studied since the 1968 NATO software engineering conference [8]. Over the years many software reuse approaches have been developed evolving from ad-hoc to systematic. As reported in [12], the reuse of code in an ad hoc way was the beginning of reuse practice, which consists in the reuse of subroutines in 1960's and reuse of modules in the 1970's. In the 1990's, the reuse of components from reuse libraries took place. Nowadays, and from the 2000's, reuse begins to be practical in product lines approaches.

Software reuse approaches can be separated in two major categories: *general purpose* and *domain-specific*.

General purpose approaches encapsulate reuse activities independently of the application domain. They include the reuse of components across all or most applications, which we call 'generic components'.

A general purpose reuse approach captures two major activities: *design for reuse and design with reuse.* Design for reuse activity consists in identifying and organising reusable components in a reuse library, and design by reuse activity consists in searching, adapting and integrating reusable components in order to build new systems. These approaches advise building libraries of components that are generally needed by the software industry. Examples of general purpose libraries include C libraries, FORTRAN math libraries, and general object hierarchies as provided with C++ and Smalltalk.

The problems associated with general purpose reuse is that software components are typically very information-rich, which makes it difficult to characterise them, match them and capture their relevant properties. Also, the libraries are typically large invoking problems in components' classification and search. General purpose libraries contain one type of components, most frequently code, abstract data types and documents; hence, the need of reusing different types of components provokes searching from different libraries, which consequently leads to the use of different search processes and then it is time consuming and it requires intensive developer knowledge.

Domain-specific approaches consider reuse activities in a specific application domain; they include the reuse of components that are specialised for a given family of software systems or application domain.

A domain-specific reuse approach captures two major engineering activities: domain engineering and application engineering. Domain engineering activity consists in the domain analysis and design for creating a library of reusable components. The application engineering activity consists in the application analysis and design by association with the domain analysis and design. The application is built by the integration of the components retrieved from the domain-specific reuse library.

In domain-specific reuse libraries problems with selecting, eventually adapting, and integrating components are still remaining. The components and domain knowledge are difficult to accede which involves search and adaptation problems.

As for the search in general purpose libraries, if different components are needed then different libraries should be acceded, domain-specific libraries may contain different types of components but all of them are associated with a particular domain, consequently reuse in general needs accessing different libraries.

For both general purpose and domain-specific libraries components' relationships are frequently neglected, therefore more problems in selecting and integrating components are involved. Also, problems with selecting an appropriate level of abstraction for describing the reusable components are involved.

To help in alleviating the problems cited above, we propose a systematic reuse approach capturing the advantages of both general purpose and domain-specific approaches and improving their use.

In our approach, we propose to:

- make possible the search for different components located in different libraries. These components may have different natures, e.g., specification, design and code, and may be associated with general purpose or domain-specific libraries.
- provide the necessary information about reusable components, reuse libraries and domain knowledge. Domain knowledge include its concepts, static, functional and dynamic properties which could be captured in a domain model. Components' information includes their important attributes, functionality, guidelines and relationships. Reuse libraries information include their locations, access possibilities and so on. These information help in the components reuse which may be included in a general purpose or domain-specific libraries.
- present in an explicit way the connectors and relationships between software components located in different reuse libraries.
- have a search process which considers different reuse libraries, providing different strategies satisfying the developer intentions and requirements.

The purpose of this paper is to present the systematic reuse approach we propose. In section 2 we present the motivations and the approach features and elements, which include a domain meta-model, a component meta-model and a search process. In section 3, we present the domain meta-model capturing the domain properties necessary in a reuse field. In section 4, we present the component meta-model encapsulating reusable components properties necessary for their reuse. In section 5, we present the search process in terms of its intentional and strategic features. Finally, in section 6 we conclude with a summary on the proposed reuse approach and we present the intended future work.

II. A SYSTEMATIC REUSE APPROACH

By the systematic reuse approach we propose, we would like to achieve a successful reuse practice. The basic ideas behind this approach are that: (1) it considers general purpose and domain-specific reuse approaches; (2) it makes possible the reuse of components of different natures and associated with different domains; (3) it facilitates components' reuse by describing them in an abstract way, which encapsulates the components reuse guidelines, their static and semantic properties, and their relationships with others; (4) it considers that domain knowledge facilitates more the reuse in a specific-domain; (5) it facilitates reuse practice by making possible the reuse from different libraries, which could be accessible through Internet and Intranet networks.

We believe that a systematic reuse approach to be practical and successful should be models based and formally specified, therefore the approach we propose is based on :

- a domain meta-model capturing an abstract description of the domain knowledge. Domain information (knowledge) can be used to aid in component retrieval and more general kinds of component processing.
- a component meta-model encapsulating an abstract description of the features characterising software components in a way that facilitates their reuse and retrieval.
- a set of certified reuse libraries including specific-domain and general purpose libraries. These libraries could be accessible through an Internet and/or Intranet networks. Reuse from different libraries increases the chance to find the components in need. By these libraries we may reuse components of different natures (code, design, specification, test case, application generator, etc.) and different granularities (class, pattern, framework, etc.).
- a unified search process providing different strategies of search and different formats of queries. Search in different libraries may be customised according to the user intentions and preferences, and the system requirements.
- a formal description, which makes the approach abstract and could be implemented independently from a specific language.

In the next sections, we present the basic elements required in the approach we propose highlighting their features and their utility in a systematic reuse.

III. DOMAIN META MODEL

To facilitate the component search in the scope of its domain, domain knowledge represents a good guide source.

In software engineering, a domain knowledge is usually captured in a domain model. Different domain models may include different domain features and may neglect others.

Seeing that in the systematic reuse approach we propose different domains could be used then, a specific domain model can be unable to describe requirements of other domains. Therefore, we propose a domain meta-model capturing the domain features/requirements independently from a specific one.

To define this meta-model, we studied a set of available domain models that are generated by some well-established domain analysis methods: FODA (Feature-Oriented Domain Analysis), DSSA (Domain-specific Software Architecture), The Synthesis Domain analysis Method and JODA (Object-Oriented Domain Analysis Method).

A. *Feature-Oriented Domain Analysis, FODA*

The Feature-Oriented Domain Analysis (FODA) method was developed at the Software Engineering Institute (SEI) [4]. FODA focuses on identifying features that characterise a domain. Applications in a domain provide several capabilities. These capabilities are modeled in FODA as features.

By the study of the FODA model, we noticed that the capabilities are not sufficient to characterise a domain for a reuse goal, see tableI. In one side, the terminology associated with the domain is not considered. A domain terminology or glossary facilitates the use of the domain terms and alleviates the ambiguity between different domains. In an other side, FODA domain model neglects the possible rules and constraints imposing specific technologies or specific rules for the applications associated with the domain.

B. *Domain-specific Software Architecture, DSSA*

The DSSA domain models are developed by the Defence Advanced Research Projects Agency (DARPA) for command and control applications. These domain models are function, dynamic, and object models.

As for the FODA domain model, the one associated with DSSA suffers from the absence of the domain terminology and its possible rules and constraints, see tableI. Further, it neglects the domain dependencies with the environment and with other domains which represent a major feature of a domain in the reuse area.

C. *The Synthesis Domain analysis Method*

Domain analysis is a part of the synthesis approach that was developed by the Software Productivity Consortium [13].

The Synthesis domain definition is represented in an informal manner. This makes difficult its use for a reuse goal, see tableI. Informal description is always not clear and may generate ambiguities. The domain features captured in its static, functional and dynamic properties should be defined in formal models to be well described.

D. *Object-Oriented Domain Analysis Method (JODA)*

The Object-Oriented Domain analysis Method (JODA) was developed by the Joint Integrated Avionics Working Group (JIAWG) [3]. JODA defines domain models using the Coad/Yourdon object-oriented analysis technique. Domain models are defined in terms of objects, their attributes and their services and relationships with other objects in the domain models.

JODA is a well defined approach, it presents the domain model by formal descriptions. It is based on different diagrams to represent the domain features. Different domain properties are considered. However, the domain rules and constraints were neglected, see tableI. These rules may be useful to enrich the domain knowledge and to guide the reuse in a specific one.

We encapsulate the major important domain aspects inside the domain meta-model, see Fig1. These aspects are described as follows:

- *Domain static aspect,* represent the domain objects and their relationships.
- *Domain functional aspects* represent the set of services, operations and functionalities that is common across various applications in the domain, expressing the domain requirements.
- *Domain dynamic aspects* may be described by a set of events and scenarios.
- *Domain rules and constraints,* include structuring rules (constraints about the structure of the applications), potential implementation constraints (imposing specific technologies), etc..
- *Domain terminology* defines a glossary of terms for the domain.
- *Domain dependencies* represent domain relationships with other domains and its dependencies with the environment.

IV. COMPONENT META MODEL

To facilitate the search for components, their understanding and their use, component semantic and structural properties should be accessible. Also, searching for components in a specific library requires the library knowledge and the reuse in a specific domain requires the domain knowledge.

All the information necessary for the component search, understanding and use should be formally represented to facilitate the reuse practice. A certain component model may be insufficient to represent all these information. Therefore, we propose a component meta-model, see Fig2, which represents a reusable component at different scopes : its features at different views, the associated library features and eventually the associated domain features. The meta-model information are encapsulated as follows:

Component information, which we define as: (1) descriptive information providing knowledge of what the component does; (2) information to facilitate when and how to use it; and (3) information to facilitate its retrieval. More specifically, a reusable component is a software artefact which:

- is identified by its name, its structure (package, function, pattern, framework, architecture, etc.), the notation in which it is written (e.g., Ada, C++, UML, etc.), its nature (specification, design, code, test case, etc.), its type (process, product or method).
- can be qualified by its size, its reuse frequency and its modification percentage (if it is adaptable), and can have more significant structural properties helping the user to retrieve and understand it. The corresponding

TABLE I
DOMAIN MODELS FEATURES

	static aspects	*functional aspects*	*dynamic aspects*	*rules and constraints*	*Terminology*	*Dependencies*
FODA	+	+	+	-	-	+
DSSA	+	+	+	-	-	-
Synthesis	-	-	-	-	+	-
JODA	+	+	+	-	+	+

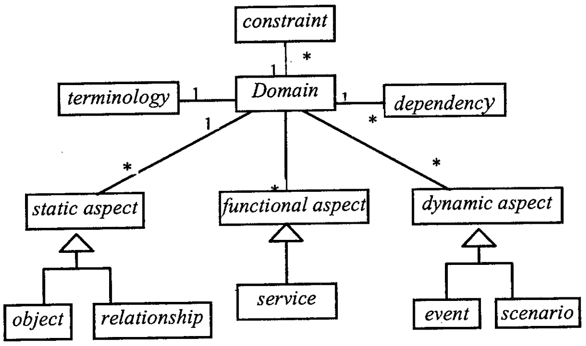


Fig.1: Domain Meta Model

features are the objectives of the component, an example of its, and comments and guidelines.

- must provide services. These services represent the functional properties of the component and are characterised by:
  - a function name, usually a verb describing the service (e.g. add, move, define, etc.)
  - a function type, which may be *observer* or *generator.* The function type is *observer* if the function does not modify the objects to which it is applied but simply gives some information about them without changing their state.
  - a signature, which represents the input and output types of a function (service)
  - pre-conditions, which are the conditions that should be satisfied in order to execute the function,
  - post-conditions, which give a logical representation of the result of executing the function assuming its pre-conditions are all satisfied.
  - a function description, which describes the service (generally text but may include figures or other graphical representations).
- has reuse guidelines which indicate how to reuse the component. These guidelines thus make reuse easier and faster and generally encourage reuse of a component.
- can be a product (building block) or a process (generative) component.
- can be reused as a white box or a black box. The latter applies to components which can be integrated into new systems without any modification.

Library information, represents the library features that should help the component search. These information are abstracted by:

- the library name.
- the library localisation which may be a URL address in Internet or Intranet.
- Library author which the library creator.

Domain information, represent the domain features which are encapsulated in the proposed domain meta-model, see Fig1.

V. SEARCH PROCESS

The problem of component retrieval from reuse libraries has long been considered to be the central technical issue surrounding the task of software library construction [11][5][7]. However, the recent movement towards general purpose and domain-specific software reuse strategies has simplified the retrieval problem by narrowing the scope of the universe of discourse. However, there remains desire to develop software retrieval strategies suited for different (general purpose and domain-specific) and large scale reuse libraries.

In such environment, there are obvious advantages to having many retrieval methods available, the major advantage being that of maximising assistance to the developer in locating components of interest. As well, having different search strategies improve the component retrieval by providing a customisable search depending on the user intentions, preferences and requirements.

Therefore, the search process we propose, is suited for different reuse libraries including general purpose and domain-specific ones. This process is based on different search methods and driven by the developer intentions and requirements, see Fig 3.

By Fig 3, we show that the search process is based on the information/knowledge associated with the available reuse libraries them selves (name, location, etc.), the reusable components, and the domains associated with the domain-specific libraries. The component information/knowledge are encapsulated in the component meta-model described in section 4, the domain ones are encapsulated in the domain meta-model presented in section 3.

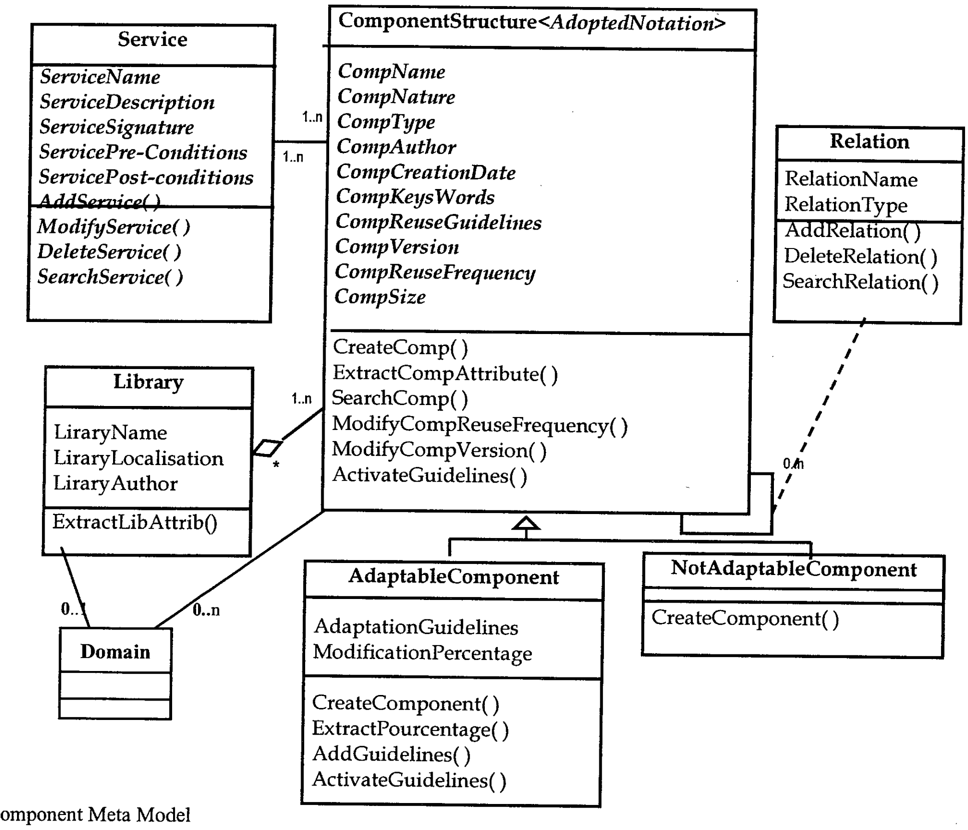


Fig 2: A Component Meta Model

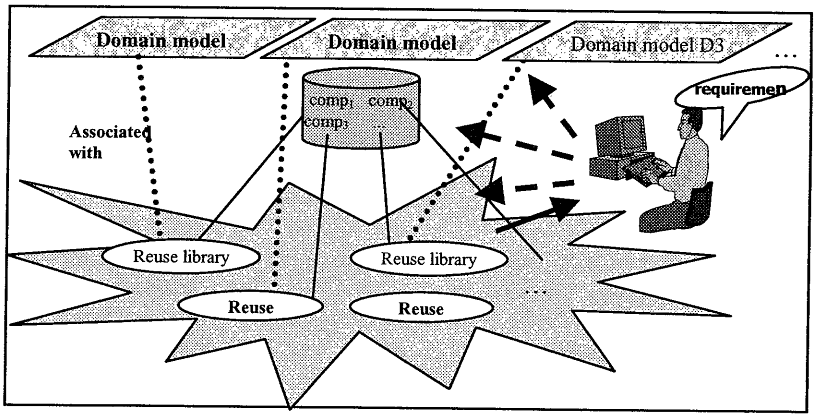


Fig3: Components Retrieval from Reuse Libraries

In order to provide various possibilities of components retrieval which are driven by the developer intentions and requirements and guided by the search strategies we used the process model *Map* to represent the search process [2][10].

A map is a process model in which a non-deterministic ordering of intentions and strategies has been included. It is a labelled directed graph with intentions as nodes and strategies as edges. The directed nature of the graph shows which intentions can follow a preceding one.

We assume the search process to be intention-oriented. At any moment, the developer has an intention, a goal in mind that he wants to fulfil. To take this feature into account the map identifies the set of intentions that have to be achieved in order to satisfy the search requirements at hand.

In Fig 4, we present the map associated with the search process we propose. In this map we present the search intentions and the different search strategies.

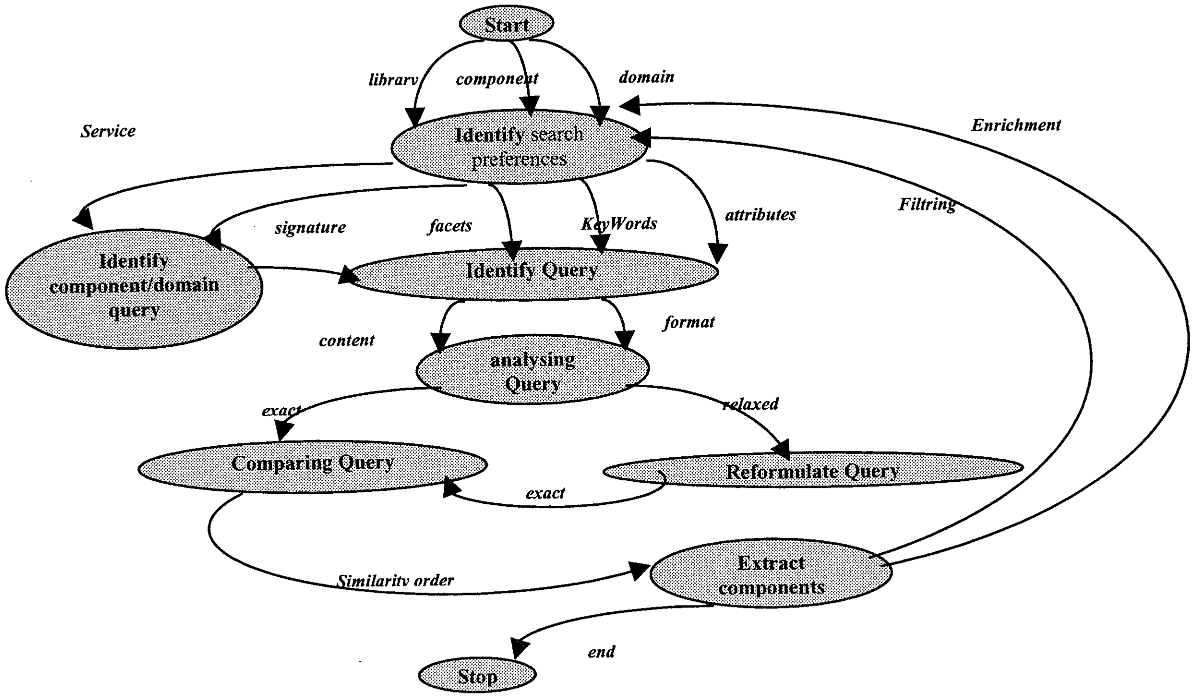


Fig 4: Search Process Map

As presented in figure 4, the developer may use the domain strategy, the component strategy and/or the library component, domain and/or library information/knowledge to search for the components in need.

If the developer needs components associated with a specific domain, then he could use the *domain strategy* in order to focus the search on a specific domain.

If the developer has information about the library where he can find the components in need, then he could follow the *library strategy*. But, if the developer is searching for components associated with a general purpose then he could choose the *component strategy*.

To identify his query, the developer may have the intention to search by component or domain properties. In fact, he could choose the *service* or *signature strategy* to achieve his intention *"identify component/domain query"*. But, he could use any of the *facets, key words or attributes strategies* to identify a query, *identify query intention*, associated with a component, a library or a domain.

VI. CONCLUSION

Several advances towards systematic software reuse have recently been reported with library systems, classification techniques, the creation and distribution of reusable components, reuse support environments, and corporate reuse programs. Despite these efforts, we argue that reuse has not delivered yet on its promise to significantly increase productivity and quality. In this paper, we proposed a systematic reuse approach, which captures the advantages of the existing reuse approaches and provides possibilities of alleviating their limitations. This approach is based on a set of meta-models and a formal description. Seeing the advantages of meta-modelling and formalisation we believe that this approach may improve reuse practice. By the meta-models we considered and we concluded that the components, library and domain knowledge are important and necessary information for the reuse practice. Even, their formalisation is necessary to define a better practice and specially a better component retrieval.

Formalisation of the approach is not defined in this paper. But, we already defined it using the RAISE Specification Language [9], then we generated automatically the associated C++ code, by using the RAISE translation tool [1]. Our current work is the implementation and the evaluation of the proposed approach, which is based on code, pattern, method chunk and process reuse libraries.

REFERENCES

[1] C. George, "Raise Tools User Guide", Technical Report, N227, UNU/IIST, 2001.

[2] G. Grosz, C. Rolland, and Benjamin, "Modelling and Enginnering the Requirements Engineering Process: an Overview of the NATURE Approach", Requirements Engineering Journal, pp: 115-131, 1997.

[3] R. Holibaugh, "Joint Integrated Avionics Working Group (JIAWG) Object-Oriented Domain Analysis Method (JODA)", Special Report, CMU/SEI-92-SR-3.

[4] K. Kang, S. Cohen, J. Hess, W. Novak, and A. Peterson, "Feature-Oriented Domain Analysis (FODA): Feasibility Study", CMU/SEI-90-TR-021, ww.sei.cmu.edu/publications/documents/90.reports/90.tr.021.html

[5] C.W. Krueger, "Software Reuse", ACM Computing Surverys, 24(2), pp: 131-183, June 1992.

[6] M.D. Mcllroy, "Mass Produced Software Components", Proceedings of the 1969 NATO Conference of Software Engineering, 1969.

[7] H.Mili, F. Mili, A.Mili, "Reusing Software: Issues and Research Directions", IEEE Trans. on Software Engineering, 21(6), pp: 528-561, June 1995.

[8] P. Nauer, B. Randell, "Software Engineering", Report on Conference by the NATO Science Committee, NATO Science Affairs Division, Brussels, Belgium, 1968.

[9] The RAISE Language Group, "The RAISE specification Language", Prentice Hall International, 1992.

[10] C. Rolland, N. Prakash, all, "A Multi-Model View of Process Modelling", RE Journal, 1999.

[11] M. Rosario Girardi, "Classification and Retrieval of Software Through their Descriptions in Natural Language", PhD in Computer Science, Geneve University, 1995.

[12] Software Engineering Institute, "Essentials of Successful Product Line Practice", slides, October 1999.

[13] Software Productivity Consotium, "Reuse-Driven Software Processes Guidebook", Version 2.0.3, December 1993.

Runtime Behavior of Conversion Interpretation of Subtyping

Yasuhiko Minamide
Institute of Information Sciences and Electronics
University of Tsukuba
and
PRESTO, JST
minamide@score.is.tsukuba.ac.jp

Abstract— **A programming language with subtyping can be translated into a language without subtyping by inserting conversion functions. Previous studies on this interpretation showed only extensional correctness of this translation. We study runtime behavior of translated programs and show that this translation preserves execution time and stack space within a factor determined by types in a program.**

Type systems with subtyping can express information obtained by various program analyses such as control flow analyses. Our results ensure safety of implementation based on conversion interpretation utilizing types obtained by such program analyses.

I. INTRODUCTION

A programming language with subtyping can be translated into a language without subtyping by inserting conversion functions. Previous studies on this interpretation showed only extensional correctness of this translation [1], [2]. In this paper we study runtime behavior of conversion interpretation of subtyping in call-by-value evaluation and show that this translation preserves execution time and stack space with a factor determined by types in a program if subtyping relation is partial order.

The translation of conversion interpretation changes runtime behavior of programs in several respects. It inserts conversion functions and may increase the total amount of closures allocated during execution. It translates tail-calls into non-tail-calls and thus may increase stack space usage. Even though the translation causes these changes of runtime behavior, execution time and stack space usage are preserved asymptotically. This contrasts with type-directed unboxing of Leroy where both time and space complexity are not preserved [3].

Type systems with subtyping can be used to express information obtained by various program analysis such as control flow analyses [4]. One strategy of utilising types obtained by program analyses is to adopt conversion interpretation of subtyping. For example, it is possible to choose optimized representation of values based on types and to insert conversion functions as type-directed unboxing of polymorphic languages [5]. In order to adopt this kind of compilation method we need to show that conversion interpretation is safe with respect to performance. The results in this paper ensure safety with respect to execution time and stack space.

The proof is based on the method of logical relations and closely related to the proof that the refined type-directed unboxing by Minamide and Garrigue preserves execution time asymptotically [3] . The structure of the proof for stack space is almost the same as that for execution time. This comes from the similarity between the operational semantics profiling time and stack space. As long as we know, this is the first proof concerning stack space based on the method of logical relations.

This paper is organized as follows. We start with review of conversion interpretation and runtime behavior of translated programs. In Section III we define the language we will use in the rest of the paper and introduce conversion interpretation formally. We prove that conversion interpretation preserves stack space and execution time in Section IV and Section V. The paper ends with a section of conclusions.

II. REVIEW OF CONVERSION INTERPRETATION

We review conversion interpretation of subtyping and intuitively explain that the interpretation preserves stack space and execution time if subtyping relation is partial order. The conversion interpretation is a translation from a language with subtyping into a simply typed language without subtyping. The idea is inserting a conversion function (or coercion) where subsumption rule is used. If the following subsumption rule is used in typing derivation,

$$\frac{\Gamma \vdash M : \tau \quad \tau \leq \sigma}{\Gamma \vdash M : \sigma}$$

the conversion function $\mathsf{coerce}_{\tau\leq\sigma}$ of type $\tau \rightarrow \sigma$ is inserted and we obtain the following term.

$$\mathsf{coerce}_{\tau\leq\sigma}(\mathsf{M})$$

Coercion $\mathsf{coerce}_{\tau\leq\sigma}$ is inductively defined on structure of types. If there are two base types bigint and

---

This work was presented at the 13th International Workshop on the Implementation of Functional Languages, 2001

int for integers where int is a subtype of bigint, we need to have a coercion primitive int2bigint of type int → bigint. A conversion function for function types is constructed as follows.

$$\lambda f.\lambda x.\mathsf{coerce}_{\tau_2 \leq \sigma_2}(\mathsf{f}(\mathsf{coerce}_{\sigma_1 \leq \tau_1}(\mathsf{x})))$$

This is a coercion from $\tau_1 \to \tau_2$ to $\sigma_1 \to \sigma_2$.

We show that this interpretation of subtyping is safe with respect to execution time and stack space usage if the subtyping relation is partial order. Intuitively, this holds because only finite number of coercions can be applied to any value. If a subtyping relation is not partial order, i.e., there exists two base types τ and σ such that $\tau \leq \sigma$ and $\sigma \leq \tau$, we can construct counter examples for both stack space and execution time easily.

A counter example for execution time is the following translation of term M with τ,

$$\mathsf{coerce}_{\sigma \leq \tau}(\mathsf{coerce}_{\tau \leq \sigma}(\ldots(\mathsf{coerce}_{\sigma \leq \tau}(\mathsf{coerce}_{\tau \leq \sigma}(\mathsf{M})))))$$

where $\tau \leq \sigma$ and $\sigma \leq \tau$. The execution time to evaluate the coercions in the translation depends on the number of the coercions and cannot be bounded by any constant.

The conversion interpretation translates tail-call applications into non-tail-call applications. Let us consider the following translation of application xy.

$$\mathsf{coerce}_{\tau \leq \sigma}(\mathsf{xy})$$

Even if xy is originally at a tail-call position, after translation it is not at a tail-call position. Thus it is not so straightforward to show conversion interpretation preserves stack space asymptotically. Actually, if we have equivalent types, we can show a counter example. Let us consider the following program where types `A` and `B` are equivalent.

```
                    (* f: int * A -> A *)
fun f (0, x : A) = x
  | f (n, x : A) = g (n-1, x)
                    (* g: int * A -> B *)
and g (n, x : A) = f (n, x) : B
```

This program contains only tail-calls, and thus requires only constant amount of stack space. By inserting conversion functions we obtain the following program.

```
fun f (0, x : A) = x
  | f (n, x : A) = B2A (g (n-1, x))
and g (n, x : A) = A2B (f (n, x))
```

For this program, evaluation of `f n` requires stack space proportional to `n` since both the applications of `f` and `g` are not tail-calls.

In order to preserve time and stack space complexity, it is essential that the subtyping relation is partial order. This ensures that there is no infinite chain of types: $\tau < \tau_1 < \tau_2 \ldots$ Thus only a finite number of conversions can be applied to any value.

III. LANGUAGE AND CONVERSION INTERPRETATION

In this section we introduce a call-by-value functional language with subtyping and its conversion interpretation. We consider a call-by-value functional language with the following syntax.

$$\begin{array}{rcl} V & ::= & x \mid \bar{i} \mid \underline{i} \mid \lambda x.M \mid \mathtt{fix}^n\, x.\lambda y.M \\ M & ::= & V \mid M\,M \mid \mathtt{let}\ x = M\ \mathtt{in}\ M \end{array}$$

There are two families of integers: $\bar{i}$ and $\underline{i}$ are integer values of types bigint and int respectively. The language includes bounded recursive functions where $\mathtt{fix}^n\, x.\lambda y.M$ is expanded at most n times [6]. Any closed program with usual recursive functions can be simulated by bounded recursive functions.

For this language we consider a simple type system extended with subtyping. The types of the language are defined as follows.

$$\tau ::= \mathsf{bigint} \mid \mathsf{int} \mid \tau \to \tau$$

We consider two base types bigint and int where int is a subtype of bigint. A metavariable σ is also used to denote a type. The subtyping relation $\tau_1 \leq \tau_2$ is given by the following three rules.

$$\tau \leq \tau \qquad \mathsf{int} \leq \mathsf{bigint} \qquad \frac{\sigma_1 \leq \tau_1 \quad \tau_2 \leq \sigma_2}{\tau_1 \to \tau_2 \leq \sigma_1 \to \sigma_2}$$

The rule for transitivity is not included here because it can be derived from the other rules for this subtyping relation. We write $\tau < \sigma$ if $\tau \leq \sigma$ and $\tau \not\equiv \sigma$. It is clear that the subtyping relation is partial order. The typing judgement has the following form:

$$\Gamma \vdash M : \tau$$

where Γ is $x_1 : \tau_1, \ldots, x_n : \tau_n$. The rules of the type system are defined as follows.

$$\Gamma \vdash \bar{i} : \mathsf{bigint} \qquad \Gamma \vdash \underline{i} : \mathsf{int} \qquad \frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}$$

$$\frac{\Gamma \vdash M_1 : \tau_1 \to \tau_2 \quad \Gamma \vdash M_1 M_2 : \tau_2}{\Gamma \vdash M_1 M_2 : \tau_2}$$

$$\frac{\Gamma, x : \tau_1 \vdash M : \tau_2}{\Gamma \vdash \lambda x.M : \tau_1 \to \tau_2} \qquad \frac{\Gamma \vdash M : \sigma \quad \sigma \leq \tau}{\Gamma \vdash M : \tau}$$

$$\frac{\Gamma, y : \tau_1 \to \tau_2, x : \tau_1 \vdash M : \tau_2}{\Gamma \vdash \mathtt{fix}^n\, y.\lambda x.M : \tau_1 \to \tau_2}$$

We consider a standard natural semantics for this language. A judgement has the following form: $M \Downarrow$

$$V \Downarrow V \qquad \frac{M_1 \Downarrow V_1 \quad M_2[V_1/x] \Downarrow V}{\mathtt{let}\ x = M_1\ \mathtt{in}\ M_2 \Downarrow V}$$

$$\frac{M_1 \Downarrow \lambda x.M \quad M_2 \Downarrow V_2 \quad M[V_2/x] \Downarrow V}{M_1 M_2 \Downarrow V}$$

$$\frac{M_1 \Downarrow \mathtt{fix}^{k+1}\, y.\lambda x.M \quad M_2 \Downarrow V_2 \quad M[\mathtt{fix}^{k}\, y.\lambda x.M/y][V_2/x] \Downarrow V}{M_1 M_2 \Downarrow V}$$

Fig. 1. Operational semantics

V. The rules are given in Fig. 1. When the recursive function $\mathtt{fix}^{k+1}\, y.\lambda x.M$ is applied, the bound of the recursive function is decremented.

We consider the following target language for translation. The only extension is the coercion function $\mathsf{int2bigint}$ from int into bigint.

$$W \ ::= \ x \mid \underline{i} \mid \bar{i} \mid \lambda x.N$$

$$N \ ::= \ W \mid N\,N \mid \mathtt{let}\ x = N\ \mathtt{in}\ N \mid \mathsf{int2bigint(N)}$$

The operational semantics and type system of the language is almost the same as those for the source language. The rule of subsumption is excluded from the type system. The typing rule and evaluation of coercion are defined as follows.

$$\frac{N \Downarrow \underline{i}}{\mathsf{int2bigint(N)} \Downarrow \bar{\mathsf{i}}} \qquad \frac{\Gamma \vdash N : \mathsf{int}}{\Gamma \vdash \mathsf{int2bigint(N)} : \mathsf{bigint}}$$

The conversion interpretation is defined inductively on typing derivation of a program. The translation $[|\Gamma \vdash M : \tau|]$ is given in Fig. 2. Coercion used in the translation is defined as follows.

$$\begin{aligned} \mathsf{coerce}_{\tau \le \tau}(\mathsf{M}) &= M \\ \mathsf{coerce}_{\mathsf{int} \le \mathsf{bigint}}(\mathsf{M}) &= \mathsf{int2bigint(M)} \\ \mathsf{coerce}_{\tau_1 \to \tau_2 \le \sigma_1 \to \sigma_2}(\mathsf{M}) &= \\ \mathtt{let}\ x = M\ \mathtt{in}\ \lambda y.&\mathsf{coerce}_{\tau_2 \le \sigma_2}(\mathsf{x}(\mathsf{coerce}_{\sigma_1 \le \tau_1}(\mathsf{y}))) \end{aligned}$$

It should be remarked that $\mathsf{coerce}_{\tau \le \tau}(\mathsf{M})$ must be not only extensionally equivalent to M, but also intensionally equal to M. If we adopt $(\lambda x.x)M$ for $\mathsf{coerce}_{\tau \le \tau}(\mathsf{M})$, execution time and stack space are not preserved.

We define two measures, $\lfloor \tau \rfloor$ and $\lceil \tau \rceil$, of types as follows.

$$\begin{aligned} \lfloor \mathsf{int} \rfloor &= 0 \\ \lfloor \mathsf{bigint} \rfloor &= 1 \\ \lfloor \tau_1 \to \tau_2 \rfloor &= \lceil \tau_1 \rceil + \lfloor \tau_2 \rfloor \end{aligned}$$

$$\begin{aligned} \lceil \mathsf{int} \rceil &= 1 \\ \lceil \mathsf{bigint} \rceil &= 0 \\ \lceil \tau_1 \to \tau_2 \rceil &= \lfloor \tau_1 \rfloor + \lceil \tau_2 \rceil \end{aligned}$$

It is clear that $\sigma < \tau$ implies $\lfloor \sigma \rfloor < \lfloor \tau \rfloor$ and $\lceil \sigma \rceil > \lceil \tau \rceil$. Since $\lfloor \tau \rfloor$ and $\lceil \tau \rceil$ are non-negative integers, we also obtain the following properties.

$$\begin{aligned} \tau_n < \ldots < \tau_1 < \tau_0 \ &\Rightarrow\ \lfloor \tau_n \rfloor < \ldots < \lfloor \tau_1 \rfloor < \lfloor \tau_0 \rfloor \\ &\Rightarrow\ n \le \lfloor \tau_0 \rfloor \end{aligned}$$

$$\begin{aligned} \tau_0 < \tau_1 < \ldots < \tau_n \ &\Rightarrow\ \lceil \tau_0 \rceil > \lceil \tau_1 \rceil > \ldots > \lceil \tau_n \rceil \\ &\Rightarrow\ n \le \lceil \tau_0 \rceil \end{aligned}$$

From the property we can estimate the maximum number of conversions applied a value of τ.

$$\mathsf{coerce}_{\tau_{n-1} \le \tau_n}(\ldots(\mathsf{coerce}_{\tau_0 \le \tau_1}(\mathsf{V}))$$

Intuitively, this is the property which ensures that conversion interpretation preserves execution time and stack within a factor determined by types in a program.

IV. PRESERVATION OF STACK SPACE USAGE

We show that coercion interpretation of subtyping preserves stack space within a factor determined by types occurring typing derivation. We prove this property by the method of logical relations as we did for type-directed unboxing [3].

First we extend the operational semantics to profile stack space usage. The extended judgement has the following form: $M \Downarrow^n V$ where n models the size of stack space required to evaluate M to V. The extended rules are given in Fig 3. We write $e \Downarrow^n$ if $e \Downarrow^n v$ for some v and $e \Downarrow^{\le n}$ if $e \Downarrow^m$ for some $m \le n$. This semantics is considered to model evaluation by an interpreter: $M \Downarrow^n V$ means that a standard interpreter requires n stack frames to evaluate M to V. This semantics and the correspondence to a semantics modelling evaluation after compilation is discussed in [7]: the ratio to the stack space used by compiled executable code is bounded by the size of a program.

The main result of this section is that the conversion interpretation preserves stack space within a factor determined by the sizes of types appearing in a program.

Theorem 1: Let C be a positive integer such that $C > \lfloor \sigma \rfloor$ for all σ appearing in the derivation of $\emptyset \vdash M : \tau$. Let $[|\emptyset \vdash M : \tau|] = N$.

If $M \Downarrow^n V$, then $N \Downarrow^{\le Cn} W$ for some W.

$$
\begin{array}{rcl}
[|\Gamma \vdash x : \tau|] & = & x \\
[|\Gamma \vdash \lambda x.M : \tau_1 \to \tau_2|] & = & \lambda x.[|\Gamma, x : \tau_1 \vdash M : \tau_2|] \\
[|\Gamma \vdash \mathtt{fix}^n\, y.\lambda x.M : \tau_1 \to \tau_2|] & = & \mathtt{fix}^n\, y.\lambda x.[|\Gamma, y : \tau_1 \to \tau_2, x : \tau_1 \vdash M : \tau_2|] \\
[|\Gamma \vdash M_1 M_2 : \tau_2|] & = & [|\Gamma \vdash M_1 : \tau_1 \to \tau_2|][|\Gamma \vdash M_2 : \tau_1|] \\
[|\Gamma \vdash M : \tau|] & = & \mathsf{coerce}_{\sigma \le \tau}([|\Gamma \vdash \mathsf{M} : \sigma|])
\end{array}
$$

Fig. 2. Conversion interpretation

$$
V \Downarrow^1 V \qquad \frac{M_1 \Downarrow^m V_1 \quad M_2[V_1/x] \Downarrow^n V}{\mathtt{let}\ x = M_1\ \mathtt{in}\ M_2 \Downarrow^{\max(m+1,n)} V}
$$

$$
\frac{M_1 \Downarrow^l \lambda x.M \quad M_2 \Downarrow^m V_2 \quad M[V_2/x] \Downarrow^n V}{M_1 M_2 \Downarrow^{\max(l+1,m+1,n)} V}
$$

$$
\frac{M_1 \Downarrow^l \mathtt{fix}^{k+1}\, y.\lambda x.M \quad M_2 \Downarrow^m V_2 \quad M[\mathtt{fix}^k\, y.\lambda x.M/y][V_2/x] \Downarrow^n V}{M_1 M_2 \Downarrow^{\max(l+1,m+1,n)} V}
$$

Fig. 3. Operational semantics profiling stack space

We prove this theorem by the method of logical relations. Before defining the logical relations we define the auxiliary relation: $V_1 V_2 \downarrow^n V$ holds if one of the following conditions holds.

1. $V_1 \equiv \lambda x.M$ and $M[V_2/x] \Downarrow^n V$.
2. $V_1 \equiv \mathtt{fix}^{k+1} y.\lambda x.M$ and $M[V_2/x][\mathtt{fix}^k y.\lambda x.M/y] \Downarrow^n V$.

By using this relation we can combine the two rules for evaluation of application into the following rule.

$$
\frac{M_1 \Downarrow^l V_1 \quad M_2 \Downarrow^m V_2 \quad V_1 V_2 \downarrow^n V}{M_1 M_2 \Downarrow^{\max(l+1,m+1,n)} V}
$$

This reformulation simplifies the definition of the logical relations and our proof. We define logical relations $V \approx^C_\tau W$ indexed by a type τ and a positive integer C as follows.

$$
\begin{array}{rl}
\underline{i} \approx^C_{\mathsf{int}} \underline{i} & \\
V \approx^C_{\mathsf{bigint}} \bar{i} & V = \underline{i} \text{ or } V = \bar{i} \\
V \approx^C_{\tau_1 \to \tau_2} W & \\
\multicolumn{2}{l}{\quad \left\{ \begin{array}{l} \text{for all } V_1 \approx^C_{\tau_1} W_1,\ \text{if } V V_1 \downarrow^{n+1} V_2 \\ \text{then } W W_1 \downarrow^{\le Cn + \lfloor \tau_2 \rfloor + 3} W_2 \text{ and } V_2 \approx^C_{\tau_2} W_2 \end{array} \right.}
\end{array}
$$

The parameter C corresponds to the ratio of increase of stack space usage. It should be remarked that increase of stack space usage depends on only the range type τ_2 of a function type $\tau_1 \to \tau_2$. This is explained by checking the following translation of a function f of type $\tau_1 \to \tau_2$.

$$\mathsf{coerce}_{\tau_1 \to \tau_2 \le \sigma_1 \to \sigma_2}(\mathsf{f}) \equiv \lambda \mathsf{y}.\mathsf{coerce}_{\tau_2 \le \sigma_2}(\mathsf{f}\,\mathsf{coerce}_{\sigma_1 \le \tau_1}(\mathsf{y}))$$

In this translation, only the coercion $\mathsf{coerce}_{\tau_2 \le \sigma_2}$ causes increase of stack space usage.

We first show that a conversion from τ to σ behaves well with respect to the logical relations.

Lemma 1: If $\tau < \sigma$ and $V \approx^C_\tau W$ then $\mathsf{coerce}_{\tau \le \sigma}(\mathsf{W}) \Downarrow^2 \mathsf{W}'$ and $V \approx^C_\sigma W'$ for some W'.

The next lemma tells that we can choose a constant C such that the evaluation of a source program and its translation are related by C. For ρ and ρ' two environments with same domain, $\rho \approx^C_\Gamma \rho'$ means that they are pointwise related. The main theorem is obtained by restricting this lemma to $\Gamma = \emptyset$.

Lemma 2: Let C be a positive integer such that $C > \lfloor \sigma \rfloor$ for all σ appearing in the derivation of $\Gamma \vdash M : \tau$. Let $[|\Gamma \vdash M : \tau|] = N$ and $\rho \approx^C_\Gamma \rho'$. If $\rho(M) \Downarrow^{n+1} V$ then $\rho'(N) \Downarrow^{\le Cn + \lfloor \tau \rfloor + 2} W$ and $V \approx^C_\tau W$ for some W.

V. PRESERVATION OF EVALUATION TIME

We show that coercion interpretation of subtyping also preserves evaluation time within a factor determined by types occurring typing derivation. We outline the proof and show similarity of the proof to that for stack space in the previous section.

As the previous section, we first extend judgement of operational semantics into following form:

$$M \Downarrow^n V$$

where n represents execution time to evaluate M to V. The rules are given in Fig. 4. All the rules are straightforward extension of the standard rules.

$$V \Downarrow^{1} V \qquad \frac{M_1 \Downarrow^{m} V_1 \quad M_2[V_1/x] \Downarrow^{n} V}{\texttt{let } x = M_1 \texttt{ in } M_2 \Downarrow^{m+n+1} V}$$

$$\frac{M_1 \Downarrow^{l} \lambda x.M \quad M_2 \Downarrow^{m} V_2 \quad M[V_2/x] \Downarrow^{n} V}{M_1 M_2 \Downarrow^{l+m+n+1} V}$$

$$\frac{M_1 \Downarrow^{l} \texttt{fix}^{k+1}\, y.\lambda x.M \quad M_2 \Downarrow^{m} V_2 \quad M[\texttt{fix}^{k}\, y.\lambda x.M/y][V_2/x] \Downarrow^{n} V}{M_1 M_2 \Downarrow^{l+m+n+1} V}$$

Fig. 4. Operational semantics profiling execution time

We reformulate the operational semantics by a auxiliary relation: $V_1 V_2 \downarrow^{n} V$ holds if one of the following conditions holds.
1. $V_1 \equiv \lambda x.M$ and $M[V_2/x] \Downarrow^{n} V$.
2. $V_1 \equiv \texttt{fix}^{k+1} y.\lambda x.M$ and $M[V_2/x][\texttt{fix}^{k} y.\lambda x.M/y] \Downarrow^{n} V$.

By using this relation we can combine the two rules for evaluation of application into the following rule.

$$\frac{M_1 \Downarrow^{l} V_1 \quad M_2 \Downarrow^{m} V_2 \quad V_1 V_2 \downarrow^{n} V}{M_1 M_2 \Downarrow^{l+m+n+1} V}$$

The following is the main theorem of this section. The conversion interpretation preserves execution time within a factor determined by the sizes of types appearing in a program.

Theorem 2: Let $[|\emptyset \vdash M : \tau|] = N$ and C be a positive integer such that $C > 7\lfloor \sigma \rfloor$ for all σ appearing in the derivation of $\emptyset \vdash M : \tau$.

If $M \Downarrow^{n} V$, then $N \Downarrow^{\leq Cn} W$ for some W.

We define logical relations $V \approx^{C}_{\tau} W$ as follows.

$$\begin{array}{ll}
\underline{i} \approx^{C}_{\mathsf{int}} \underline{i} & \\
V \approx^{C}_{\mathsf{bigint}} \bar{i} & V = \underline{i} \text{ or } V = \bar{i} \\
V \approx^{C}_{\tau_1 \to \tau_2} W & \\
\multicolumn{2}{l}{\quad \left\{ \begin{array}{l} \text{for all } V_1 \approx^{C}_{\tau_1} W_1, \text{ if } V V_1 \downarrow^{n+1} V_2 \\ \text{then } W W_1 \downarrow^{Cn+7\lfloor \tau_1 \to \tau_2 \rfloor+1} W_2 \text{ and } V_2 \approx^{C}_{\tau_2} W_2 \end{array} \right.}
\end{array}$$

The important difference from the relations for stack space is that slowdown of applications depends on the domain type τ_1 as well as the range type τ_2 of a function type $\tau_1 \to \tau_2$.

Then it is shown that that a conversion from τ to σ behaves well with respect to the logical relations.

Lemma 3: If $\tau < \sigma$ and $V \approx^{C}_{\tau} W$ then $\mathsf{coerce}_{\tau \leq \sigma}(\mathsf{W}) \Downarrow^{\leq 3} \mathsf{W}'$ and $V \approx^{C}_{\sigma} W'$ for some W'.

With this lemma, the following lemma is proved in the almost same manner as the proof for stack space usage. The theorem is obtained by restricting this lemma.

Lemma 4: Let C be a positive integer such that $C > 7\lfloor \sigma \rfloor$ for all σ appearing in the derivation of $\Gamma \vdash M : \tau$.

Let $[|\Gamma \vdash M : \tau|] = N$ and $\rho \approx^{C}_{\Gamma} \rho'$. If $\rho(M) \Downarrow^{n+1} V$ then $\rho'(N) \Downarrow^{\leq Cn+7\lfloor \tau \rfloor+1} W$ and $V \approx^{C}_{\tau} W$ for some W.

VI. CONCLUSION

We have shown that conversion interpretation of subtyping preserves execution time and stack space within a factor determined by types in a program. Based on these results, it is possible to adopt conversion interpretation for optimising compilation of functional languages. However, as we discussed the subtyping relation must be partial order. Type-directed unboxing of Leroy is a translation similar to conversion interpretation of subtyping, but it does not preserve execution time and stack space. It is because conversions of equivalent types appear in type-directed unboxing.

We have considered only a very simple type system which does not include product types and recursive types. We believe that the results in this paper can be easily extended to product types. But recursive types must be treated carefully. If we consider subtyping on recursive types, the cost of one application of coercion cannot be bounded by any constant as Leroy discussed in his work of type-directed unboxing for polymorphic languages [5]. Thus the conversion interpretation does not preserve execution time nor stack space usage in the presence of subtyping on recursive types.

REFERENCES

[1] V. Breazu-Tannen, C. A. Gunter, and A. Scedrov, "Compuing with coercions," in *Proceedings of the 1990 ACM conference on LISP and functional programming*, 1990.

[2] John C. Mitchell, *Foundations for Programming Languages*, chapter 10, The MIT Press, 1996.

[3] Yasuhiko Minamide and Jacques Garrigue, "On the runtime complexity of type-directed unboxing," in *Proc. of ACM SIGPLAN International Conference on Functional Programming*, 1998, pp. 1–12.

[4] Nevin Heintze, "Control-flow analysis and type systems," in *Proc. the 1995 International Static Analysis Symposium*, 1995, pp. 189–206, LNCS 983.

[5] Xavier Leroy, "Unboxed objects and polymorphic typing," in *Proc. of ACM Symposium on Principles of Programming Languages*, 1992, pp. 177 – 188.

[6] Carl. A. Gunter, *Semantics of Programming Languages*, chapter 4, The MIT Press, 1992.

[7] Yasuhiko Minamide, "A new criterion for safe program transformations," in *Proceedings of the Forth International Workshop on Higher Order Operational Techniques in Semantics (HOOTS)*, Montreal, 1999, vol. 41 of *ENTCS*.

Measuring the Effectiveness of Web-based Learning System

Hiroyuki Murakoshi
Japan Advanced Institute
of Science and Technology
1-1 Asahidai Tatsunokuchi
Nomi-Gun 923-1292 Japan
murakosi@jaist.ac.jp

Koichiro Ochimizu
Japan Advanced Institute
of Science and Technology
1-1 Asahidai Tatsunokuchi
Nomi-Gun 923-1292 Japan
ochimizu@jaist.ac.jp

Abstract— **Web-based distance learning system supports distributed and asynchronous learning. Web-based learning approach became more popular, and many Web-based learning systems have been developed. It is necessary to give the above community a method of measuring the effectiveness of the Web-based learning systems to make it easier to improve them. In this paper, we propose two kinds of methods for measuring the effectiveness of Web-based learning system, the method using AHP and using a concept map. We also report the results of our methods applied to real classroom learning.**

Keywords— **Web-based Learning, Distance Education, Evaluation, AHP, Concept Map**

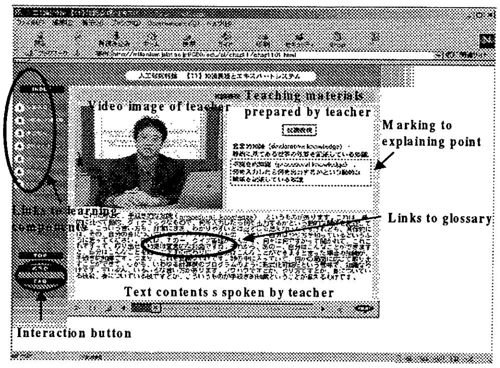


Fig. 1. Web-based Learning System

I. Introduction

WEB-BASED distance learning system supports distributed and asynchronous learning. Web-based learning approach became more popular, and many Web-based learning systems have been developed. It is necessary to give the above community a method of measuring the effectiveness of the Web-based learning systems to make it easier to improve them. An questionnaire survey or an usual achievement test is usually adopted as means of the evaluation. However, they will not make it easier to improve the Web-based learning systems because it is difficult to extract clues to points to be improved from a grate variety of qualitative opinions and scores of learners. We have already proposed a preliminary version of a method for measuring the effectiveness using analytic hierarchy process (AHP) [1]. This method was useful for measuring the effectiveness of our Web-based learning system by comparing Web-based learning with classroom learning. However, it was necessary to examine the criteria of the AHP diagram and evaluation process in the method. We also think that it will be necessary to develop new method for executing the evaluation based on the learning assessment to improve the Web-based learning system efficiently.

In this paper, we propose two kinds of methods for measuring the effectiveness of Web-based learning system, the method using AHP and using a concept map. Important parameters for improving the Web-based learning system are extracted in the evaluation using AHP. Improved points are extracted from the cause of parts which learners were not able to understood sufficiently in the evaluation using a concept map. We applied our methods to real classroom and succeeded in capturing several important parameters and in points to be improved.

II. Web/VOD-based Learning System

We developed two kinds of learning systems, Web-based and VOD-based, in JAIST Virtual University Project. These systems are designed for learning "Artificial Intelligence" and given by Professor Tojo in JAIST.

Fig. 1 shows the snapshot of Web-based learning system. This course is composed of 15 lectures. One lecture takes about 50 minutes. Note that one usual classroom lecture takes about 90 minutes. Our system is composed of the video image of teacher, presentation material viewer, text viewer, movie slider function tool boxes, knowledge units index, FAQ and bulletin board. In the video viewer of teacher, the video image of the RealVideo format taken a picture by focusing on the teacher is displayed. In sub materials viewer, the presentation materials are displayed synchronizing with the video image. They can be also displayed the marking to explaining part by using Macromedia Flash for intelligibility. In the text viewer, the content that the teacher spoke is displayed as it is. So learners can confirm the part they missed hearing, and take a note.

Fig. 2. VOD-based Learning System

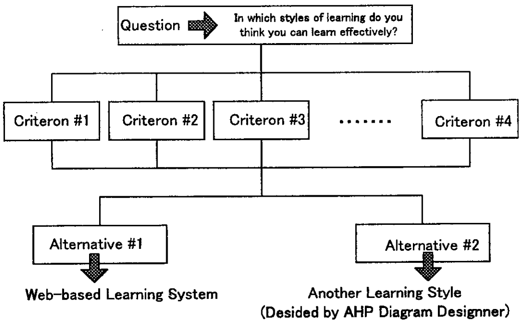


Fig. 3. AHP Hierarchical Diagram Template

They can also study the meaning of the term of texts because there is a link to the glossary. The left area is INDEX frame. Each knowledge unit is linked here. In addition, leaner can take communications as interactive function by using FAQ and the bulletin board that can be used by not only the teacher but also those who learn. This system requires a personal computer with Windows OS.

Fig. 2 shows the snapshot of VOD-based learning system. The video image of this system is identical with the one of the Web-based learning system. Note that the video image is encoded in the MPEG1 format.

III. METHOD FOR MEASURING

In the context of education, assessment means measurement of attributes of an individual. And evaluation means measurement of attributes of a program, course, or curriculum. The area of assessment and evaluation are very interesting and a teacher might want to explore them in the context of designing, implementing, and studying his/her course. An questionnaire survey or an achievement test is usually adopted as means of evaluation. However, they will not make it easier to improve the Web-based learning systems because it is difficult to extract clues to points to be improved from a grate variety of qualitative opinions and scores of learners. We focus on AHP [2] and a concept map [3] as methods for measuring the effectiveness of our Web-based learning system.

A. Extraction of Important Parameters using AHP

We have developed method for extracting important parameters as clues to improve the Web-based learning system using AHP [4].

1) What is AHP?

In decision-making, there is usually a "question" and some "alternatives" for the final decision. There are several "criteria" for choosing a final decision among some "alternatives". AHP [2] helps us choose one of several alternatives effectively based on some criteria. In the AHP hierarchical structure shown in Fig. 3 , we call it an AHP template, the question is "In which styles of learning do you think you can learn effectively?". Alternative#1 is Web-based learning system. Alternative#2 is another learning style decided by the AHP Diagram Designer. Decision-making process is executed by the following procedure.

1. The weight of each criterion is calculated by giving a weight to every pairs of criteria.
2. Total score (100 points) is divided among criteria according to the weight of each criterion (score of each criterion).
3. At each criterion, the weight of each alternative is calculated by giving a weight to every pairs of alternative. The score of each criterion is divided among alternatives according to the weight.
4. The score of each alternative is calculated by adding the scores of every criterion.

2) Method

We show our method in Fig. 4. The method is represented by a UML activity diagram. In Fig. 4, there are four swim lanes: Subject; Experimenter; AHP diagram Designer; and Analyst. A circled box shows an activity and a solid arrow shows a flow of control. A rectangular box shows an object passed through from some activity to the other represented by a dashed arrow. In Fig. 4,

1. "Subjects" take a lecture using Web-based learning system and answer a question given by a "Experimenter".
2. "Experimenter" performs an experiment to collect data.
3. "AHP Diagram Designer"

   (a) decides another learning style , e.g. VDO-based or classroom, compared to Web-based learning.

   (b) decides the criteria and completes the AHP di-

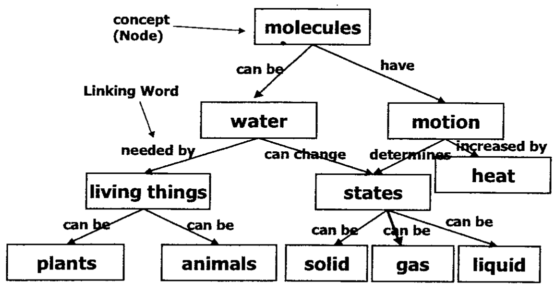


Fig. 5. Example of Concept Map

agram.

(c) calculates weight and score for each criterion.

(d) finally decides the score for each alternative.

4. " Analyst"

(a) calculates an average of scores for each criterion. This score implies that criterion is useful for evaluating the learning system.

(b) classifies criteria into several categories by an average of scores, e.g. high and low.

(c) compares the average of scores of Web-based learning system with the average of scores of another learning style for each criterion. Does F-test to know the difference is significant. If it is significant, the criterion is a useful parameters to characterize the difference between the Web-based learning and another learning style.

5. Makes the characteristic table shown in TABLE I.

6. Selects the important parameters as clues to improve the Web-based learning system shown in TABLE II.

7. Improves the teaching material based on the results.

B. Evaluation using Concept Map

1) What is a Concept Map?

A concept map is intended to represent meaningful relationships between concepts in the form of propositions [3]. Propositions are two or more concept labels linked by words in a semantic unit. In its simplest form, a concept map would be just two concepts connected by a linking word to form a proposition. We show an example of a concept map in Fig. 5. "Water can change state" would represent a simple concept map forming a valid proposition about the concepts "water" and "state" .

2) Method

The potential use of the concept map to assess students' knowledge structures has been recognized [5]. We have developed a method for evaluating learners' assessment and Web-based learning system [6]. Fig. 6 shows an outline of the method. The method is executed by the following procedure.

1. Parts which learners were not able to understood sufficiently are extracted by extracting differences between the concept map drawn by the learner (we call it "learner concept map") and the map drawn by a teacher (we call it "teacher concept map").

2. Nakakoji developed a system to recognize the breakdown and support mutual understanding, by showing workers the differences among them visually [7]. The learner is shown the above two concept maps. Experimenters interview the learner to promote self-reflection. They have the learner learn again using the Web-based learning system and examine the cause of a lack of understanding.

3. Based on the above findings, points to improve the Web-based learning system are given.

IV. EXPERIMENT ENVIRONMENT

We set up several experiments where students study themselves using our Web-based learning system or VOD-based learning system instead of teacher's talk [4][8]. The experiments were performed as a part of real classroom lecture of "Artificial Intelligence" in JAIST. At the experiments, we performed the following lecture (five times).

1. 4th Logical Inference #1 (Web-based)
2. 5th Logical Inference #2 (Web-based)
3. 9th Prolog Programming #1 (Web-based)
4. 10th Prolog Programming #2 (VOD-based)
5. 11th Knowledge Representation and Expert System (Web-based)
6. 12th Planning (Web-based)

Students who attended the lectures were able to use our Web-based learning system and VOD-based learning system by personal computers connected to local area network.

V. EVALUATION USING AHP

A. Web vs. Classroom

1) Design of AHP Diagram

The AHP hierarchical diagram designed is shown in Fig. 7. Dotted circles show the selected alternative and enumerated criteria. The criteria are selected to compare the effectiveness of Web-based learning with that of classroom learning, from viewpoint of which factors are needed to learn effectively.

A. Learning Environment: Learners have different styles of learning. They might learn in a manner that

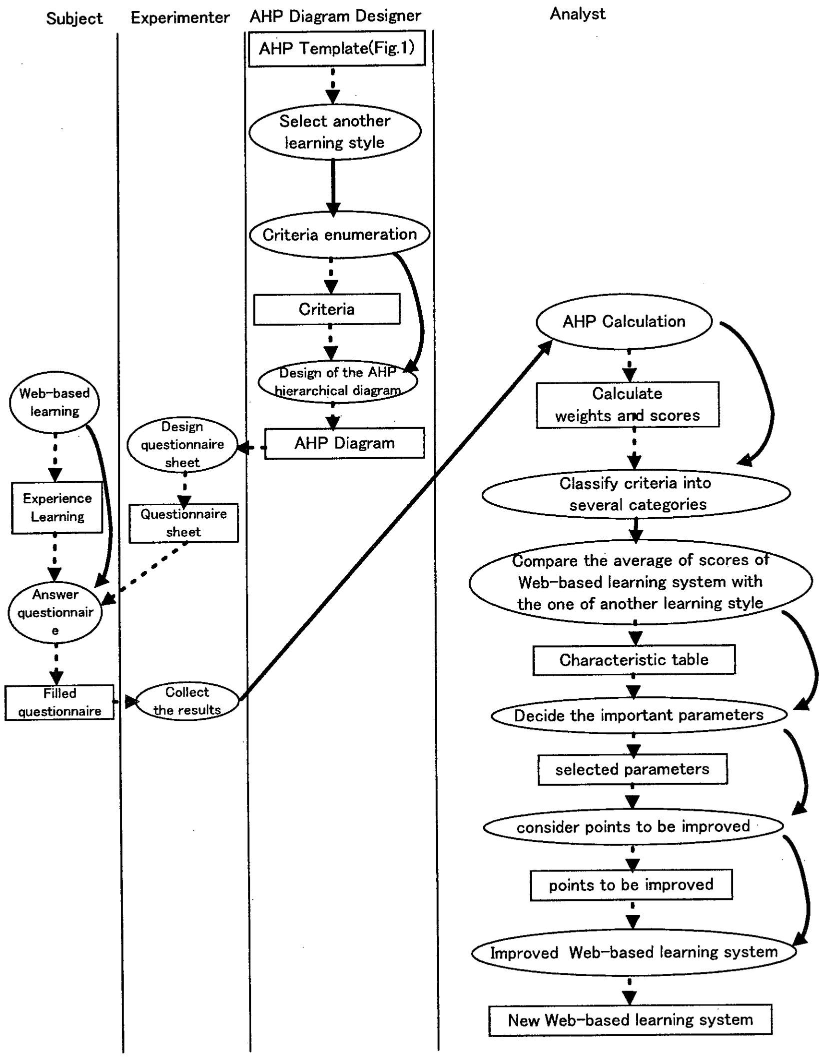


Fig. 4. Method using AHP

suits their styles. This criterion is divided into the following six criteria.

1. Ease of Interaction between Learners: Ease and smoothness of communication among learners.

2. Ease of Interaction with Teacher: Ease and smoothness of communication between learners and a teacher.

3. Space Convenience: Learners may want to take lecture at the place where they are.

4.Time Convenience: Learners may want to take lecture whenever they like.

5. Presence and Atmosphere: A student may feel the presence of the teacher in a classroom. When students learn the most important things, there may be in a tense atmosphere in the classroom.

6. Richness of Teaching Material: Usefulness of teaching materials.

B. Teacher's Presentation: Presentation method with which the teacher provides knowledge and concepts. We divide it into the following three criteria.

7. Talk and Tone: Comfortableness of the teacher's talk and tone.

TABLE I
FOUR CATEGORIES BY SCORE LEVEL AND EXISTENCE OF DIFFERENCE (CHARACTERISTIC TABLE)

	Statistical Significant Difference	No Statistical Significant Difference
Score is High	Criteria in this area (Area 1) will be regarded as important parameters that influence the effectiveness of learning and characterize difference between another learning and Web-based learning.	Criteria in this area (Area 3) will be regarded as important parameters that influence the effectiveness of learning.
Score is Low	Criteria in this area (Area 2) will be regarded as important parameters that characterize difference between another learning and Web-based learning.	Criteria in this area (Area 4) will be regarded as unimportant parameters.

TABLE II
SELECTED PARAMETERS AS CLUES TO IMPROVE WEB-BASED SYSTEM

Stage Level	Selected Parameters
The First Stage	Parameters whose the score of Web-based is lower than the one of another learning in Area 1 shown in TABLE I. Parameters in Area 3.
The Last Stage	Parameters whose the score of Web-based is lower than the one of another learning in Area 2 shown in TABLE I. Parameters in Area 4.

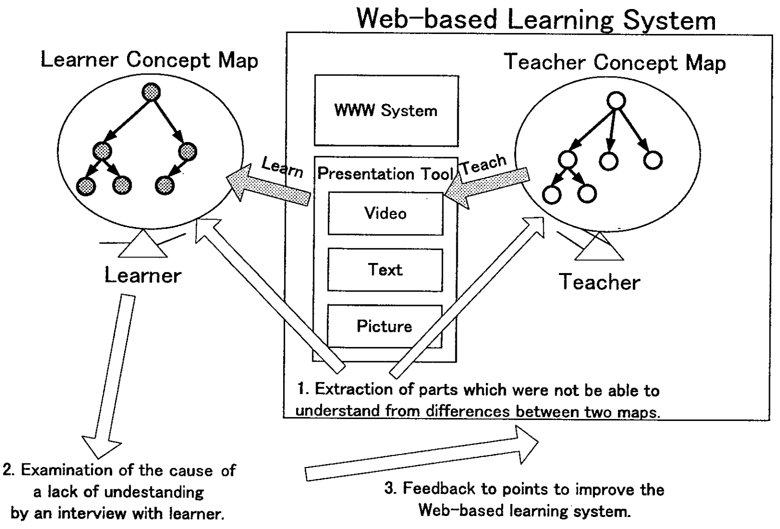


Fig. 6. Method using Concept Map

8. The way to Explain Teaching Materials: Ease of understanding the way to explain teaching materials.

9. The way to Summarize Teaching Materials: Ease of understanding summarization of teaching materials.

C. Motivation for Learning: This criterion is a measure for evaluating learner's motivation and conti-

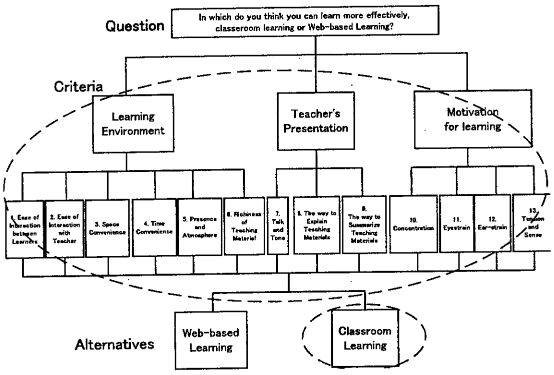


Fig. 7. AHP Hierarchical Diagram (Web vs. Classroom)

nuity of learning. We divide it into the following four criteria.

10. Concentration: Concentration without considering surroundings.

11. Eyestrain: Degree of eye-strain.

12. Ear-strain: Degree of ear-strain.

13. Tension and Sense: Commitment to learning.

2) Results

We made two kinds of questionnaire sheets for the calculation by the AHP diagram shown in Fig. 7. One is a sheet to give a weight to every pairs of criteria. Another is a sheet to give a weight to Web-based learning and classroom learning to each criterion. After the lecture of "5th Logical Inference #2", we collected sheets answered by 68 students who attended empirical lectures. We understood from the answer of the questionnaire of the sheets that the most of learners used Web-based learning system first time. We excluded the four students because the value of his/her AHP adjustment (C.I.) exceeded the permitted value. The C.I. is a criterion for measuring consistency when subjects evaluate a weight.

TABLE III shows 13 criteria classified into four categories (characteristic table). We regarded the average score of criteria as the threshold of the high and low. In TABLE III, the threshold value is 7.69 and a statistical significant level is 5% ($F(1,127,0.05)$). The italic font numbers mean that the score of Web-based is lower than the one of classroom. The results of TABLE III will mean the following findings.

- Studying themselves in Web-based learning may cause the significant difference. Motivation for learning will be an important parameter that influences the effectiveness of learning and characterizes classroom learning.
- The convenience will show the characteristic of Web-based learning. The presence and atmosphere will show the characteristics of classroom learning.
- The teacher's presentation would be an important parameter the effectiveness of learning.

TABLE III

RESULT OF THE FIRST QUESTIONNAIRE (VS.CLASSROOM)

	Statistical Significant Difference	No Statistical Significant Difference
Score is High ($\geq$ 7.69)	*10,11,12*	8,9
Score is Low (< 7.69)	3,4,*5*,*13*	*1*,*2*,6,*7*

The weight of the "Learner's Motivation" was much higher than the other criteria from the result of first questionnaire. To examine influence caused by the other criteria in detail, we removed "Learner's Motivation" of the AHP hierarchical diagram shown in Fig. 7. After the lecture of "12th Planning", we collected sheets answered by 64 students who attended empirical lectures. We excluded the five students because the value of his/her AHP adjustment (C.I.) exceeded the permitted value. TABLE IV shows 9 criteria classified into four categories. We regarded the average score of criteria as the threshold of the high and low. In TABLE IV, the threshold value is 11.1 and a statistical significant level is 5% ($F(1,119,0.05)$). The italic font numbers mean that the score of Web-based is lower than the one of classroom. We compared the results of TABLE III with ones of TABLE IV. We think that talk and tone of a teacher will be an important parameter the effectiveness of learning. We also think that the ease of interaction with a teacher will show the characteristics of classroom learning.

TABLE IV

RESULT OF THE SECOND QUESTIONNAIRE (VS. CLASSROOM)

	Statistical Significant Difference	No Statistical Significant Difference
Score is High ($\geq$ 11.1)		*7*,8,9
Score is Low (< 11.1)	*2*,3,4,*5*	1,6

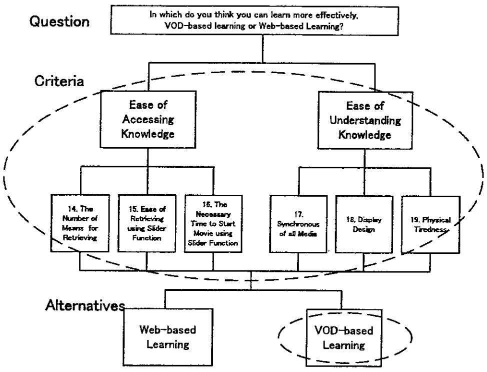


Fig. 8. AHP Hierarchical Diagram (Web vs. VOD)

B. Web vs. VOD

1) Design of AHP Diagram

The AHP hierarchical diagram designed is shown in Fig. 8. Dotted circle shows the selected alternative and enumerated criteria. The criteria are selected to compare the effectiveness of Web-based learning with that of VOD-based learning, from viewpoint of which factors are needed to learn effectively.

D. Ease of Accessing Knowledge: Ease of accessing knowledge in the learning system.

14. The number of Means for Retrieving: The number of means that learners can use retrieving according to their demand.

15. Ease of Retrieving using Slider Function: Ease of slider function when learners retrieve.

16. The necessary Time to start Movie: The necessary time to start movie when learners use slider function.

E. Ease of Understanding Knowledge: Ease of understanding knowledge using the learning system.

17. Synchronous of all Media: Synchronous of the presented movie, voice and material.

18. Display Design: Diplay design of the learning system.

19. Physical Tiredness: Degree of physical fatigue.

2) Results

We made two kinds of questionnaire sheets for the calculation by the AHP diagram shown in Fig. 8. One is a sheet to give a weight to every pairs of criteria. Another is a sheet to give a weight to Web-based learning and VOD-based learning to each criterion. After the lecture of “10th Prolog Programming #2 ”, we collected sheets answered by 65 students who attended empirical lectures. We excluded the eight students because the value of his/her AHP adjustment (C.I.) exceeded the permitted value. Note that we instructed the students to compare learning of “9th Prolog Programming #1 ” with “10th Prolog Programming #2 ”.

TABLE V shows 6 criteria classified into four categories. We regarded the average score of criteria as the threshold of the high and low. In TABLE V, the threshold value is 16.7 and a statistical significant level is 5% ($F(1, 113, 0.05)$). The italic font numbers mean that the score of Web-based is lower than the one of VOD-based. The results of TABLE V will mean the following findings.

- Physical tiredness for learning will be an important parameter that influences the effectiveness of learning and characterizes learning. Quality of video which our Web-based learning system provides may cause the significant difference.
- In our Web-based learning system, the necessary time to start movie may cause the wrong effect of learning when learners use slider function.

TABLE V
RESULT OF THE THIRD QUESTIONNAIRE (VS. VOD)

	Statistical Significant Difference	No Statistical Significant Difference
Score is High (≥ 16.7)	*19*	15,*17*
Score is Low (< 16.7)	14,*16*,18	

C. Selected Parameters

TABLE VI shows selected parameters as clues to improve the Web-based learning system. At the experiment using the improved Web-based leaning system, if parameters move from Area 1 to Area 3, from Area 3 to Area 1, from Area 2 to Area 4, and from Area 4 to Area 2, The method will be useful and the selected parameters will be appropriate for clues to improve Web-based learning system.

VI. EVALUATION USING CONCEPT MAP

A. Environment

A method using a concept map which we have developed was executed by 13 students who attended empirical lectures. 7 students executed at learning of “4th Logical Inference #1 ” and “5th Logical Inference #2 ” (The first evaluation). 6 students executed “11th Knowledge Representation Expert System ” and “12th Planning ” (The second evaluation).

Ruiz-Primo and Shavelson explored the problems and issues in the use of concept mapping and proposed a framework for conceptualizing concept maps

TABLE VI
SELECTED PARAMETERS

Stage Level	Selected Parameters(vs.Classroom)	Selected Parameters(vs.VOD)
The First Stage	10,11,12 in Area 1 7,8.9 in Area 3	19 in Area 1 15,17 in Area 3
The Last Stage	16 in Area 2	Nothing in Area 4

as a potential assessment tool in science assessment [5]. They conceived of an assessment as a combination of *a task*, *a response format*, and *a scoring system*. Based on this framework, a concept map used as an assessment tool can be characterized as: (a) a task that invites students to provide evidence bearing on their knowledge structure in a domain, (b) a format for the students response, and (c) a scoring system by which students' concept maps can be evaluated accurately and consistently. Without these three components, a concept map cannot be considered in assessment. At this experiment, we set up the following component in learning assessment of the method shown in Fig. 6.

Assessment Task: The assessor provides the concepts and/or linking words or or asks students to construct a hierarchical map. This technique is called "construct-a-map". A teacher provided the concepts at the 1st evaluation. Students provided the concepts at the 2st evaluation. We provided two kinds of linking words, "is-a relation" and "has-a relation". "Is-a relation" has four kinds of relations, "inclusiveness", "exclusiveness" , "both inclusiveness and exclusiveness" and "them of neither". "Has-a relation" has two kinds of relations. "essential part" and " optional part".

Response Format: Paper and pencil response. Students draw the concept map on a piece of paper.

Scoring System: The assessor compares the learner concept map drawn by the learner with the teacher concept map (criterion map) drawn by the teacher at the 1st evaluation. The teacher checks the learner concept map and examines serious mistakes at the both evaluations.

Experimenters extracted parts which were not able to understood sufficiently from the results of the above assessment. They showed the students the teacher concept map and the revised learner concept map, and interviewed the learner to promote self-reflection. The students learned again using the Web-based learning system and answered the cause of a lack of understanding.

B. Results

We recognized the following findings from the results of the interviews with the students and the answers by them, When there is the relationship between two concepts explained at the different lectures, learners tended to be difficult to understand the relationship. Teacher tends to explain the meaning of the technical words only one time because learners using Web-based leaning system can study the same part repeatedly. We think that this will cause a lack of understanding.

VII. CONCLUSIONS

We developed two kinds of methods using AHP and a concept map to measure the effectiveness of Web-based learning system. Important parameters are selected and improved points are extracted from results of the experiments performed as a part of real classroom lecture of "Artificial Intelligence" in JAIST.

We plan to examine clues to points to be improved from the above important parameters and improve our Web-based learning system. We also plan to measure the effectiveness of the improved system and confirm usefulness of our method and points to be improved.

REFERENCES

[1] Hiroyuki Murakoshi, Tsuyoshi Kawarasaki, and Koichiro Ochimizu, "Comparison using AHP of Web-based Learning with Classroom Learning," in *First International Workshop on Internet-Supported Education(WISE2001)*. Jan. 2001, pp. 67–73, IEEE Computer Society Press.

[2] Kaoru Tone and Ryutaro Manabe, *Exapmples of AHP*, JUSE Press, Jul. 1990(in Japanese).

[3] Joseph D. Novak and D. Bob Gowin, *LEARNING HOW TO LEARN*, Cambridge University Press, 1984.

[4] Atsuo Inomata, Mikio Kishi, Hiroyuki Murakoshi, Satoshi Tojo, and Koichiro Ochimizu, "Evaluation Method of Web-based Learning System," in *International Conference on Information Technology Based Higher Education and Training(ITHET01)*, July 2001.

[5] Maria A. Ruiz-Primo and Richard J. Shavelson, "Problems and issues in the use of concept maps in science asssessment," *Journal of Research in Science Teaching*, vol. 33, no. 6, pp. 569–600, 1996.

[6] Mitsuru Suzuki, Atsuo Inomata, Hiroyuki Murakoshi, Satoshi Tojo, and Koichiro Ochimizu, "A Method of Evaluating for Electronic Learning Material using Concept Map," *TECHNICAL REPORT OF IEICE ET2001-7*, vol. 101, no. 41, pp. 41–48, May. 2001(in Japanese).

[7] Kumiyo Nakakoji, Yasuhiro Yamamoto, Shingo Takada, and Breat N. Reeves., "Two-Dimensional Spatial Positioning as a Means for Reflection in Design," in *Design of Interactive Systems(DIS'2000)*, 2000, pp. 145–154.

[8] Atsuo Inomata, Hiroyuki Murakoshi, and Koichiro Ochimizu, "Evaluation Method of Distance Learning Teaching Materials by AHP," in *18th Conference Proceedings of Japan Society for Software Science and Technology*, Sep. 2001(in Japanese).

Towards A Retro Conversion of Mathematical Documents

A. Kacem
RIADI-ENSI (Tunisia)
Ecole Nationale des Sciences
de l'Informatique
Afef_kacem@yahoo.fr

A. Belaid
LORIA-CNRS(France)
Abdel.belaid@loria.fr

M. Benahmed
RIADI-ENSI (Tunisia)
Ecole Nationale des Sciences de
l'Informatique
Mohamed.benahmed@riadi.rnu.tn

Abstract– **Automatic extraction and recognition of mathematical formulas is one of the keys in the drive towards transcribing mathematical documents into electronic form. The ultimate objective is to enable mathematics to be served, received, and processed on the Web. This problem typically consists of three major stages, namely, formula extraction, formula recognition and formula encoding. The former stage consists of locating mathematical formulas in the document for both cases : as isolated formulas or as embedded in text lines. Experiments done on some commonly seen mathematical documents, show that our proposed method to extract mathematical formulas can achieve quite satisfactory rate. The extraction rate is close to 93%. In the second stage, we discuss the use of coordinate grammars for the recognition of mathematical formulas extracted from the document. A top-down parsing scheme is used to partition a formula into sub problems. Syntax rules contain all necessary partitioning instructions. The syntax-directed recognition system described here has been successfully demonstrated, and could be used to provide interpretation of mathematical documents. In the latter stage, we propose to convert the mathematical document using XML where formula will be encoded in MathML.**

I. INTRODUCTION

THE recent development of the OCR technology made possible its practical use in various applications. However, there is no commercial OCR software which an recognize the content of scientific documents including mathematical formulas [14]. It is obvious that separating mixed materials in mathematical documents should help the accuracy of commercial OCR programs. We propose to improve the OCR success rate on mixed material by separating mathematics from the usual text while on the other hand being able to analyze formulas. This paper addresses the issue of locating and analyzing formulas in mathematical documents. Such a tool could be really useful to be able to recognize mathematical documents and re-use them in other applications.

The mathematical document recognition process can be divided into the following steps : (1) Segmentation of the text area and the mathematical formula area, (2) Analyzing the content of the extracted mathematical formulas and (3) Recognition of the whole mathematical document. In our system, step (1) is performed using fuzzy logic. This step is described in section 2. In step (2), mathematical formulas are recognized by the top-down approach with a recurrent algorithm for sub formula areas included in formulas such as fractions, integrals, summations, etc. The outline of this step is described in section 3. In step (3), the whole mathematical document is recognized then converted in XML using the results of formula recognition and encoding

II. MATHEMATICAL FORMULA EXTRACTION

In a mathematical formula, characters and symbols can be spatially arranged as a complex two-dimensional structure, possibly of different character and symbol sizes. This makes the extraction and the recognition process more complicated even when all the individual characters and symbols can be recognized correctly. In this step, we separate formulas from ordinary text on a scanned mathematical document image. Formulas are involved in mathematical documents, either as isolated formulas, or embedded directly into a text line. We will provide in section 2.1, a survey of existing work in mathematical formula extraction. Besides, we will describe in section 2.2, our proposal approach which is then detailed in sections 2.3 and 2.4. Afterwards, we will discuss some experimental results.

A. *Previous Works*

Papers that provide literature survey of the area of mathematical formula extraction research are very rare [12-15]. Most papers that delve into mathematical document segmentation, without being specific, make use of OCR systems. They did not put much emphasis on explaining how the mathematical symbols are recognized and how the embedded formula are separated from the plain text or the explanations are too tedious and sometimes too *ad hoc*. In addition, the majority of works are done on some types of documents written in a known language and with a specific style and typography.

B. *Overview of the Extraction Process*

Mathematical formulas are represented with various kind of objects. Such objects include all possible alphabetic characters (English, Greek, Hebrew, etc.), numerals (1,2,3, etc.), math operators (+, *, -, Σ, Π, ∫, (, [, etc.) and so on. Though extraction of such objects is the first step to locate mathematical formulas. To proceed with our system, we must tentatively identify many of the connected components as particular characters. Characters that are known to be mathematical (such as Σ, Π, $\int$, etc.) are used as tokens for formula extraction.

To be able to identify mathematical symbols, our system begins with learning them. We have studied 1182 instances of mathematical symbols: 263 summation and product symbols (*SP*), 83 integrals (*IS*), 101 roots (*RS*), 109 horizontal fraction bars (*HFB*), 177 great delimiters (*GD* great brackets and parenthesis), 205 small delimiters (*SD* usual brackets and parenthesis) and 244 operators composed of small horizontal lines such as equal and subtraction signs (*OP*). For each instance of symbol, values of the aspect ratio (*R*), area (*A*) and density (*D*) of their connected components are computed. Let $P=\{R, A, D\}$ be the set of parameters used for mathematical symbol classification and $MS=\{SP, IS, RS, HFB, GD, SD, OP\}$ the set of mathematical symbol classes. We have used a classification method based on the fuzzy logic. This method consists of constructing histograms from the whole obtained measures. The histogram abscissa refers to the different value classes that is the set of measures shared on regular intervals. The ordinate is the relative frequency that is the number of measures belonging to a value class divided by the total number of measures. The ordinate could be considered as the membership degrees to the different classes of mathematical symbols [16-18].

To identify a mathematical symbol given its connected component (*C*), values of each parameters are computed. By referring to the histograms of each class of symbols, we each time keep the membership degree of *C* to a class of symbols according to one parameter noted $\mu_{MS,P}(\chi)$. We then keep, for each class of symbols, the minimal membership degree of *C* according to the three parameters. We finally take the maximal value. The membership degree of *C* to a class of a mathematical symbol is defined using the following formula:

$$\mu_{MS}(C)=Max(Min(\mu_{MS,R}(C),\mu_{MS,A}(C),\mu_{MS,D}(C))=Max(\mu_{SP}(C),\mu_{IS}(C),\mu_{RS}(C),\mu_{HFB}(C),\mu_{GD}(C),\mu_{SD}(C),\mu_{OP}(C)).$$

C. *Isolated Formula Extraction*

Isolated formulas are big formulas, which constitute a line by them selves, without or with very little text, and they are often centered in the page. Locating them is an easy and quick task. We begin by extracting the image lines following the next steps. First, the connected components are sorted by their ascending Y_{min}. Initially, the Y_{min} and the Y_{max} of the line correspond to those the first meted component. Then the line ordinates are updated by checking vertically overlapped components. Those belonging to the same line are then sorted by their ascending X_{min} to determine the X_{min} and the X_{max} of the line. Sometimes, a line fusion step seems to be necessary especially for formula spread over than one line such as fractional, summation, product or integral expressions. Once lines are extracted, isolated formulas could be located using two assumptions based on their morphology (a quite high and long line) and position on the image (centered line).

D. *Embedded Formula Extraction*

Embedded formulas are extracted, by identifying their most significant mathematical operators, then extension to adjoining operands and operators using contextual rules until delimitation of the whole formula spaces. A secondary labeling is applied. It is a finer labeling of the connected components where their position according to the central band of their line is considered to solve some ambiguities observed at their primary labeling. Six categories of components are proposed based on their topographies : overflowing, ascending, descending, centered, high and deepen. With this consideration, summation, product and integral symbols could be distinguished from some alphanumeric characters and oblique fraction bars, since integral, summation and product symbols are overflowing while alphanumeric characters and oblique fractions bars are not. Additionally, subscripts and superscripts could now be detected since they are generally deepen or high. As subscripts could be descending (not too deepen) and the superscripts could be ascending (not too high) and since both of them are implicit operators, Two other features : the relative size (*X*) and position (*Y*) are considered for their detection.

Before we can isolate embedded formula from text, we must first group the identified operators properly into units. Proper combination of operators must be syntactically correct in a mathematical sense. This can be done by some conventions in writing mathematical formulas as heuristics. For summation, product and integral symbols and operators, the propagation of context is done around them. For parenthesis, brackets and roots, it is done between them. For fraction bars, it is done above and under them.

The current developed system to extract mathematical formulas runs under PC Pentium II. The documents are scanned and saved as a binary image files at a resolution of 300dpi. In the experiments, the system is trained using 1182 mathematical symbols, 200 implicit operators and tested by 460 symbols, 100 implicit operators, 300 formulas and a variety of mathematical documents. The obtained results indicate that approximately 93% of formulas could be extracted from images of the mathematical documents.

III. MATHEMATICAL FORMULA RECOGNITION

In the past few decades, many researchers have developed a promising number of approaches for mathematical formula recognition [1-11]. Quite a number of mathematical expression recognition systems obtained the structure without parsing. Instead, some procedurally-coded rules were used while others applied parsing techniques with a range of variations[4]. However, it should be noted that most methods for the structural analysis of mathematical expressions are actually based on some kinds of syntax[3]. Whether to define the syntax implicitly or explicitly is just a matter of choice. For simple expressions, both ways should do the job well. The situation changes if we try to recognize more complex expressions. Rather than adding *ad hoc* procedurally-coded rules to the system and yet still being uncertain of the correctness of the structural analysis module, explicit rules in a parser may provide a clearer and more concise form for formal verification[3].

ANDERSON used a top-down parsing scheme which starts with the ultimate syntactic goal and the entire set of input characters and attempts to partition the problem into sub goals until either every sub goal is reached, or else all possibilities have failed. A syntax rule provides instructions for the partitioning of a character set into subsets, and assigns a syntactic goal to each of these subsets. Experimentation has shown that "bare" top-down syntactic analysis is quite inefficient. A major source of this inefficiency is the partitioning strategy used for rules with two non terminal syntactic units on their right-hand side. It should be noted that the recognition algorithm stops after the first successful parse. The syntax rules should be ordered so that the first interpretation encountered is the desired one. However, by taking advantage of certain features of mathematical notation, many "dead-end" analyses may be avoided.

In this section, we indicate a method for considerably improving the efficiency of the parsing algorithm and reducing the restrictions on symbol placement. We propose a mathematical formulas interpreter based on a syntactic parser. It starts the parsing by locating of the most important operator in the formula and attempts to partition it into sub formulas which are similarly analyzed by looking for a starting point. A top-down parsing algorithm was chosen because it provides a natural way of hypothesizing the global properties of the formulas at an early stage of the recognition procedure. The formula interpreter works here in conjunction with a character recognition system.The choice of the starting point is function of its precedence and dominance in the formula. When consecutive operators exist in an expression, we apply operator precedence rules. However, when those operator are not lined up, we have to use the concept of operator dominance. For example, in « $a+\frac{b}{c}$ », the meaning is « a+(b/c) » due to the fact that the operator '+' dominates '/' (where '/' lies in the range of '+'). However, in « $\frac{b+c}{d}$ », the meaning becomes « (b+c)/d » since '/' dominates '+' (where '+' lies in the range of '/') in this case.

TABLE I

INPUT OBJECTS, THEIR SYNTACTIC CATEGORIES AND SEMANTIC MEANING

Input object	Syntactic category	Semantic meaning
Σ Π	SP	\sum / \prod
√	RS	\sqrt
∫	IS	\int
———	HFB	\frac
(){}⟨⟩[]⌈⌊⌉⌋ |	GD	\lbrack / \rbrack
( ) { } ⟨ ⟩ [ ] ⌈ ⌊ ⌉ ⌋ |	SD	(/ [/ { / | / } /] /)
- = ≤ ≥ ± ⊆ ⊇	OP	- / = / ≡ / <= />=
	SUB	
	SUP	
a b…z A B…Z 0 1…9 α β…ώ	CAR	a/b/…/z/A/B/…/Z/0/1/…/ 9/α/β/…ώ/

A. Formula Description

We consider a mathematical formula as a coherent combination of objects (operators and operands) connected together by their positions and by relationships. The analysis of a formula consists of locating the sub patterns which compose it and finding relations which bind them together [2].

The objects are presented to the interpreter with their name and extend, indicated by the four value : X_{min}, Y_{min}, X_{min} and Y_{max}. The objects are sorted by ascending order of their X_{min}. All syntactic relationships are based on the positions of objects relative to each other (see Figure 1). For efficiency in the syntactic analysis, each object is given a syntactic category. We also use an *m* coordinate to store the semantic meaning (in Latex) attached to a syntactic unit as a character string (see Table 1).

- ***U** (Up)* : $U(Z_i)=[\forall Z_j \;/\; (Y_{max}(Z_j)\le Y_{max}(Z_i) \text{ and } Y_{min}(Z_j)\le Y_{min}(Z_i)) \text{ and } ((X_{min}(Z_j)\ge X_{min}(Z_i) \text{ and } X_{max}(Z_j) \le X_{max}(Z_i)) \text{ or } (X_{min}(Z_j)\le X_{min}(Z_i) \text{ and } X_{max}(Z_j) \ge X_{max}(Z_i)))]$
- ***D** (Down)* : $D(Z_i)=[\forall Z_j \;/\; (Y_{max}(Z_j)\ge Y_{max}(Z_i) \text{ and } Y_{min}(Z_j)\ge Y_{min}(Z_i)) \text{ and } ((X_{min}(Z_j)\ge X_{min}(Z_i) \text{ and } X_{max}(Z_j) \le X_{max}(Z_i)) \text{ or } (X_{min}(Z_j)\le X_{min}(Z_i) \text{ et } X_{max}(Z_j)\ge X_{max}(Z_i)))]$
- ***R** (Right)* : $R(Z_i)=[\forall Z_j \;/\; (X_{max}(Z_j)\ge X_{max}(Z_i) \text{ and } X_{min}(Z_j)\ge X_{min}(Z_i)) \text{ and } ((Y_{min}(Z_j)\ge Y_{min}(Z_i) \text{ and } Y_{max}(Z_j) \le Y_{max}(Z_i))) \text{ or } (Y_{min}(Z_j)\le Y_{min}(Z_i) \text{ and } Y_{max}(Z_j) \ge Y_{max}(Z_i))]$
- ***L** (Left)* : $L(Z_i)=[\forall Z_j \;/\; (X_{min}(Z_j)\le X_{min}(Z_i) \text{ and } X_{max}(Z_j)\le X_{max}(Z_i)) \text{ and } ((Y_{min}(Z_j)\ge Y_{min}(Z_i) \text{ and } Y_{max}(Z_j) \le Y_{max}(Z_i))) \text{ or } (Y_{min}(Z_j)\le Y_{min}(Z_i) \text{ et } Y_{max}(Z_j) \ge Y_{max}(Z_i))]$
- ***S** (Superscript)* : $S(Z_i)=[\forall Z_j \;/\; (X_{min}(Z_j)\le X_{min}(Z_i) \text{ and } Y_{max}(Z_j)\le Y_{max}(Z_i) \text{ and } Y_{min}(Z_j)\le Y_{min}(Z_i)) \text{ or } (X_{max}(Z_j)\ge X_{max}(Z_i) \text{ and } Y_{max}(Z_j)\le Y_{max}(Z_i) \text{ and } Y_{min}(Z_j)\le Y_{min}(Z_i))]$
- ***s** (Subscript)* : $s(Z_i)=[\forall Z_j \;/\; (X_{min}(Z_j)\le X_{min}(Z_i) \text{ and } Y_{min}(Z_j)\ge Y_{min}(Z_i) \text{ and } Y_{max}(Z_j)\ge Y_{max}(Z_i)) \text{ or } (X_{max}(Z_j)\ge X_{max}(Z_i) \text{ and } Y_{min}(Z_j)\ge Y_{min}(Z_i) \text{ and } Y_{max}(Z_j)\ge Y_{max}(Z_i))]$
- ***I** (Inclusion)* : $I(Z_i)=[\forall Z_j \;/\; (X_{min}(Z_j)\le X_{max}(Z_i) \text{ and } Y_{min}(Z_j) \ge Y_{min}(Z_i) \text{ and } Y_{max}(Z_j) \le Y_{max}(Z_i))]$

B. Grammar Description

The formula grammar we use can be formally defined as a 4-tuple : G=(T,N,P,S) where

- T is a set of terminal symbols which correspond to the syntactic units : SP, RS, IS, HFB, GD, SD, OP, SUB, SUP and CAR,
- N is a set of non terminal symbols ($T \cap N = \emptyset$),
- P is a finite set of productions,
- S is the grammar axiom.

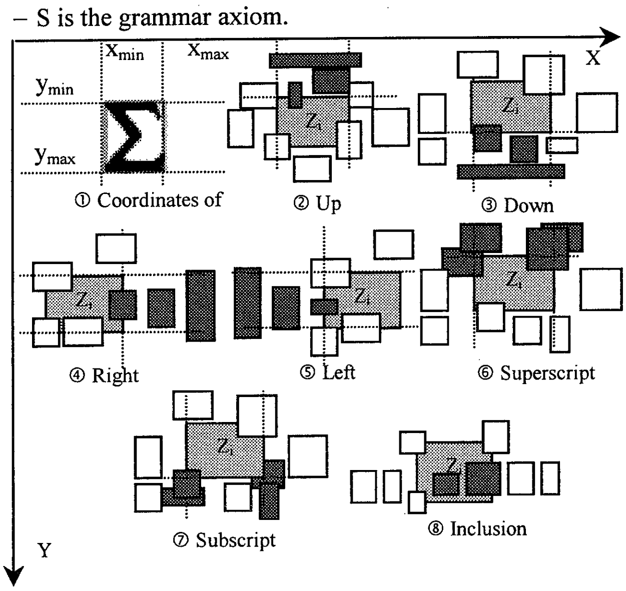
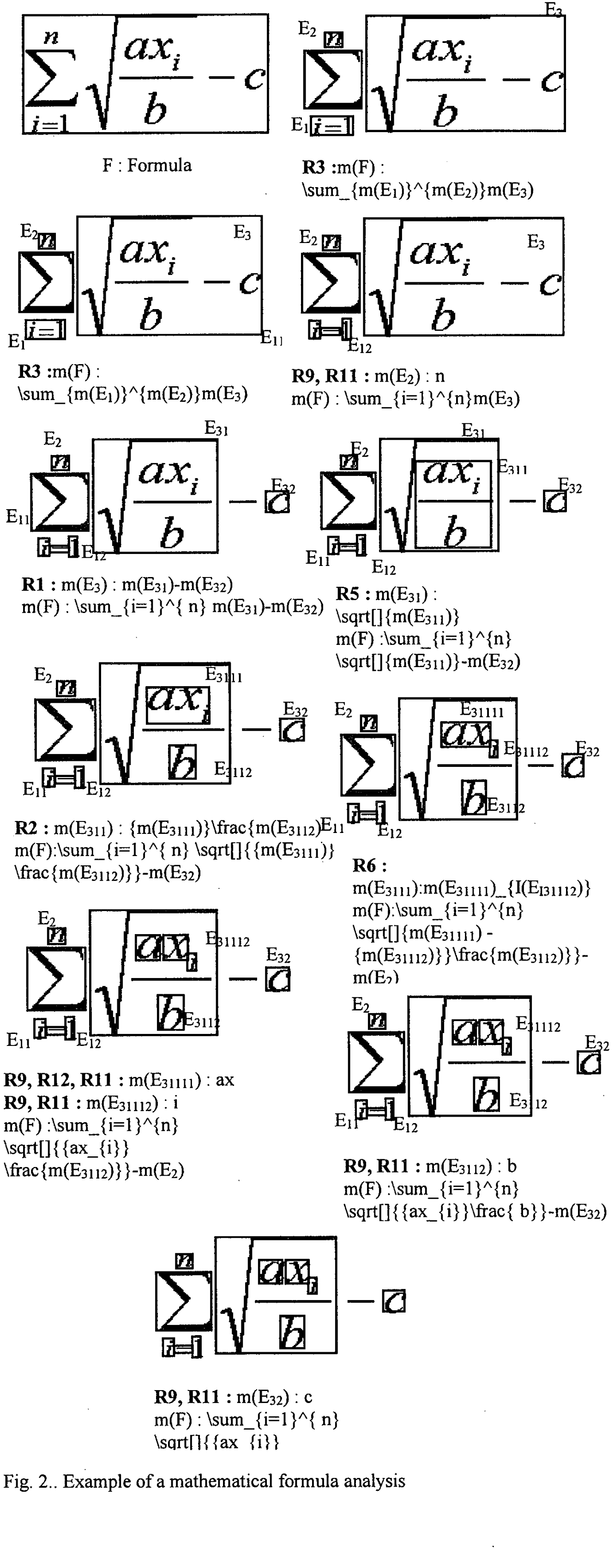


Fig. 1. Coordinates of the objects and their syntactic relationships.

This grammar is composed of context-free rewriting rules. Here are some examples of rules. We developed a notation in which all the above information may be encoded. Terminal symbols are in bold. The non terminal symbols on the right-side of rules which are enclosed inside a pair of brackets are considered optional.

R1 : *F : Formula → [E_1 : Expression]* ***OP*** *E_2 : Expression*
m(F) : m(E_1) m(OP) m(E_2) L(OP, E_1) and R(OP, E_2)
R2 : *F : Formula →E_1 : Expression* ***HFB*** *E_2 : Expression*
m(F) : {m(E_1)}m(HFB){m(E_2)} U(HFB, E_1) and D(OP, E_2)
R3 : *F : Formula →* ***SP*** *[E_1 : Expression][E_2 : Expression]*
m(F) :m(SP)_{m(E_1)}^{m(E_2)} E_3 : Expression
m(E_3) (U(SP, E_1) or S(SP, E_1)) and (D(SP, E_2) or s(SP, E_2)) and R(SP, E_3)
R4 : *F : Formula →* ***IS****[E_1:Expression][E_2:Expression]*
m(F) :m(IS)_{m(E_1)}^{m(E_2)} E_3:Expression ***d*** *E_4:Expression*
m(E3)\dm(E_4) (U(IS, E_1) or S(IS, E_1)) and (D(IS, E_2) or s(IS, E_2)) and R(IS, E_3) and R(d,E_4)
R5 : *F : Formula →* ***RS*** *E_1 : Expression [E_2 : Expression]*
m(F):m(RS)[m(E_2)]{m(E_1)} I(RS, E_1) and S(RS, E_2)
R6 : *F : Formula → E_1 : Expression [E_2 : Expression]*
m(F) : m(E_1)_{m(E_2)}^{mI(E_3)} [E_3 : Expression]
s(E_1, E_2) and S(E_1, E_3)
R7 : *F : Formula →* ***GD_1*** *E : Expression* ***GD_2***
m(F) : m(GD_1) m(E) m(GD_2) R(GD_1, E) and L(GD_2, E)
R8 : *F : Formula →* ***SD_1*** *E : Expression* ***SD_2***
m(F) : m(SD_1) m(E) m(SD_2) R(SD_1, E) and L(SD_2, E)
R9 : *E : Expression → T : Term*
m(E) : m(T)
R10 : *E : Expression → F : Formula*
m(E) : m(F)
R11 : *T : Term →* ***CAR***
m(T) : m(CAR)
R12 : *T : Term → T : Term* ***CAR***
m(T) : m(CAR) R(T, CAR)

Consider the following example of a mathematical formula analysis:



F : Formula

R3 :m(F) : \sum_{m(E_1)}^{m(E_2)}m(E_3)

R3 :m(F) : \sum_{m(E_1)}^{m(E_2)}m(E_3)

R9, R11 : m(E_2) : n
m(F) : \sum_{i=1}^{n}m(E_3)

R1 : m(E_3) : m(E_{31})-m(E_{32})
m(F) : \sum_{i=1}^{ n} m(E_{31})-m(E_{32})

R5 : m(E_{31}) :
\sqrt[]{m(E_{311})}
m(F) :\sum_{i=1}^{n}
\sqrt[]{m(E_{311})}-m(E_{32})

R2 : m(E_{311}) : {m(E_{3111})}\frac{m(E_{3112})
m(F):\sum_{i=1}^{ n} \sqrt[]{{m(E_{3111})}
\frac{m(E_{3112})}}-m(E_{32})

R6 :
m(E_{3111}):m(E_{31111})_{I(E_{I31112})}
m(F):\sum_{i=1}^{n}
\sqrt[]{m(E_{31111}) -
{m(E_{31112})}}\frac{m(E_{3112})}}-
m(E_2)

R9, R12, R11 : m(E_{31111}) : ax
R9, R11 : m(E_{31112}) : i
m(F) :\sum_{i=1}^{n}
\sqrt[]{{ax_{i}}
\frac{m(E_{3112})}}-m(E_2)

R9, R11 : m(E_{3112}) : b
m(F) :\sum_{i=1}^{n}
\sqrt[]{{ax_{i}}\frac{ b}}-m(E_{32})

R9, R11 : m(E_{32}) : c
m(F) : \sum_{i=1}^{ n}
\sqrt[]{{ax_{i}}

Fig. 2.. Example of a mathematical formula analysis

IV. MATHEMATICAL FORMULA ENCODING

Once mathematical formulas are extracted then analysed, we aims to encode them for computing processing or electronic communication. Perhaps the most important influence on mathematical markup languages on the last two decades is the TeX typesetting system developed by D. KNUTH in 1986. TeX is a de facto standard in the mathematical research community, and it is pervasive in the scientific community at large. TeX sets a standard for quality of visual rendering quality. Moreover, because of the many legacy documents in TeX, and because of the large authoring community versed in TeX, we have used Latex to encode the mathematical formulas (see the example of a mathematical formula analysis).

MathML is a powerful new language for encoding mathematics. Its original purpose was to provide a way for marking up mathematics in Web pages, since HTML provided no easy way of handling it. However, because of the enormous influence of the Web on all types of communication, MathML has begun to influence the way mathematics are shared between all kinds of math and science applications. MathML is similar in many ways to earlier encoding for math expressions However, the things that sets MathML apart is the amount of information encoded. Most previous encoding stored only information about what an equation looked like. By contrast, MathML stores information about the logical structure and meaning of equations as well as their appearance. The philosophy of storing information about structure as well as appearance is very much a part of the Web viewpoint. Because of the reasons cited above, a priority in the design of MathML was the ability to convert TeX math input into MathML format. We present here the MathML output for the example of figure 2. The final step consists of applying an OCR engine to recognise text, converting the document in XML format and integrating the result of MathML formula conversion

```
<mrow>
 <munderover> /* lower bound */
  <mo>&Sum;</mo> /* a summation operator */
  <mrow>
   <mi>i</mi> /* an identifier */
   <mo>=</mo> /* equal operator */
   <mn>1</mn> /* a number */
  </mrow>
   <mi>n</mi> /* an identifier */
 </munderover> /* upper bound */
 <msqrt>  /* square root */
  <mfrac> /* fraction */
   <mrow>
    <mi>a</mi> /* identifier */
    <mo>&InvisibleTimes;</mo> /* implicit multiplication */
    <msub>
     <mi>x</mi>  /* an identifier */
     <mi>i</mi>  /* an identifier */
    </msub>
   </mrow>
   <mi>b</mi> /* an identifier */
  </frac>
 </msqrt>
 <mo>-</mo> /*substraction operator */
 <mi>c</mi> /* an identifier */
</mrow>
```

V. CONCLUSION AND FUTURE WORKS

This paper has discussed the use of fuzzy logic for the extraction of formulas in mathematical documents. Though a satisfactory rate of extraction is obtained, more research is still required to be able to attain human-like performance. We have then proposed and demonstrated a method for parsing mathematical formulas using a coordinate grammar. The overall system has shown its efficiency on a number of practical mathematical formulas. But, further work is required to extend this method to more complex formulas and documents and confirm efficiency and performance of our method using a large database. We have finally tried to encode formulas in MathML so as to develop mathematics on the Web which facilitate automatic processing, searching and indexing, and reuse in other mathematical applications.

REFERENCES

[1] R.H. Anderson, "Two-Dimensional Mathematical Notation", In proceedings of Syntactic Pattern Recognition Applications, K.S. Fu, Ed. Springer Verlag, NewYork, 1977, pp. 147-177.

[2] A. Belaïd, and J. P. Haton, "A syntactic Approach for Handwritten Mathematical Formula Recognition", In IEEE Trans. PAMI, vol. 6. n°1, 1984, pp. 105-111.

[3] A. Grbavec and D. Blosteïn, "Mathematics Recognition Using Graph Rewriting", In Proceedings of ICDAR'93, France, 1995, pp. 417-421.

[4] K.-F. Chan and D.-Y. Yeung, "Mathematical Expression Recognition: A survey", Technical report HKUST-CS99-04, April 1999.

[5] H. J. Lee and M. C. Lee, "Understanding Mathematical Expression in a Printed Document", In Proceedings of ICDAR'93, Japan, 1993, pp. 502-505.

[6] M. Okamoto and B. Miao, "Recognition of Mathematical Expressions by Using the Layout Structures of Symbols", In Proceedings of ICDAR'91, France, 1991, pp. 242-250.

[7] H. M. Twaakyondo and M. Okamoto, "Structure Analysis and Recognition of Mathematical Expressions", In Proceedings of ICDAR'95, Canada, 1995, pp. 430-437.

[8] J. Ha, R. M. Haralick, and I. T. Phillips, "Understanding mathematical expressions from document images", In Proceedings of ICDAR'95, Canada, 1995, pp. 956-959.

[9] Z. X. Wang and C. Faure, "Structural analysis of handwritten mathematical expressions", In Proceedings of ICPR'88, Washington, 1988, pp. 32-34.

[10] S. Lavirotte and L. Pottier, "Optical formula recognition", In Proceedings of ICDAR'97, Germany, 1997, pp. 357-361.

[11] M. Okamoto and A. Miyazawa, "An experimental implementation of a document recognition system for papers containing mathematical expressions", In H.S. Baird et al (ed). Structured Document Image Analysis Springer-Verlag, 1992, pp. 36-51.

[12] H. J. Lee and J. S. Wang, "Design of mathematical expression recognition system", In Proceedings of ICDAR'95, Canada, 1995, pp. 1084-1087.

[13] R. Fateman, T. Tokuyasu, B. Berman and N. Mitchell, "Optical Character Recognition and Parsing of Typeset Mathematics", In J. of Visual Commun. And Image Representation vol 7 no. 1, March 1996, pp. 2-15.

[14] K. Inoue R. Miyazaki, and M. Suzuku, "Optical recognition of printed mathematical documents", In Proceedings of ATCM'98, 1998.

[15] J.-Y Toumit, S. Garcia-Salicetti, H. Emptoz, "A Hiearachical and RECURSIVE Model of Mathematical Expressions for Automatic Reading of Mathematical Documents", In Proceedings of ICDAR'99, India, 1999, pp. 116-122.

[16] A. Kacem, A. Belaïd, and M. Ben Ahmed, "EXTRAFOR : Automatic EXTRAction of mathematical FORmulas", In Proceedings of ICDAR'99, Bangalore-India, 1999, pp. 527-530.

[17] A. Kacem, A. Belaïd, and M. Ben Ahmed, "Extracteur de formulas de documents mathématiques", In Proceedings of CIFED'00, Lyon-France, 2000, pp. 295-304.

[18] A. Kacem, A. Belaïd, and M. Ben Ahmed, "Automatic extraction of printed mathematical formulas using fuzzy logic and propagation of context", To appear in IJDAR : International Journal on Document Analysis and Recognition.

Visual Parsers based on Extended Constraint Multiset Grammars

Jiro Tanaka
Institute of Information Sciences and Electronics
The University of Tsukuba
Tsukuba, Ibaraki 305-8573, Japan
jiro@is.tsukuba.ac.jp

Abstract— **We are working for visual paring in these years. We have developed the series of visual parser generators, such as Eviss, VIC and Rainbow. They generate a spatial parser by defining the grammars of visual language. Using generated spatial parser, they can analyze the figures and can execute the specified actions. The GUI generator and the subset of VISPATCH are shown as examples.**

Keywords— **Graphical user interface, Visual parser, Visual system, Constraint solving**

I. INTRODUCTION

THOUGH the textual paring has already been well-established in computer science field, the visual paring is still in the preliminary stage.

Visual languages are used in various fields, such as ER diagrams, object diagrams of OMT, formula, music, and diagrams that show relationships between characters appearing in TV dramas. Visual languages have structures like textual languages.

We can assume the diagrams which represent graph structures as a visual language. A special purpose graphic editor can be considered as a system to process a visual language. We call a system which processes a visual language a *visual system*. Visual systems proposed so far have been fixed on certain specifications. It was a difficult and time consuming job to modify those systems.

Therefore, we are working for general purpose visual systems in these years. We have developed the series of visual parser generators, such as Eviss [1], [2], VIC [3] and Rainbow [4], [5].

II. THE EXTENDED CONSTRAINT MULTISET GRAMMARS

We use the extended Constraint Multiset Grammars (CMG) [6], [7] in defining the grammars of visual system in visual parser generators. CMG consist of a set of terminal symbols, a set of non-terminal symbols, a distinctive start symbol, and a set of production rules. The terminal and non-terminal symbols have various attributes. The production rules are used to rewrite a multiset of tokens (the instances of the terminal or non-terminal symbols) for a new symbol.

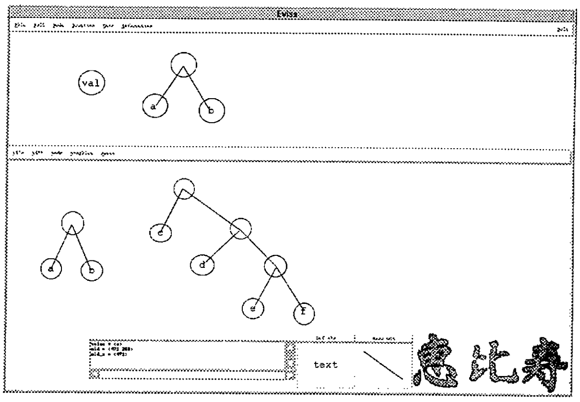


Fig. 1. A snapshot of Eviss

The constraints maintain the relationships between the attributes of the tokens. A production rule is defined as follows:

$$
\begin{aligned}
T(\vec{x}) \quad ::= \quad & T_1(\vec{x_1}), \cdots, T_n(\vec{x_n}) \ \mathit{where} \\
& \mathit{exists}\ T'_1(\vec{x'_1}), \cdots, T'_m(\vec{x'_m}) \\
& \mathit{where}\ C\ \mathit{and}\ \vec{x} = F\ \mathit{and}\ A
\end{aligned}
$$

When the attributes of the tokens $T_1, \cdots, T_n$ ("normal" components) and $T'_1, \cdots, T'_m$ ("exist" components) satisfy the constraints C, the tokens $T_1, \cdots, T_n$ are rewritten to the non-terminal symbol T. Exist components are needed to recognize T and are not rewritten to T [2]. F is the function that has the attributes $\vec{x_1}, \cdots, \vec{x_n}$ and $\vec{x'_1}, \cdots, \vec{x'_m}$ of the components as arguments, and the return value of the function is given to the non-terminal symbol T as its attribute.

Note that we have extended the original CMG to include action A. A is defined as "script program executed when the production rule is applied." In the extended CMG, we can specify arbitrary actions, such as computing values and rewriting figures.

---

[1] This paper has been extracted from: Jiro Tanaka, "Visual Parsing and 3D Visual Interface," *Proceedings of 2000 International Conference on Information Society in the 21st Century (IS2000)*, 2000.

[2] CMG also has "not_exist" and "all" components. For details, refer to [2], [6], [7].

III. LIST TREE EXAMPLE

List Tree is defined recursively by the following two production rules.

Rule 1: A non-terminal symbol "list" consists of a "circle" and a "text" in the center of it.

Rule 2: A non-terminal symbol "list" consists of a "circle," two "lines" and two "lists." The two "lists" are connected to the "circle" by the "lines."

These production rules can be written by the extended CMG as follows.

```
1:  list(point mid, integer mid_x,
2:                         string value) ::=
3:     C:circle, T:text
4:     where (
5:        C.mid == T.mid
7:     ) and {
8:        mid = C.mid
9:        mid_x = C.mid_x
10:       value = {script.string {
11:                        list @T.text@}}
12:    } and {
13:       display(value = @value@)
14:    }
15:
16: list(point mid, integer mid_x,
17:                        string value) ::=
18:    C:circle
19:    exists S1:list, S2:list,
20:                       L1:line, L2:line
21:    where (
22:       S1.mid == L1.end
23:       S2.mid == L2.end
24:       C.mid == L1.start
25:       C.mid == L2.start
26:       C.mid == T.mid
27:       S1.mid_x < S2.mid_x
28:    ) and {
29:       mid = C.mid
30:       mid_x = C.mid_x
31:       value = {script.string {
32:                  concat [list @S1.value@]
33:                         [list @S2.value@]}}
34:       lef = C.lu_x
35:       right = C.rl_x
36:    } and {
37:       display(value = @value@)
38:    }
```

Lines 1 to 14 show the definition of the production rule 1. Line 1 shows the attributes of the non-terminal symbol "list." Attributes are "mid," "mid_x," and "value." Line 3 shows that this non-terminal consists of a "circle" and a "text" string and these components are "Normal." At line 5, constraints are defined. This line shows that the attribute "mid" of "circle" C is equal to the attribute "mid" of "text" T. "mid" is an attribute that indicates the center's coordinates. In lines 8 to 11, the values of Attributes are defined. Line 8 shows that an attribute "mid" of "list" is equal to "mid" of "circle" C. Line 9 shows that an attribute "mid_x" of "list" is equal to "mid_x" of "circle" C. "mid_x" is an attribute indicating an abscissa. Lines 10 and 11 show the definition of an attribute "value." This definition represents that "text" string of "text" T is treated as a list. At the line 13, action is defined. This line shows output (*value*) when this production rule is applied.

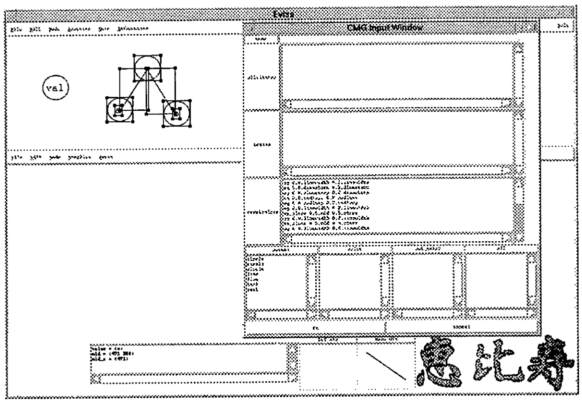


Fig. 2. Defining grammars in Eviss

Lines 16 to 38 show the definition for the production rule 2. Lines 19 and 20 show that two "lists" and two "lines" must exist somewhere in the visual sentence. Line 27 shows that "list" S1 is on the left side of "list" S2. This constraint distinguishes the left "list" from the right "list." Lines 31 to 35 show the definition of the attribute "value." Here, two "lists" S1 and S2 are connected.

IV. EVISS

We made the visual system Eviss [1], [2] which has a spatial parser generator. Figure 1 shows the snapshot of a visual system that represents "List Tree." The upper half of the screen is called the *definition window.* A person who implements a visual system defines grammars of visual languages in the definition window. The bottom half is called the *execution window.*

In Eviss, figures are used to define rough grammars. At first, the user draws figures which he wants to define as a new non-terminal symbol from the definition window. We call these figures as "example figures." Eviss automatically extracts simple constraints and components from "example figure" and outputs to the CMG Input Window with text. Then the user edits the constraints and components in the CMG Input Window (Figure 2). The user can also specify actions in this phase.

At the *execution phase,* a user draws figure elements which should be analyzed to the execution window.

V. EXAMPLES OF MAKING VISUAL SYSTEMS IN EVISS

A. The GUI Creator

We define expressions for widgets and binding in the visual language. A frame widget (`Frame`) is repre-

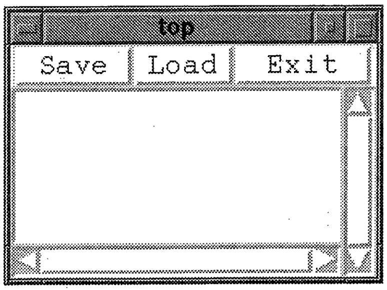


Fig. 3. An example of GUI.

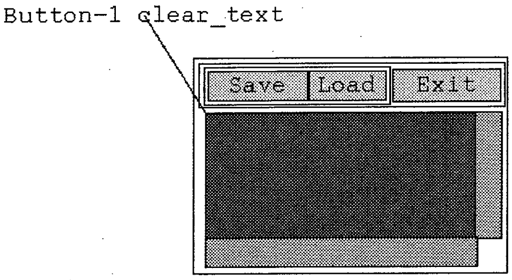


Fig. 4. Visual program that represents Figure 3.

sented by a rectangle which is not painted. A scroll bar widget (`Scroll`) is represented by a rectangle painted with orange. A text widget (`TextW`) is represented by a rectangle painted with red. A button widget (`Button`) is represented by a rectangle painted with yellow and a text string in it.

We show an example of describing a GUI in the visual language. Figure 3 is an example of a GUI and Figure 4 is the visual program which represents it. Suppose that the procedure `clear_text` is called when left button of the mouse is clicked on the text widget in Figure 3. The binding appears in the visual program though it does not actually appear on the screen. This is because GUIs do not consist of only visual informations.

We have defined production rules for creating a GUI by combining widgets. The production rule for `Binding` has a text string and a line as its components. In figure 4, the text string is "`Button-1 clear_text`" and the `GUI` is the text widget. The text string is a list which consists of the name of the event and the name of the procedure which is called when the event occurs. In Figure 4, the event is `Button-1` and the name of the procedure is `clear_text`.

B. The subset of VISPATCH

VISPATCH [8] is a visual system which redraws figures according to rules represented by figures. Redrawing is started by events caused by users or the system such as mouse clicks and drags.

We have implemented the subset of VISPATCH in

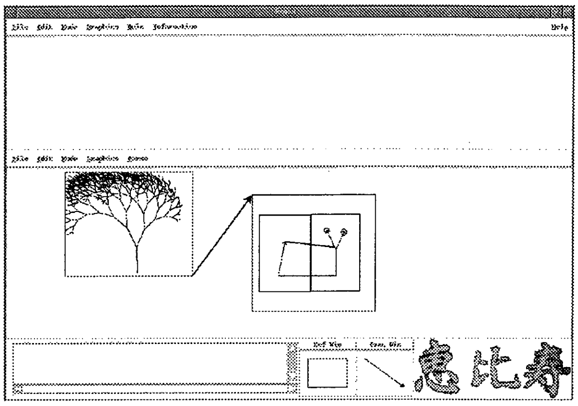


Fig. 5. Snapshot of VISPATCH implemented in Eviss.

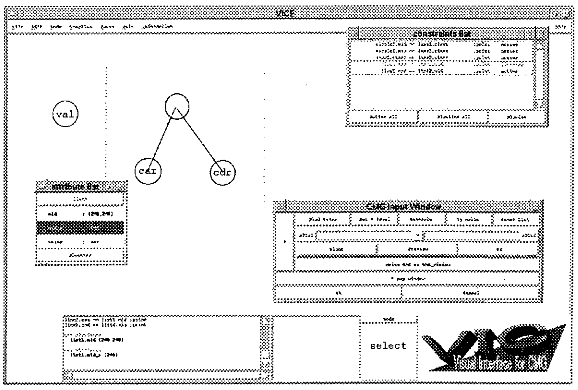


Fig. 6. A snapshot of VIC

Eviss (Figure 5). VISPATCH starts redrawing by events, in other words, VISPATCH is *event driven.* Eviss starts spatial parsing (and redrawing) by drawing, deleting and altering figure elements, in other words, Eviss is *data driven.* To start spatial parsing, if an event occurs in the event sensor, a figure element that is the same as in the rule head is drawn in the event sensor. After spatial parsing is finished, a procedure which creates production rules for VISPATCH rules is called in action.

VI. VIC

VIC [3] is the successor of Eviss. There are two main differences between Eviss and VIC. The first difference is that Eviss has two windows, i.e., Definition Window and Execution Window, and only Execution Window has a spatial parser. Whereas, VIC has only one window. Because of having one window only, VIC has no border between Definition Window and Execution Window. VIC can understand the non-terminal symbols when the user defines the grammars.

The second difference is that VIC can define constraints by the direct manipulation of "example figures" without using CMG Input Window. In Eviss, user had to input CMG textually from CMG Input

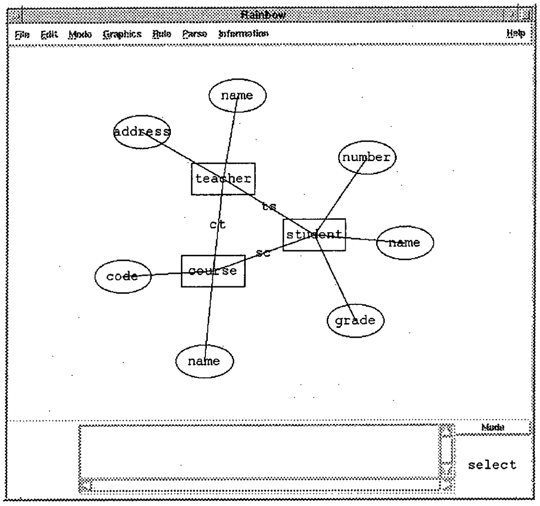


Fig. 7. Example of the E-R diagrams

Window. This made difficult to define the visual system intuitively. In the case of VIC, the user can define various visual systems intuitively, even if the user does not know the grammar of CMG.

The snapshot of VIC is shown in Figure 6.

VII. RAINBOW

The visual system must have layout capability, since it performs actions such as creating, deleting, and moving the figures. Even if a user lays out a portion of the figure, the entire figure can be hard to understand. It is important to make the entire figure more balanced and understandable.

We therefore developed Rainbow[4], [5], a visual system generator that can handle layout constraints. The system can interactively layout whole figures while parsing them, and make the parsed figures more balanced and understandable. Rainbow was implemented by adding the layout constraints to Eviss.

Using Rainbow makes it possible to more interactively handle figures, such as the various diagrams that are used in the software engineering field. Figures can be interactively laid out while they are parsed by adding layout capability to their spatial parser, and the parsed figures are more understandable. Rainbow is therefore a useful tool for making CASE tools.

The snapshot of Rainbow in the E-R diagrams example is shown in Figure 7.

VIII. ACKNOWLEDGMENTS

The author would like to express thanks to Akihiko Baba, SackTae Joung, Kenichirou Fujiyama and Kazuhisa Iizuka for their cooperation in the project. The work reported in this paper has been based on their research efforts.

REFERENCES

[1] Akihiro Baba and Jiro Tanaka: "A Visual System Having a Spatial Parser Generator," *Transactions of IPSJ*, Vol. 39, No. 5, pp. 1385–1394, 1998, *in Japanese*.

[2] Akihiro Baba and Jiro Tanaka: "Eviss: a Visual System Having a Spatial Parser Generator," *Proceedings of Asia Pacific Computer Human Interaction*, pp. 158–164, 1998.

[3] Kenichirou Fujiyama, Kazuhisa Iizuka and Jiro Tanaka, "VIC: CMG Input System Using Example Figures," *Proceedings of the International Symposium on Future Software Technology*, pp. 67–72, 1999.

[4] SackTae Joung and Jiro Tanaka: Rainbow: "Implementing Layout Constraints in Visual System Generator," *Transactions of IPSJ*, Vol.41, No.5, pp. 1246-1256, 2000, *in Japanese*.

[5] SackTae Joung and Jiro Tanaka: "Generating a Visual System with Soft Layout Constraints," *Proceedings of the International Conference on Information – Information'2000 –*, Fukuoka, Japan, Oct. 16-19, pp. 138-145, 2000.

[6] Kim Marriott: "Constraint Multiset Grammars," *Proceedings of the IEEE Symposium on Visual Languages*, pp. 118–125, 1994.

[7] Sitt Sen Chok and Kim Marriott: "Automatic Construction of User Interfaces from Constraint Multiset Grammars," *Proceedings of the IEEE Workshop on Visual Languages*, pp. 242–249, 1995.

[8] Yasunori Harada, Kenji Miyamoto and Rikio Onai: "VISPATCH: Graphical rule-based language controlled by user event," *Proceedings of the 1997 IEEE Symposium on Visual Languages*, pp. 162–163, 1997.

Numerical Methods for the Limit Load Analysis of Nonhomogeneous Elastoplastic Structures

Adnene Elyacoubi
The 7 November University at Carthage
Polytechnic School of Tunisia
29 Rue Asdrubal, 1002 Tunis
Tunisia
adnene.elyacoubi@ept.rnu.tn

Taieb Hadhri
The 7 November University at Carthage
Polytechnic School of Tunisia
29 Rue Asdrubal, 1002 Tunis
Tunisia
taieb.hadhri@ept.rnu.tn

Abstract— **In this work we consider a solid body $\Omega \subset R^3$ constituted by a nonhomogeneous elastoplastic material, submitted to a density of volumic forces λ f and a density of forces λg acting on the boundary where λ is the loading parameter. The problem is to determine, in the case of a nonbounded convex of elasticity, the Limit load denoted by $\bar{\lambda}$ beyond which there is a break of the structure. Then, assuming that the convex of elasticity in point x of Ω, denoted by K(x), is written in the form of K^D(x)+RI, I is the identity of $R^9{}_{sym}$, where the spherical component K^D is bounded regardless of x$\in \Omega$, we show under the condition "Rotf $\neq$ 0" or "g is not colinear to the normal in a part of the boundary of Ω", that the Limit load $\bar{\lambda}$ searched equals the inverse of the infimum of the gauge of the convex of elasticity translated by stress field equilibrating the unitary load corresponding to $\lambda = 1$. We suppose, then, that Ω is written under the form of $\omega \times [0,1]$ where ω is a bounded open set of R^2. This allows us to replace the threedimensional problem by a bidimensional one. We introduce a numerical method to calculate the Limit load $\bar{\lambda}$, based on the finite nonconform elements of Morley [3]. In fact, we consider the operator $\tilde{A}$ which is the extension of the operator of ARy 2D by adding the component (3,3) constant by triangle of the triangularization. In the next step, we show that the gauge of the convex of plastic admissible constraints equals the norm $L^\infty(\omega)$ of the gauge of the local convex which sets a numeric difficulty given that the norm $\mathrm{L}^\infty(\omega)$ is not differentiable. In this way we replace $\mathrm{L}^\infty(\omega)$ by $\mathrm{L}^q(\omega)$ and the Limit load $\bar{\lambda}$ by $\bar{\lambda}_q$. Finally, in adopting the convex of Von Mises as convex of elasticity, we approach the numeric implementation and we elaborate a Fortran code for the calculation of the Limit load of a nonhomogeneous and periodic structure based on the Newton method. In the case of a nonhomegeneous and periodic structure with a cubic basic cellular of side $\epsilon > 0$ we denote by λ^ϵ the Limit load and show that $\lambda^\epsilon \longrightarrow \lambda^{\mathrm{hom}}$ when ϵ tends to 0 and λ^{hom} equals the inverse of the gauge of the homogeneised convex. In prospect, we approach λ^ϵ by λ^{hom} for ϵ near to zero.**

Keywords— **Elastoplasticity, Nonhomogeneous material, Limit load, Finite elements, Newton algorithm.**

I. CHARACTERIZATION OF THE LIMIT LOAD $\bar{\lambda}$

Consider a solid body $\Omega \subset R^3$ constituted by a nonhomogeneous elastoplastic material, submitted to a density of volumic forces λ f and a density of forces λg acting on the boundary where λ is the loading parameter. Assume that the convex of elasticity in every point x of Ω, denoted by K(x), is written in the form of K^D(x)+RI, I is the identity of $R^9{}_{sym}$, where the spherical component K^D is bounded regardless of x$\in \Omega$

A. Definition

Considering the functional H_1 defined on V_1 :

$$\mathrm{V}_1 = \left\{ \eta \in L^2(\Omega, R^9_{sym}) : \quad \begin{array}{c} \mathrm{div}\eta = 0 \text{ a.e in } \Omega \\ \eta \cdot n = 0 \text{ on } \Gamma_1 \end{array} \right\}$$

by :

$$H_1(\eta) = \mathrm{J}_{Kad}(\sigma^e - \eta)$$

where, J_{Kad} is the gauge function of the convex K_{ad} defined by :

$$K_{ad} = \{\eta \in L^2(\Omega, R^9_{sym});\ \eta(x) \in K(x) \text{ a.e in } \Omega\}$$

$$\mathrm{J}_{Kad}(\alpha) = \inf\{\mathrm{s} > 0 \text{ such that } \alpha \in \mathrm{s.K(x)} \text{ a.e in } \Omega\}$$

The limit load $\bar{\lambda}$ is defined in [2] by :

$$\bar{\lambda} = \sup\{\lambda > 0 \text{ such that } \mathrm{D}_\lambda \neq \emptyset\}$$

where

$$\mathrm{D}_\lambda = \{\sigma \in L^2(\Omega, R^9_{sym}) \text{ such that}$$

$$\left. \begin{array}{l} \mathrm{div}\ \sigma - \lambda f = 0 \quad \text{a.e in } \Omega \\ \sigma \cdot n \ \ = \lambda g \quad \text{on } \Gamma_1 \\ \sigma(x) \in \mathrm{K(x)} \text{ a.e in } \Omega \end{array} \right\}$$

On the other hand, we suppose that :

- (H_1) $\Gamma_1 \cup \Gamma_0 = \partial\Omega$, the boundary of Ω with mes$(\Gamma_1) \neq 0$ and the interiors of Γ_1 and Γ_0 satisfy:$\Gamma_1^0 \cap \Gamma_0^0 = \emptyset$

- (H_2) $K(x)$ is a closed convex part of R^9_{sym},

  $\exists \quad c > 0$ such that : $B(0,c) \subset K(x)$ a.e in Ω

- (H_3) $g \in (L^\infty(\Gamma_1))^3, f \in (L^4(\Omega))^3$

and :
$\exists \tilde{g} \in (L^\infty(\partial\Omega))^3$ such that :
$\int_\Omega f \mathrm{dx} + \int_{\partial\Omega} \tilde{g} \mathrm{d}\Gamma = 0$ and $\tilde{g}/\Gamma_1 = g$

- (H_4) $K(x) = K^D(x) + R\,\mathrm{I}$ where I is the identity of R^9_{sym} and we assume that :$\exists \quad M > 0$ such that $K^D(x) \subset B(0,M)$ a.e in Ω.

We define the following set :

$$(L^2(\Omega))^9_s = L^2(\Omega, R^9_{sym})$$

B. Problem obtained by extension of Ω

We suppose that Ω satisfies :

$$\begin{cases} \Omega \subset \Omega_0, \text{ an open subset of } R^3 \\ \Omega_0 \text{ is convex ; } \Omega_0 \backslash \Omega \text{ is connex} \\ \partial\Omega \cap (\partial(\Omega_0 \backslash \Omega)) = \Gamma_1 \\ \textit{and} \\ \forall\, \phi \in \mathrm{H}^1(\Omega), \exists \phi_1 \in \mathrm{H}^1(\Omega_0 \backslash \Omega) \text{ such that :} \\ \phi = \phi_1 \text{ on } \Gamma_1 \end{cases} \quad (1)$$

Theorem 1:
We suppose that hypothesis $(H_1), (H_2), (H_3)(H_4)$ and property (1) are satisfied, then we have :
there exists $\sigma_1 \in V_1$ such that

$$H_1(\sigma_1) = \inf_{\eta \in V_1} H_1(\eta)$$

in this way the infimum of H_1 is reached on V_1.

Corollary 1:
Under the hypothesis of the theorem above, we have :
If Rotf $\neq 0$ in Ω or
$\exists$ B $\subset \Gamma_1$, mes(B) $\neq 0$ such that :

$$\forall x \in R, g \wedge n \neq 0$$

that is to say that g is not colinear to the unit normal exterior vector on the boundary of Ω, then :

$$\inf_{\eta \in V_1} H_1(\eta) \neq 0$$

and

$$\exists\, \sigma_1 \in V_1 \text{ such that } \bar{\lambda} = \frac{1}{\inf\limits_{\eta \in V_1} H_1(\eta)} = \frac{1}{H_1(\sigma_1)}$$

II. A HENCKY DISPLACEMENT PROBLEM UNDER PLANE DEFORMATION HYPOTHESIS

Now we suppose that $\Omega = \omega \times]0,1[$ where ω is an open subset of R^2, considering the case of plane deformations and of a null displacement following the axis (ox_3) and we suppose that the density of energy of deformations ψ_x, as well as the loading, the limit conditions and the mechanical behaviour of the material are independent of x_3.
Given $\gamma = \partial\omega$ and $\gamma_1 \cup \gamma_0 = \gamma$ a partition of γ.
Setting :

- $\Gamma_1 = \gamma_1 \times]0,1[$
- $\Gamma_0 = \gamma_0 \times]0,1[$
- $\mathrm{V} = \left\{ \tilde{\mathrm{v}} \in (\mathrm{H}^1(\Omega))^3 \text{ such that :} \{ \begin{array}{l} \tilde{\mathrm{v}}_3 = 0 \text{ in } \Omega \\ \tilde{\mathrm{v}}_{1,3} = \tilde{\mathrm{v}}_{2,3} = 0 \text{ in } \Omega \end{array} \right\}$
- $L(\tilde{\mathrm{v}}) = \int_\Omega f\tilde{\mathrm{v}} \mathrm{dx} + \int_{\Gamma_1} g\tilde{\mathrm{v}}\, \mathrm{d}\Gamma$

According to [2], the displacement problem is the following :

$$\inf_{\tilde{v} \in \mathrm{V}} \left(\int_\Omega \psi_x(\varepsilon(\tilde{\mathrm{v}}))(x) \mathrm{dx} - \lambda L(\tilde{\mathrm{v}}) \right)$$

and the Limit load $\bar{\lambda}$ is given by :

$$\bar{\lambda} = \sup \{ \lambda > 0 \text{ such that }$$

$$\inf_{\substack{\tilde{\mathrm{v}} \in \mathrm{V} \\ \tilde{\mathrm{v}} = 0 \text{ on } \Gamma_0}} \int_\Omega \psi_x(\varepsilon(\tilde{\mathrm{v}}))(x) \mathrm{dx} - \lambda L(\tilde{\mathrm{v}}) \quad > -\infty \}$$

Using the fact that $\Omega = \omega \times]0,1[$, we have :

$$\bar{\lambda} = \sup \left\{ \lambda > 0; \quad \inf_{\substack{\mathrm{v} \in (\mathrm{H}^1(\Omega))^2 \\ \mathrm{v} = 0 \text{ on } \Gamma_0}} (A(v)) > -\infty \right\} \quad (2)$$

where :

$$A(v) = \int_\omega \psi_x(\varepsilon(\mathrm{v}))(x) \mathrm{dx}_1\mathrm{dx}_2 - \lambda \int_\omega f\mathrm{v}\; \mathrm{dx}_1\mathrm{dx}_2 - \lambda \int_{\gamma_1} g\mathrm{vd}\Gamma$$

After that we establish the following new characterization of the limit load $\bar{\lambda}$

Theorem 2:
Under the hypothesis mentioned above, as well as

Rot$f \neq 0$ we have :

$$\bar{\lambda} = \frac{1}{\inf\limits_{\eta \in \mathrm{V}_2} \mathrm{J}_{\mathrm{Kad}}(\sigma^e - \eta)}$$

where
V_2 is the space defined by :

$$\mathrm{V}_2 = \{\eta \in L^2(\omega, {R^9}_{sym}) \text{ such that :}$$

$$\left.\begin{array}{l} \eta_{13} = \eta_{23} = 0 \\ \mathrm{div}\eta = 0 \text{ in } \omega \\ \eta.n = 0 \text{ on } \gamma_1 \end{array}\right\}$$

$$\mathrm{div}\eta = (\frac{\partial \eta_{11}}{\partial x_1} + \frac{\partial \eta_{12}}{\partial x_2}, \frac{\partial \eta_{12}}{\partial x_1} + \frac{\partial \eta_{22}}{\partial x_2}, 0)$$

and

$$\sigma^e = \begin{pmatrix} (\sigma^e)_{11} & (\sigma^e)_{12} & 0 \\ (\sigma^e)_{21} & (\sigma^e)_{22} & 0 \\ 0 & 0 & (\sigma^e)_{33} \end{pmatrix}$$

is a solution of the problem :

$$\left\{\begin{array}{ll} \mathrm{div}\sigma^e + f = 0 & \text{in } \omega \\ \sigma^e.n = g & \text{on } \gamma_1 \end{array}\right. \qquad (3)$$

III. DISCRETIZATION OF THE LIMIT LOAD PROBLEM UNDER THE PLANE DEFORMATION HYPOTHESIS

In the following we suppose that :
Rot$f \neq 0$ and that the hypothesis $(H_1), (H_2), (H_3)$ and (H_4) are satisfied
Then we have the following results :

Lemma 1:

$$\inf_{\eta \in \mathrm{V}_2} \mathrm{J}_{\mathrm{Kad}}(\sigma^e - \eta) = \inf_{\eta \in \mathrm{V}_2} \|\mathrm{J}_{\mathrm{K(x)}}((\sigma^e - \eta)(x))\|_{L^\infty(\omega)}$$

Since the norm of $L^\infty(\omega)$ is not differentiable, we replace it by the norm of $L^q(\omega)$ and introduce the problem :

$$\inf_{\eta \in \mathrm{V}_2} \|\mathrm{J}_{\mathrm{K(x)}}((\sigma^e - \eta)(x))\|_{L^q(\omega)}$$

with $q \geq 2$, then we study the limit of this infimum when q goes to infinity.
We prove the following convergence result.

Theorem 3:
Under the hypothesis of this section, we have :

$$\bar{\lambda}_q \text{ converges to } \bar{\lambda} \text{ when q goes to } +\infty$$

where :

$$\bar{\lambda}_q = \frac{1}{\inf\limits_{\eta \in \mathrm{V}_2} \|\mathrm{J}_{\mathrm{K(x)}}((\sigma^e - \eta)(x))\|_{L^q(\omega)}}$$

and

$$\bar{\lambda} = \frac{1}{\inf\limits_{\eta \in \mathrm{V}_2} \|\mathrm{J}_{\mathrm{K(x)}}((\sigma^e - \eta)(x))\|_{L^\infty(\omega)}}$$

A. Extended Airy operator

Consider, the following space :

$$E = H^2(\omega) \times L^2(\omega)$$

and

$$M^0(\gamma_1) = \{(\beta, \beta_1) \in E \text{ such that } \beta = \frac{\partial \beta}{\partial n} = 0 \quad \text{on } \gamma_1\}$$

We define the extended Airy operator $\tilde{A}$ by :

$$\tilde{A}(\beta, \beta_1) = \begin{pmatrix} \frac{\partial^2 \beta}{\partial y^2} & \frac{-\partial^2 \beta}{\partial x \partial y} & 0 \\ \frac{-\partial^2 \beta}{\partial x \partial y} & \frac{\partial^2 \beta}{\partial x^2} & 0 \\ 0 & 0 & \beta_1 \end{pmatrix}$$

After that, we prove the following lemma :

Lemma 2:
For all $\eta \in \mathrm{V}_2$, we can find $(\beta, \beta_1) \in \mathrm{E}$ such that :

$$\eta = \tilde{A}(\beta, \beta_1)$$

and conversely, for all $(\beta, \beta_1) \in E$, we have :

$$\tilde{A}(\beta, \beta_1) \in V_2$$

B. Description of the triangulation and the adopted discreet spaces

We consider a triangulation of ω in triangles :
$\omega = \cup_{T \in \mathcal{T}_h} T$
where, the intersection of two triangles T_i is either a summit, or an edge or the empty set.
We introduce :
${M_h}^0 = \{(\beta, \beta_1) : \omega \to R^2$ such that

$$\beta_{/T} \in P_2(T), \beta_1 \text{ is a constant function on } T$$

$$\beta_{/T_i}(S) = \beta_{/T_j}(S) \text{ and } [\frac{\partial \beta}{\partial n}](M_i) = 0\}$$

where :
S is a common summit of T_i and T_j
M_i is the center of the side T_i and n is the unit normal vector to T_i
and

$$M_h{}^0(\gamma_1) = \{(\beta, \beta_1) \in M_h{}^0 \text{ such that}$$

$$\beta(S) = \frac{\partial \beta}{\partial n}(M_i) = 0$$

$$\text{for all summit S of } T_i \subset \gamma_1$$

$$\text{and center } M_i \text{ of } T_i \subset \gamma_1\}$$

C. Extended Morley Finite Elements

Definition 1:
For the discretization of β, we consider the following finite elements:
(1) The degrees of freedom are the values of the function in the summits of the triangle and the values of the normal derivative at the center points of the sides of the triangle .
(2) the space P_T of form functions is the space P_2 of polynom with degree 2.
and for β_1,we consider :
the degrees of freedom are the values of the choosen constant function by triangle.

Lemma 3: Each function P of P_T is entRely determined in a unique manner, by the six degrees of freedom $(P_i)_{1 \leq i \leq 6}$ given by the above definition and we have:

$$P(x,y) = \sum_{1 \leq i \leq 3} P_i \phi_i(x,y) + \sum_{1 \leq i \leq 3} \frac{\partial P}{\partial n_i} \psi_i(x,y)$$

where the basic function ψ_i and ϕ_i are given by :

$$\psi_i(x,y) = \frac{2\text{mes(T)}}{l_i} \chi_i(\chi_i - 1) \text{ and this for } 1 \leq \ i \ \leq 3 \tag{4}$$

and

$$\phi_i(x,y) = \chi_i{}^2 + \sum_{\substack{1 \leq \ i \ \leq 3 \\ j \neq i}} \alpha_j{}^{(i)} \psi_j \text{ fo all } 1 \leq \ i \ \leq 3 \tag{5}$$

where χ_i are the barycentric coordinates according to the summits of T, mes(T), l_i and $\alpha_j{}^{(i)}$ are characteristic parameters relieted to T [3]
Noticing that :
for $(\beta, \beta_1) \in M_h{}^0(\gamma_1)$ we have $\tilde{A}(\beta, \beta_1) \in \text{V}_2$
we introduce the following discretized problem :

$$\inf_{(\beta,\beta_1) \in M_h{}^0(\gamma_1)} \frac{1}{\text{mes}(\omega)^{\frac{1}{q}}} A_K$$

where

$$A_K = \|\text{J}_{\text{K(x)}}(\sigma^e, \tilde{A}(\beta, \beta_1))(x)\|_{L^q(\omega)}$$

D. Discreet expression of the functional to minimize

In order to simplify the numerical problem, we calculate the expression :

$$\inf_{(\beta,\beta_1) \in M_h{}^0(\gamma_1)} \frac{1}{\text{mes}(\omega)} (A_K)^q$$

in place of :

$$\inf_{(\beta,\beta_1) \in M_h{}^0(\gamma_1)} \frac{1}{(\text{mes}(\omega))^{\frac{1}{q}}} A_K$$

This leads us to the following problem :

$$\inf_{(\beta,\beta_1) \in M_h{}^0(\gamma_1)} H_{q,h}(\beta, \beta_1)$$

where :

$$H_{q,h}(\beta, \beta_1)$$

$$=$$

$$\frac{1}{\text{mes}(\omega)} \sum_{T \in \tau_h} \frac{1}{(3^{\frac{1}{2}} k_1)^q} \int_T (|\sigma^e{}_h - \tilde{A}_h(\beta, \beta_1)|_G{}^2)^{\frac{q}{2}} \text{ dx}$$

and
$|\eta|_G = (< \eta, \eta >_G)^{\frac{1}{2}}$
where
$< \eta, \eta^1 >_G = \frac{2}{3}(\eta_{11}\eta^1{}_{11} + \eta_{22}\eta^1{}_{22} + \eta_{33}\eta^1{}_{33}) - \frac{1}{3}(\eta_{11}\eta^1{}_{22} + \eta^1{}_{11}\eta_{22} + \eta_{11}\eta^1{}_{33} + \eta^1{}_{11}\eta_{33} + \eta^1{}_{22}\eta_{33} + \eta_{22}\eta^1{}_{33}) + 2\eta_{12}\eta^1{}_{12}$

IV. NEWTON ALGORITHM FOR THE CALCULATION OF THE LIMIT LOAD

We use Newton algorithm to minimize the function

$$\eta = (\beta, \beta_1) \longrightarrow \text{H}_{q,h}(\eta)$$

that is written :

$$\begin{cases} \eta_0 \text{ given} \\ \eta_{n+1} = \eta_n - (\nabla^2 \text{H}_{q,h}(\eta_n))^{-1} \nabla \text{H}_{q,h}(\eta_n) \end{cases}$$

For that we proceed by the 7 following steps and use the stopping test given below :

Step1
Calculate σ^e and choose $\eta_0 \in R^N$ where :
$N = dim M_h{}^0(\gamma_1)$

Step2
Calculation of $\text{J}_{Kad}(\sigma^e - \eta_0)$

Step3
Calculation of the gradient and the Hessien of the functional $\text{H}_{q,h}(\eta)$ in η_0

Step4

Taking into account the limit conditions on γ_1, in this way we eliminate in $\nabla H_{q,h}(\eta_0)$ and $\nabla^2 H_{q,h}(\eta_0)$ the lines and the columns corresponding to the degrees of freedom located on the border of γ_1, by this method we reduce the size of the vector $DH_{q,h}(\eta_0)$ and that of the matrix $\nabla^2 H_{q,h}(\eta_0)$ which will facilate the next resolution.

Step5

Resolution of the system

$$\nabla^2 H_{q,h}(\eta_0)\delta = \nabla H_{q,h}(\eta_0)$$

This resolution is done by the Gauss method.

Step6

Recovery of the size of δ by injecting zeros in the coordinates corresponding to the degrees of freedom located on γ_1

Step7

We consider the vector $\eta_1 = \eta_0 - \delta$ and we calculate :

$$J_{Kad}(\sigma^e - \eta_1) \text{ as well as } \| \eta_0 - \eta_1 \|_\infty$$

Stopping test

If $\| \eta_0 - \eta_1 \|_\infty \leq 0.01$

and $| J_{Kad}(\sigma^e - \eta_1) - J_{Kad}(\sigma^e - \eta_0) | \leq 0.01$

The Limit load equals : $\dfrac{1}{J_{Kad}(\sigma^e - \eta_1)}$

STOP

Otherwise

We return to Step 2 and replace η_0 by η_1.

Of course to prevent the risk of extremely long calculations we limit the number of iterations.

V. PERSPECTIVES

We consider now a periodic structure with a cubic basic cellular of side $\epsilon > 0$, contituted by two different homogeneous materials, then we have

Theorem 4: Under the hypothesis of the last section and Rot$f \neq 0$ we have, for all sequence ϵ_i which tends to zero :

$$\lambda^{\epsilon_i} \longrightarrow \lambda^{\text{hom}} \text{ when } \epsilon_i \text{ tends to } 0$$

where :

$$\lambda^{\epsilon_i} = \frac{1}{\inf\limits_{\eta \in V_1} J_{Kad^{\epsilon_i}}(\sigma^e - \eta)}$$

$$\lambda^{\text{hom}} = \frac{1}{\inf\limits_{\eta \in V_1 \cap (L^\infty(\Omega))^9} J_{IK}(\sigma^e - \eta)}$$

We hope that λ^{hom} will be easyer to calculate than λ^ϵ for ϵ near to 0 but not equal to 0. It will be interesting to determinate the expression of the limit convex IK and we approach then λ^ϵ by λ^{hom}.

REFERENCES

[1] H.El Fekih, Hadhri.T "Calcul des charges Limites d'une structure élastoplastique en contraintes planes."RAIRO,M2AN, 1996

[2] R.Temam, Mathematical Problems in Plasticity, Bordas, Paris, 1985.

[3] L.S.D. Morley : The triangular equilibrium element in the solution of plate bending problems. Aero-Quart, 19, pp.149-169, 1968.

Generalizations of Multisets

Sadaaki Miyamoto
The University of Tsukuba
1-1-1 Tennodai Tsukuba City
Ibaraki 305-8573, Japan
`miyamoto@esys.tsukuba.ac.jp`

Abstract— **This paper aims at discussing two generalizations of multisets. First, a class of fuzzy multisets which have an infinite membership set for an element of the universe and finite cardinality is introduced. It is shown that the sum, union, intersection as well as most *t*-norm and conorm operations for two sets in this class except the drastic sum keep the property of the finite cardinality of the derived set. Second, the membership sequence is generalized to a closed set on the plane whereby both the fuzzy multiset and real-valued multisets are discussed within this unified framework.**

Keywords— **multiset, fuzzy set, real-valued multiset, *t*-norm.**

I. INTRODUCTION

Multisets, sometimes called bags, have been considered by many authors (e.g., [5], [7], [1]) and actively discussed with application to database queries (e.g., [2]). The discussion of real-valued multisets by Blizard [1] should be noted among various studies.

A generalization of multisets, i.e., fuzzy multisets have also been considered by several researchers [15], [4], [8], [13], [12], [9], [6]. An application of fuzzy multisets is information retrieval on Web, since an information item may appear more than once with possibly different degrees of relevance to a query [10].

The application to the information retrieval invokes interesting problems. Huge amount, almost infinite, of information items exists in the space of WWW. A query may search a very large number of the items wherefrom all information is unable to be obtained by human capability. We hence should content ourselves with observation of a small part of the obtained information pieces. Such experiences lead us to consideration of infinite fuzzy multisets. The infiniteness implies that although the information pieces may be finite but the number of information items is very large and there is no fixed upper bound to this number.

We are concerned with infinite fuzzy multisets in this paper. The infiniteness does not mean the universal space on which fuzzy multisets are discussed is infinite: the universe may or may not be finite. It means that a membership set for an element the universe is infinite even when the underlying crisp multisets cannot have infinite multiplicity.

A basic measure of a fuzzy set is the cardinality, which is the sum of membership values of a fuzzy set. We introduce a class of fuzzy multisets for which the cardinality is finite, and shows that most t-norm and conorm operations for two sets in this class keeps the derived set within this class.

Second generalization of multisets using a closed set of a plane is moreover considered where the interest is mainly theoretical. This generalization includes real-valued fuzzy sets [1], [11] and fuzzy multisets. We have moreover shown in another paper that there is another fuzzification of multisets using the fuzzy number [6]. A simple transformation put this fuzzification into the second generalization by the closed set.

II. PRELIMINARIES

Fuzzy sets

Assume throughout this paper that the universal set X is finite, since the cardinality of X does not concern the argument herein. We thus write $X = \{x_1, x_2, \dots, x_n\}$ or $X = \{x, y, \dots, z\}$. Scalars are denoted by $a, b, \dots$

Operations for fuzzy sets are extended to fuzzy multisets. We first provide notations for fuzzy sets. Infix notations for max and min are denoted by $\vee$ and $\wedge$, respectively as in most literature.

The t-norms and t-conorms [14] are discussed for fuzzy multisets. A t-norm and conorm are denoted by $t(\cdot,\cdot)$ and $s(\cdot,\cdot)$, respectively. Namely, for two real numbers $a, b \in I$ ($I = [0,1]$ is the unit interval), the results of applying the t-norm and conorm are $t(a,b)$ and $s(a,b)$. As there are various t-norms and conorms [14], [3], they are distinguished by the subscript;

1. **the standard min and max**:

$$t_S(a,b) = a \wedge b,\ s_S(a,b) = a \vee b;$$

2. **the algebraic sum and product**:

$$t_A(a,b) = ab,\ s_A(a,b) = a + b - ab;$$

3. **the bounded product and sum**:

$$t_B(a,b) = 0 \vee (a + b - 1),\ s_B(a,b) = 1 \wedge (a + b);$$

for other norms the definitions are omitted to save the space:

4. **the drastic product and sum**: $t_D(a,b)$, $s_D(a,b)$;

5. **the Frank norms**: $t_F(a,b)$, $s_F(a,b)$;
6. **the Hamacher norms**: $t_H(a,b)$, $s_H(a,b)$;
7. **the Yager norms**: $t_Y(a,b)$, $s_Y(a,b)$;
8. **the Sugeno norms**: $t_{SG}(a,b)$, $s_{SG}(a,b)$.

When the t-norms and conorms are applied to the set operation for fuzzy sets A and B, the infix notations $A\mathbf{T}B$ and $A\mathbf{S}B$ are used:

$$\mu_{A\mathbf{T}B}(x) = t(\mu_A(x), \mu_B(x)),$$
$$\mu_{A\mathbf{S}B}(x) = s(\mu_A(x), \mu_B(x)).$$

The different norms are distinguished by the subscripts: $A\mathbf{T_S}B$ and $A\mathbf{S_S}B$ for the standard min and max, $A\mathbf{T_A}B$, $A\mathbf{S_A}B$ for the algebraic product and sum, and so on.

Multisets

A multiset M of X is characterized by the count function $C_M\colon X \to \boldsymbol{N}$, where $\boldsymbol{N} = \{0,1,2,\dots\}$. Thus, $C_M(x)$ is the number of copies of the element $x \in X$.

The followings are basic relations and operations for crisp multisets;

A **(inclusion):**

$$M \subseteq N \Leftrightarrow C_M(x) \le C_N(x), \ \ \forall x \in X.$$

B **(equality):**

$$M = N \Leftrightarrow C_M(x) = C_N(x), \ \ \forall x \in X.$$

C **(union):**

$$C_{M \cup N}(x) = C_M(x) \vee C_N(x).$$

D **(intersection):**

$$C_{M \cap N}(x) = C_M(x) \wedge C_N(x).$$

E **(sum):**

$$C_{M+N}(x) = C_M(x) + C_N(x).$$

It is reasonable to assume that the number $C_M(\cdot)$ should be finite.

Real-valued multisets

The generalization to real-valued multisets is straightforward. It is sufficient to consider real-valued count functions. Namely, a real-valued multiset Q herein is characterized by the nonnegative real-valued count function ($C_Q(\cdot)\colon X \to \boldsymbol{R}^+ \cup \{0\}$). The above operations A–E of the count function is used for real-valued multisets.

Fuzzy multisets

Fuzzy multiset A of X (more often called fuzzy bag) is characterized by the function $C_A(\cdot)$ of the same symbol, but the value $C_A(x)$ is a finite set in I [15]. In other words, given $x \in X$,

$$C_A(x) = \{\mu, \mu', \dots, \mu''\}, \qquad \mu, \mu', \dots, \mu'' \in I.$$

For two fuzzy multisets A and B of X such that

$$C_A(x) = \{\mu, \mu', \dots, \mu''\}, \quad C_B(x) = \{\nu, \nu', \dots, \nu'''\},$$

the sum $A + B$ is given by

$$C_{A+B}(x) = \{\mu, \mu', \dots, \mu'', \nu, \nu', \dots, \nu'''\},$$

but other operations need another representation called membership sequence [8], [6].

A membership sequence is defined for each $C_A(x) = \{\mu, \mu', \dots, \mu''\}$; the set $\{\mu, \mu', \dots, \mu''\}$ is arranged into the decreasing order denoted by $\mu_A^1(x), \mu_A^2(x), \dots, \mu_A^m(x)$; namely,

$$\{\mu_A^1(x), \mu_A^2(x), \dots, \mu_A^m(x)\} = \{\mu, \mu', \dots, \mu''\}$$

and

$$\mu_A^1(x) \ge \mu_A^2(x) \ge \cdots \ge \mu_A^m(x).$$

The followings are other basic relations and operations for fuzzy multisets [8]; they are given in terms of the membership sequences.

1. **inclusion:**

$$A \subseteq B \Leftrightarrow \mu_A^j(x) \le \mu_B^j(x), \ j = 1, \dots, m, \ \forall x \in X.$$

2. **equality:**

$$A = B \Leftrightarrow \mu_A^j(x) = \mu_B^j(x), \ j = 1, \dots, m, \ \forall x \in X.$$

3. **union:**

$$\mu_{A \cup B}^j(x) = \mu_A^j(x) \vee \mu_B^j(x), \ j = 1, \dots, m, \ \forall x \in X.$$

4. **intersection:**

$$\mu_{A \cap B}^j(x) = \mu_A^j(x) \wedge \mu_B^j(x), \ j = 1, \dots, m, \ \forall x \in X.$$

5. **α-cuts:**
The weak and strong α-cuts for a fuzzy multiset A, denoted by $[A]_\alpha$ and $]A[_\alpha$, are defined as follows.

$$\mu_A^1(x) < \alpha \Rightarrow Count_{[A]_\alpha}(x) = 0,$$
$$\mu_A^j(x) \ge \alpha, \ \mu_A^{j+1}(x) < \alpha \Rightarrow Count_{[A]_\alpha}(x) = j,$$
$$j = 1, \dots, m.$$
$$\mu_A^1(x) \le \alpha \Rightarrow Count_{]A[_\alpha}(x) = 0,$$
$$\mu_A^j(x) > \alpha, \ \mu_A^{j+1}(x) \le \alpha \Rightarrow Count_{]A[_\alpha}(x) = j,$$
$$j = 1, \dots, m.$$

6. **t-norm and conorm:**

$$\mu^j_{A\mathbf{T}B}(x) = t(\mu^j_A(x), \mu^j_B(x)),$$
$$j = 1,\ldots,m,\ \forall x \in X.$$
$$\mu^j_{A\mathbf{S}B}(x) = s(\mu^j_A(x), \mu^j_B(x)),$$
$$j = 1,\ldots,m,\ \forall x \in X.$$

7. **average:** A variety of averaging operators for two fuzzy sets have been studied [17]. For two numbers $a, b \in I$, denote an averaging operator by $a * b$. From the definition $a \wedge b \leq a * b \leq a \vee b$. We define

$$\mu^j_{A*B}(x) = \mu^j_A(x) * \mu^j_B(x),\ j = 1,\ldots,m,\ \forall x \in X.$$

III. INFINITE MEMBERSHIPS

Even when crisp multisets cannot admit infinite values of the function $C_M(x)$, fuzzy multisets are capable of having infinite number of memberships. Remark that every infinite set does not provide a well-defined fuzzy multiset, since an α-cut of a fuzzy multiset should give a crisp multiset of the finite count.

Fuzzy multisets of infinite membership sequences

Instead of the finite set, infinite $C_A(x) = \{\mu, \mu', \ldots\}$ is used. We assume that the members $\{\mu, \mu', \ldots\}$ of $C_A(x)$ can be arranged into the decreasing order:

$$C_A(x) = \{\mu^1_A(x), \mu^2_A(x), \ldots\}, \quad \mu^1_A(x) \geq \mu^2_A(x) \geq \cdots$$

In order that the α-cuts provide well-defined crisp multisets, it is necessary and sufficient that

$$\mu^j_A(x) \to 0, \quad \text{as } j \to \infty,$$

for all $x \in X$. This class of fuzzy multiset of X is denoted by $\mathcal{FM}_0(X)$.

The operations such as $A+B$, $A \cup B$, *etc.* are defined in the same way as above except that $m \to \infty$ in the definitions. We have

Proposition 1. For $A, B \in \mathcal{FM}_0(X)$,

1. $A + B \in \mathcal{FM}_0(X)$,
2. $A \cup B \in \mathcal{FM}_0(X)$,
3. $A \cap B \in \mathcal{FM}_0(X)$,
4. $A\mathbf{T}B \in \mathcal{FM}_0(X)$;
5. $A\mathbf{S}_\mathbf{S}B \in \mathcal{FM}_0(X)$,
6. $A\mathbf{S}_\mathbf{A}B \in \mathcal{FM}_0(X)$,
7. $A\mathbf{S}_\mathbf{B}B \in \mathcal{FM}_0(X)$,
8. $A\mathbf{S}_\mathbf{F}B \in \mathcal{FM}_0(X)$,
9. $A\mathbf{S}_\mathbf{H}B \in \mathcal{FM}_0(X)$,
10. $A\mathbf{S}_\mathbf{Y}B \in \mathcal{FM}_0(X)$,
11. $A\mathbf{S}_\mathbf{SG}B \in \mathcal{FM}_0(X)$,
12. $A\mathbf{S}_\mathbf{D}B \in \mathcal{FM}_0(X)$ does not necessarily hold.
13. $A * B \in \mathcal{FM}_0(X)$

Fuzzy multisets of finite cardinality

A basic measure of a fuzzy set F is its cardinality defined by

$$|F| = \sum_{x \in X} \mu_F(x).$$

When a fuzzy multiset A of finite membership sets is considered, its generalization is immediate:

$$|A| = \sum_{x \in X} \sum_{j=1}^{m} \mu^j_A(x).$$

Let us proceed to consider the cardinality for the infinite memberships. We first define

$$|A|_x = \sum_{j=1}^{\infty} \mu^j_A(x). \tag{1}$$

Then,

$$|A| = \sum_{x \in X} |A|_x.$$

It is easy to see that $|A|$ is finite if and only if $|A|_x$ is finite for all $x \in X$, since we are considering finite X.

Note that for some sets, say $B \in \mathcal{FM}_0(X)$, $|B|_x = +\infty$. (Consider $\mu^j_B(x) = 1/j$.) We hence introduce a subclass $\mathcal{FM}_1(X)$ for which the cardinality is finite:

$$\mathcal{FM}_1(X) = \{A \in \mathcal{FM}_0(X) : |A|_x < \infty,\ \forall x \in X\}. \tag{2}$$

We now show the following propositions.

Proposition 2. For arbitrary $A, B \in \mathcal{FM}_1(X)$,

(I) $A + B \in \mathcal{FM}_1(X)$,
(II) $A \cup B \in \mathcal{FM}_1(X)$,
(III) $A \cap B \in \mathcal{FM}_1(X)$.
(IV) $A\mathbf{T}B \in \mathcal{FM}_1(X)$.
(V) $A * B \in \mathcal{FM}_1(X)$.

Proposition 3. For arbitrary $A, B \in \mathcal{FM}_1(X)$,

(i) $A\mathbf{S}_\mathbf{S}B \in \mathcal{FM}_1(X)$,
(ii) $A\mathbf{S}_\mathbf{A}B \in \mathcal{FM}_1(X)$,
(iii) $A\mathbf{S}_\mathbf{B}B \in \mathcal{FM}_1(X)$,
(iv) $A\mathbf{S}_\mathbf{F}B \in \mathcal{FM}_1(X)$,
(v) $A\mathbf{S}_\mathbf{H}B \in \mathcal{FM}_1(X)$,
(vi) $A\mathbf{S}_\mathbf{Y}B \in \mathcal{FM}_1(X)$,
(vii) $A\mathbf{S}_\mathbf{SG}B \in \mathcal{FM}_1(X)$,
(viii) $A\mathbf{S}_\mathbf{D}B$ is not necessarily in $\mathcal{FM}_1(X)$.

Proofs of Propositions 1–3 are also omitted here.

IV. GENERALIZATION OF MEMBERSHIP SEQUENCE

The fuzzy multisets do not include the real-valued multisets. We consider the smallest generalization of multisets that include both fuzzy multisets and real-valued multisets.

Set-valued multisets

Let us notice that the membership sequence, whether it is finite or infinite, is regarded as a nonincreasing step function. In view of this, we first consider a monotone nonincreasing function $\zeta_A(y;x)$ of the variable $y \in [0,+\infty)$ with the values in $[0,+\infty)$ for every $x \in X$ as a parameter. Moreover the function is assumed to satisfy $\zeta_A(y;x) \to 0$ as $y \to \infty$. Even if we do not assume any kind of continuity, it is well-known that the function $\zeta_A(y;x)$ is continuous almost everywhere due to the monotone property. We however assume, for the next step, that the function is upper-semicontinous.

Second, this function $\zeta_A(y;x)$ is transformed to a closed set $\nu_A(y,z;x)$ on the (y,z)-plane; we use the set $\nu_A(\cdot,\cdot;x)$ as the membership for the generalized fuzzy multiset. This set is defined by

$$\nu_A(y,z;x) = \{(y,z) \in [0,\infty)^2 : \zeta_A(y;x) \geq z\}.$$

Another function $\eta_A(z;x)$ with the variable z derived from ν_A is moreover defined:

$$\eta_A(z;x) = \sup\{y \in \nu_A(y,z;x)\}, \quad (z \in (0,\infty)).$$

It is evident that if we define

$$\nu'_A(z,y;x) = \{(y,z) \in [0,\infty) \times (0,\infty) : \eta_A(z;x) \geq y\} \cup \{(y,0) : y \in [0,\infty)\}, \tag{3}$$

then

$$\nu_A(y,z;x) = \nu'_A(z,y;x).$$

The generalized fuzzy multiset A is characterized by $\nu_A(y,z;x)$ derived from $\zeta_A(y;x)$.

For two generalized fuzzy multisets A and B of X, the basic relations and operations are defined by the operations on the sets ν_A and ν_B.

(I) (inclusion)
$A \subseteq B \Leftrightarrow \nu_A(\cdot,\cdot;x) \subseteq \nu_B(\cdot,\cdot;x), \ \forall x \in X.$

(II) (equality)
$A = B \Leftrightarrow \nu_A(\cdot,\cdot;x) = \nu_B(\cdot,\cdot;x), \ \forall x \in X.$

(III) (sum) Define

$$\eta_{A+B}(z;x) = \eta_A(z;x) + \eta_B(z;x)$$

and derive ν'_{A+B} from $\eta_{A+B}(z;x)$ using (3). Define ν_{A+B} by

$$\nu_{A+B}(y,z;x) = \nu'_{A+B}(z,y;x).$$

(IV) (union) Define

$$\zeta_{A\cup B}(y;x) = \zeta_A(y;x) \vee \zeta_B(y;x)$$

and derive $\nu_{A\cup B}$ from $\zeta_{A\cup B}(y;x)$.

(V) (intersection) Define

$$\zeta_{A\cap B}(y;x) = \zeta_A(y;x) \wedge \zeta_B(y;x)$$

and derive $\nu_{A\cap B}$ from $\zeta_{A\cap B}(y;x)$.

(VI) (t-norm and conorm) Define

$$\zeta_{A\mathbf{T}B}(y;x) = t(\zeta_A(y;x), \zeta_B(y;x)),$$
$$\zeta_{A\mathbf{S}B}(y;x) = s(\zeta_A(y;x), \zeta_B(y;x))$$

and derive $\nu_{A\mathbf{T}B}$ and $\nu_{A\mathbf{S}B}$ from $\zeta_{A\mathbf{T}B}(y;x)$ and $\zeta_{A\mathbf{S}B}(y;x)$, respectively.

Although it is evident that this generalization includes the fuzzy multisets and the positive real-valued multisets, it is not obvious that this also includes a fuzzification by the fuzzy number [6].

Multisets with values of fuzzy numbers

In this fuzzification, the count function $C_M(x)$ takes the value of fuzzy numbers on $[0,\infty)$. The operations are defined by the same manner as those for the crisp multisets except that the values are fuzzy numbers in the definitions below:

1. (inclusion):
$M \subseteq N \Leftrightarrow C_M(x) \leq C_N(x), \ \forall x \in X.$
2. (equality):
$M = N \Leftrightarrow C_M(x) = C_N(x), \ \forall x \in X.$
3. (union):
$C_{M\cup N}(x) = C_M(x) \vee C_N(x).$
4. (intersection):
$C_{M\cap N}(x) = C_M(x) \wedge C_N(x).$
5. (sum):
$C_{M+N}(x) = C_M(x) + C_N(x).$

A simple mapping from the class of $C_A(x)$ to $\zeta_A(\cdot;x)$ is used for showing this generalization encompasses the latter fuzzification. Notice that $C_A(x)$, a fuzzy number, consists of two upper-semicontinuous functions $L(y)$ and $R(y)$:

$$C_A(x) = \begin{cases} L(y), & 0 \leq y \leq c, \\ R(y), & c \leq y, \end{cases}$$

where $L(c) = R(c) = 1$.

First, $L(y)$ is transformed into a lower-semicontinuous function $\tilde{L}(y)$ which is equal to $L(y)$ on all continuity points. Then $C_A(x)$ is mapped to $\zeta_A(\cdot;x)$ by the next rule:

$$\zeta_A(y;x) = \begin{cases} 1 - \dfrac{1}{2}\tilde{L}(y), & 0 \leq y \leq c, \\[2ex] \dfrac{1}{2}R(y), & c < y. \end{cases}$$

It is immediate to see that the inclusion and equality as well as the operations of the sum, union, and intersection for the fuzzification by the fuzzy number is expressed in terms of the present generalization by the above mapping.

V. CONCLUSION

We have discussed generalizations which include infinite features in multisets. In the first generalization fuzzy multisets of infinite memberships and finite cardinality has been introduced and it has been shown that the standard set operations are performed within this class, whereas an exceptional t-conorm of the drastic sum may put the derived set out of this class. More general results will be expected about t-norms and conorms having the former property.

In the second generalization real-valued multisets and fuzzy multisets are considered in the unified framework.

Multisets have close relationships with rough sets and their generalizations [16]. Theoretical aspects of multisets in relation to rough sets should further be considered.

We have suggested application of infinite fuzzy multisets to information retrieval on WWW. More efforts should be concentrated on such applications as future studies.

ACKNOWLEDGMENT

This research has partially been supported by the Grant-in-Aid for Scientific Research (C), Japan Society for the Promotion of Science, No.13680475.

REFERENCES

[1] W.Blizard, Real-valued multisets and fuzzy sets, *Fuzzy Sets and Systems*, Vol. 33, pp. 77–97, 1989.

[2] J.Celko. *Joe Celko's SQL for Smarties: Advanced SQL Programming*, Morgan Kaufmann, 1995.

[3] A.di Nola, S.Sessa, W.Pedrycz, E.Sanchez, *Fuzzy Relation Equations and Their Applications to Knowledge Engineering*, Kluwer, Dordrecht, 1989.

[4] B.Li, W.Peizhang, L.Xihui, Fuzzy bags with set-valued statistics, *Comput. Math. Applic.*, Vol. 15, pp. 811–818, 1988.

[5] D.E.Knuth, *The Art of Computer Programming, Vol. 2*, Addison-Wesley, 1969.

[6] Z.-Q. Liu, S.Miyamoto (Eds.), *Soft Computing and Human-Centered Machines*, Springer, Tokyo, 2000.

[7] Z. Manna, R. Waldinger, *The Logical Basis for Computer Programming, Vol. 1: Deductive Reasoning*, Addison-Wesley, 1985.

[8] S.Miyamoto, Basic operations of fuzzy multisets, *J. of Japan Society for Fuzzy Theory and Systems*, Vol. 8, No. 4, pp. 639–645, 1996 (in Japanese).

[9] S.Miyamoto, K.S.Kim, Multiset-valued images of fuzzy sets, *Proceedings of the Third Asian Fuzzy Systems Symposium*, June 18-21, 1998, Masan, Korea, pp.543-548.

[10] S.Miyamoto, Rough sets and multisets in a model of information retrieval, in F.Crestani *et al.* eds., *Soft Computing in Information Retrieval: Techniques and Applications*, Springer, pp. 373–393, 2000.

[11] Z.Pawlak, Hard and soft sets, In: W.P.Zialko (Ed.), *Rough Sets, Fuzzy Sets, and Knowledge Discovery*, Springer-Verlag, London, 1994, pp. 130–135.

[12] A.Ramer, C.-C. Wang, Fuzzy multisets, *Proc. of 1996 Asian Fuzzy Systems Symposium*, Dec. 11-14, 1996, Kenting, Taiwan, pp. 429–434.

[13] A.Rebai, J.-M. Martel, A fuzzy bag approach to choosing the "best" multiattributed potential actions in a multiple judgement and non cardinal data context, *Fuzzy Sets and Systems*, Vol. 87, pp. 159–166, 1997.

[14] B.Schweizer, A.Sklar, *Probabilistic Metric Spaces*, North-Holland, New York, 1983.

[15] R.R.Yager, On the theory of bags, *Int. J. General Systems*, Vol. 13, pp. 23–37, 1986.

[16] Y.Y.Yao, S.K.M.Wong, T.Y.Lin, A review of rough set models, in T.Y.Lin, N.Cercone, eds., *Rough Sets and Data Mining: Analysis of Imprecise Data*, Kluwer, Boston, pp. 47–75, 1997.

[17] H.-J.Zimmermann, *Fuzzy Set Thoery and Its Applications, 3rd Ed.*, Kluwer, Boston, 1996.

Products of Random Matrices
in
Control and Signal Processing

Jelel EZZINE.

ACS, ENIT, Campus Universitaire, BP 37 Le Belvédère, Tunis, TUNISIA
e-mail: jelel.ezzine@ieee.org

Abstract

The paper gives a review of the products of random matrices and their applications in different areas of research. It dwells mainly on two applications in control and signal processing. The one in control deals with the almost sure stabilization of jump parameter systems via the assignment of the leading Lyapunov exponent. The second one deals with a quite interesting application in adaptive filtering. The particularity of this problem is the finiteness of the alphabet dealt with in the filtering problem. Though this could be seen as a special case, it happens to be of great interest in digital communication. Indeed such a model is far more appropriate than classical ones. It is shown that due to the finiteness of the alphabet, that rigorous stability results can be achieved without any unrealistic independence assumptions.

Keywords: Products of random matrices, Almost-sure stability/stabilization, Lyapunov exponents, Linear matrix inequalities (LMI's), Adaptive filtering.

Notation

$(\cdot)^\dagger$	pseudo inverse of $(\cdot)$
$\|\cdot\|$	Euclidean norm of a vector, or induced Euclidean norm of a matrix
$\mathcal{A}$	set of matrices A_i, $i = 1, \cdots, N$
$A_i^{[l]}$	product of l matrices $A_i \in \mathcal{A}$, referred to as l-block
$\mathcal{A}^{[l]}$	set of matrices $A_i^{[l]}$, $i = 1, \cdots, N^l$
$\mathbf{E}$	expected value operator
λ_i	ith Lyapunov exponent
λ_1	leading or largest Lyapunov exponent
$\mathcal{N}$	$\{i \mid i = 1, \cdots, N\}$
$\mathcal{N}^{[l]}$	$\{i \mid i = 1, \cdots, N^l\}$
p_i	steady state probability of the matrix A_i
$p_i^{[l]}$	steady state probability of the l-block matrix $A_i^{[l]}$

Acronym	**Meaning**
a.s.	almost-surely (or with probability one)
DTR	discrete-time random dynamical system
FSMC	finite states Markov chain
FKT	Furstenberg-Kesten theorem
GEVP	Generalized Eigenvalues Problem
LMI	Linear Matrix Inequality
NSGASS	non-switching gains a.s. stabilizable
SGASS	switching gains a.s. stabilizable
SGLASS	switching gains l-block a.s. stabilizable
SGSASS	switching gains strongly a.s. stabilizable
SGSS	switching gains strongly stabilizable

1 Introduction

The product of random matrices has attracted the attention of mathematicians [1], physicists [2]and engineers in recent years[3][4][5][6][7][8]. One of the early results was proposed by Furstenberg and Kesten in 1960 [9]. In their contribution they gave the first quite general result on the almost sure stability of the products of random matrices. Their results is based on the Lyapunov exponent concept [10]. The most general result to date, which made the one century old ideas of Lyapunov applicable, is the work of Osceledest that appeared in 1968 [11]. It goes without saying that the products of random matrices found applications beyond those cited above. For instance the products of random matrices appears in disordered dynamical system[12], Ising models with quenched disorder, wave propagation through random medium, particle floating on turbulent fluid surface, discrete-time Schrödinger equations describing a spectrum of physical phenomena evolving in random potentials [13], demographic [1] and econometric [1] applications just to cite a few.

From an engineering application view, products of random matrices can be found on a large spectrum of situations. Systems of this type can be used to model networks with periodically varying switches, synchronously switched linear systems, randomly sampled-data systems, systems subject to failures, manufacturing systems, macroeconomics models, and last but not least to approximate nonlinear stochastic systems. References to these applications can be found in [3] and [5] and references therein.

2 The General Setup

In this section we introduce the general setup that all discussed problems will pertain to. Moreover, a key theorem is stated to facilitate the introduction of the results discussed in this paper as well as to make the presentation as self contained as possible. The second section discusses the new and important issues of computational complexity related to the products of random matrices.

2.1 Discrete-Time Random Dynamical Systems

In this section, attention is restricted to linear discrete-time systems with randomly varying parameters. This class of systems will be called linear discrete-time random (DTR) dynamical systems in contrast to linear discrete-time stochastic systems where randomness enters the system as an input and/or measurement noise. N-form hybrid systems are a special class of DTR dynamical systems.

More precisely, this section deals with the following problem : Let $\{A_i, i \in \mathbf{N}\}$ be a sequence of random, $n \times n$ matrices. To each $x_0 \in \mathbf{R}^n$ associate the process $\{X_k, k \in \mathbf{N}\}$ with values in $\mathbf{R}^n$, which is the solution to

$$X_{k+1} = A_k X_k, \quad k \in \mathbf{N}, \quad X_0 = x_0. \tag{1}$$

We have $X_{k+1} = A_k \cdots A_1 x_0$. The asymptotic behavior of this process is addressed by the following theorem.

Furstenberg-Kesten Theorem (FKT) *[9] : Let $\{A_i,\ i \in \mathbf{N}\}$ be a stationary, metrically transitive stochastic process with values in the set of $n\times n$ matrices such that $\mathbf{E}\{\log^+ \|A_0\|\} < \infty$. Then, with probability one*

$$\begin{aligned} \lim_{k\to\infty} \frac{1}{k}\mathbf{E}\{\log \|S_k\|\} &= \inf_k \frac{1}{k}\mathbf{E}\{\log \|S_k\|\} \qquad (2) \\ &= \lim_{k\to\infty} \frac{1}{k} \log \|S_k\| := \lambda_1, \end{aligned}$$

where $S_k := A_k \cdots A_1$ and $\lambda_1 \in \mathbf{R} \cup \{-\infty\}$, is the largest Lyapunov exponent of the process.

The FKT is a generalization of Birkoff's ergodic theorem to the case of matrix valued functions. It is also a simple version of the Osceledest theorem [30]. Though it provides a necessary and sufficient a.s. stability criterion, it is highly impractical from a design point of view. Therefore, one way to avoid this difficulty is to use the FKT to derive simpler criteria to test for the a.s. stability of the system, as well as aid in devising controllers.

2.2 Products of Random Matrices and Computational Complexity

Since the products of random matrices started getting more and more attention, and since their applications kept multiplying, some rigorous and definitive answers were called for as to the computational complexities related to the products of random matrices such as computing their Lyapunov exponents. It turned out, just lately, that such a task is highly complex and that it is highly unlikely that efficient algorithms will be found to tackle such problem and related ones. Indeed, the ability to compute arbitrarily accurate approximations of the Lyapunov exponent and related quantities does not rule out that the problem of deciding wether the underlying dynamical system is stable or not is undecidable. A plethora of related computational problems is reviewed in [14].

3 Leading Lyapunov Exponent Bounds

There are few results concerning the approximation of the leading Lyapunov exponent. Most of these approximations can be found mainly in the physics literature, for instance the reader can refer to [15], [16], [17], and references therein. An information theoretic based approximation of the Lyapunov spectrum can be found in [8]. There are also results dealing with the difficult and important problem of stopping rules. In [7] an asymptotically exact probabilistic stopping rule is proposed. It is important to mention that it is the first rigorous stopping rule, though expected to be quite conservative.

3.1 The Furstenberg-Kesten Bounds (FKB)

A series of increasingly tighter upper bounds on the largest Lyapunov exponent are provided along with a simple lower bound. These bounds provide a motivation for the main result of the paper which will be introduced later.

Theorem 1: *The largest Lyapunov exponent, λ_1, of a homogeneous N-form DTR dynamical system, with a stationary irreducible FSMC satisfies the following inequality*

$$\begin{aligned} \tfrac{1}{n}\textstyle\sum_{i\in\mathcal{N}} p_i \log|det(A_i)| &\le \lambda_1 \\ \lambda_1 &\le \textstyle\sum_{i\in\mathcal{N}^{[l]}} p_i^{[l]} \log\|TA_i^{[l]}T^{-1}\| \\ &:= \lambda^{[l]}, \end{aligned} \tag{3}$$

where T is any similarity transformation.

Moreover,

$$\lambda^{[ql]} \le q\lambda^{[l]}, \quad q, l \in \mathbf{N} \setminus \{0\}. \tag{4}$$

It is important to note that the lower bound holds for the Lyapunov spectrum and not only for λ_1. That is, the lower bound is a lower bound for the smallest Lyapunov exponent as well. Consequently, the lower and upper bounds estimate the spread of the Lyapunov spectrum, thus providing valuable information regarding the slowest as well as fastest dynamics.

Though the first part of the proposition asserts that the proposed sequences of upper bounds eventually converge to the largest Lyapunov exponent, they shed no light on the *nature* of this convergence! For instance, it would be of practical interest to identify, at least, one *monotonically* convergent subsequence. This subsequence would allow one, to systematically pick increasingly tighter bounds. For example, the following is such subsequence: $\lambda^{[q^{i+1}l]} \leq \lambda^{[q^{i}l]}$, $i \in \mathbf{N}$. Thus, it is possible to pick as large l as one needs, to achieve a desired accuracy while computing λ_1.

The following corollaries provide useful results for stabilization purposes. The first supplies a sufficient robust a.s. stability criterion for parametrically perturbed hybrid systems. The second furnishes simple sufficient a.s. stability criteria, as well as a necessary a.s. stability test.

Corollary 1: *The N-form hybrid system with a stationary irreducible FSMC is a.s. stable if*

$$\sum_{i \in \mathcal{N}} p_i \|TA_iT^{-1}\| < 1, \tag{5}$$

where T is a similarity transformation, and is a.s. stable unless

$$\prod_{i \in \mathcal{N}} |det(A_i)|^{p_i} < 1. \tag{6}$$

3.2 Simultaneous Norm Minimization by Scaling

Obviously, for an arbitrary T, the above upper bounds are in general quite conservative. In order to alleviate this shortcoming, we use the degree of freedom provided by T, to get tighter bounds. Therefore, consider a single similarity transformation $z = Tx$, which is applied to all systems. The new systems will be

$$z(n+1) = TA_iT^{-1}z(n). \tag{7}$$

We than have

$$e^{\lambda_1} \leq \sum_{i \in \mathcal{N}} p_i \|TA_iT^{-1}\|. \tag{8}$$

The next step is to minimize the latter upper bound over nonsingular matrices T. In the sequel an LMI based solution of the latter optimization problem is proposed.

Corollary 2: *The λ_1 of a homogeneous N-form hybrid system with a stationary irreducible FSMC satisfies the following inequality*

$$e^{\lambda_1} \leq \sum_{i \in \mathcal{N}} p_i \alpha_i, \tag{9}$$

where the optimal α_i's are given by the following linear optimization problem subject to LMI constraints

$$\begin{array}{l} minimize \sum_{i \in \mathcal{N}} p_i \alpha_i \\ subject\ to\ \ P = P^T > 0,\ \ A_i^T P A_i < \alpha_i^2 P, \\ i = 1, \cdots, N. \end{array} \tag{10}$$

where the p_i's are the steady state probabilities of the FSMC.

Unfortunately, the above mathematical program is not an LMI. Nevertheless, it can still be solved via a series of LMI's, where one would apply a bisection method in the n-dimensional space of the α_i's. While this approach will yield the optimal solution, it is computationally demanding. Thus, as a compromise between optimality and computational burden, a suboptimal solution, based on a series of LMI's is proposed in the next section.

This is a useful result, for it solves the robust stability problem for the linear difference inclusion (LDI) defined by the N forms of the hybrid system.

3.3 An Approximate Solution

In this section we propose an approximate solution to the optimization problem defined in corollary 4.3. The problem is broken in a series of LMI's which are solved iteratively.

Scaling Algorithm (SA):

1. $k = 0$; set $P = I$
2. solve the following LMI's for fixed P

$$\begin{array}{ll} minimize & \sum_{i \in \mathcal{N}} p_i \alpha_i \\ subject\ to & A_i^T P A_i \leq \alpha_i^2 P, \quad i = 1, \cdots, N. \end{array} \tag{11}$$

3. solve the following GEVP's using the α_i's of the previous step

$$\begin{array}{ll} minimize & t \\ subject\ to & A_i^T P A_i \leq t\alpha_i^2 P, \quad i = 1, \cdots, N. \end{array} \tag{12}$$

4. go to the second step, or stop if there are no further improvements.

4 Two Novel Applications in Control and Signal Processing

The two applications that are presented in this section, draw on two research activities pursued by the author and his colleagues. The first one deals with the almost sure stabilization of Linear Jump Parameter Systems (LJPS). The problem, since related to the products of random matrices, remains unsolved and some approximate means are introduced. The second tackles the problem of adaptive filtering in the highly practical case of finite alphabet. Though this is a special case it models digital communication systems. The breakthrough in respect to adaptive filtering is the complete elimination of the wrong independence assumption that plagues this important DSP area.

4.1 Stabilization of Jump Linear Systems

The discrete-time hybrid systems considered in this paper are assumed to have the form

$$\left\{ \begin{array}{lcl} x(k+1) & = & A(r(k))x(k) + B(r(k))u(k) \\ y(k) & = & C(r(k))x(k) \end{array} \right. \tag{13}$$

where x is the system state vector of dimension n, u is the control input vector of dimension p, y is the output vector of dimension m, and $r(k)$ is the *form index* which is either a deterministic or a stochastic scalar sequence taking values in the finite index set $\mathcal{N} = \{1, 2, ..., N\}$. The system takes the realization $\sum_i = (A_i, B_i, C_i)$ when $r(k) = i$, with $i \in \mathcal{N}$. This realization is called the ith *form*. The primary emphasis of the work is the case where $r(k)$ is a stochastic process. In this case $r(k)$ is assumed to be a Finite State Markov Chain (FSMC) with transition probabilities

$$Prob.\{r(k+1) = j | r(k) = i\} = p_{ij}. \tag{14}$$

It is assumed, unless stated otherwise, that the FSMC is stationary and irreducible.

All available stabilization algorithms, for hybrid systems, can be mostly classified into two classes. While the first class considers perfectly observed form and state processes, the second considers unobserved or unknown ones. These stabilization algorithms achieve mean-square stability, to the exception of those reported in where a.s. stabilization is achieved directly through the negativity of the Lyapunov spectrum.

4.1.1 On Switching Gains Stabilization

The stabilization of hybrid systems was almost always carried via optimal control tools. While this route yields satisfactory results, it fails short from shedding light on the mechanism(s) taking place while achieving stabilization. Moreover, it is a mean-square stabilization approach. In this section we will formulate an a.s. stabilization problem, based on the above proposed bounds.

We are concerned with the following problem. Given N desired eigenstructures, or desired Lyapunov spectrum, specified through desired matrices A_{d_i}, $i = 1, \cdots, N$, find K_i, $i = 1, \cdots, N$, and a similarity transformation T such that

$$\|A_{d_i} - (A_i - B_i K_i)\|, \quad i = 1, \cdots, N, \tag{15}$$

are simultaneously minimized, and the left hand side of the following inequality

$$\sum_{i \in \mathcal{N}} p_i \|T(A_i - B_i K_i)T^{-1}\| < 1, \tag{16}$$

is smallest.

Should the above goals be achieved, the closed-loop system is made a.s. exponentially stable via the switching gains K_i. Replacing the K_i's by K, will result in a constant or a non-switching gain controller.

4.1.2 Almost-Sure Stabilizability

Whereas a.s. stability is a more practical notion than mean-square stability, and since the latter has long appeared in the literature, the former one is defined here. The natural way to define a.s. stabilizability is through the negativity of the leading Lyapunov exponent, in complete analogy with eigenvalues negativity.

Definition 1: *A hybrid system (1, 2), with an irreducible FSMC, is said to be switching gains almost-surely stabilizable (SGASS), via static state feedback, if there exist K_i, $i = 1, \cdots, N$, such that $\lambda_1 < 0$. It is said non-switching gains almost-surely stabilizable (NSGASS) if $K_i = K, i = 1, \cdots, N$.*

Theorem 2: *A hybrid system (1, 2), with an irreducible FSMC, is SGASS, if there exist $l > 0$, a similarity transformation T, and K_i, $i = 1, \cdots, N$, such that*

$$\sum_{i \in \mathcal{N}^{[l]}} p_i^{[l]} \log \|T(A_i - B_i K_i)^{[l]} T^{-1}\| \leq 0. \tag{17}$$

It is NSGASS if the above inequality holds for $K_i = K$, $i = 1, \cdots, N$.

Motivated by the above result, the following modified stabilization definition is introduced.

Definition 2: *A hybrid system (1, 2), with an irreducible FSMC, is said to be switching gains l-block a.s. stabilizable (SGLASS), via static state feedback, if there exist a similarity transformation T, and gains K_i, $i = 1, \cdots, N$, such that*

$$\sum_{i \in \mathcal{N}^{[l]}} p_i^{[l]} \log \|T(A_i - B_i K_i)^{[l]} T^{-1}\| \leq 0, \tag{18}$$

and switching gains strongly a.s. stabilizable (SGSASS), or switching gains strongly stabilizable (SGSS), if it is 1-block switching gains almost-surely stabilizable.

In order to check if the system is SGSASS, and compute the N gains that will achieve a.s. stabilization, via switching gains, along with the norm scaling T matrix, in case it is SGSASS, one simply applies the following result.

Theorem 3: *A hybrid system (1, 2), with an irreducible FSMC, is (SGSS), provided that the matrices $P := T^T T > 0$, and K_i, $i = 1, \cdots, N$, generated by the following mathematical program*

$$\begin{aligned} &\textit{minimize} \quad \textstyle\sum_{i \in \mathcal{N}} p_i \alpha_i \\ &\textit{subject to} \quad P = P^T > 0, \\ &\qquad\qquad (A_i - B_i K_i)^T P (A_i - B_i K_i) < \alpha_i^2 P, \\ &\qquad\qquad i = 1, \cdots, N. \end{aligned} \tag{19}$$

satisfy

$$\sum_{i \in \mathcal{N}} p_i \log \|T(A_i - B_i K_i)T^{-1}\| \leq 0. \tag{20}$$

It is NSGSS if the above inequality holds for $K_i = K$, $i = 1, \cdots, N$.

An approximate solution to the nonlinear optimization problem, via LMIs [18], can be obtained as follows.

Test Algorithm (TA):

1. $k = 0$; set $P = I$

2. solve the following LMI's for fixed P

$$\begin{array}{l} \mathit{minimize} \quad \alpha_i \\ \quad K_i,\ \alpha_i \\ \mathit{subject\ to} \\ \begin{bmatrix} -\alpha_i P & (A_i - B_i K_i) \\ (A_i - B_i K_i)^T & -\alpha_i P^{-1} \end{bmatrix} \leq 0, \\ i = 1, \cdots, N. \end{array} \tag{21}$$

3. solve the following GEVP's using the K_i's and α_i's of the previous step using the SA Algorithm.

$$\begin{array}{l} \mathit{minimize} \quad t \\ \mathit{subject\ to} \quad (A_i - B_i K_i)^T P (A_i - B_i K_i) \leq t\alpha_i^2 P, \\ i = 1, \cdots, N. \end{array} \tag{22}$$

4. go to the second step, or stop if there are no further improvements.

The stabilization algorithm proposed in the sequel provides either switching or non-witching gains controllers, as the difference to achieve one or the other is straight forward.

4.1.3 Stabilization Algorithms

For controller design, it is customary to have desired design specifications. For instance eigenstructure assignment is a classical design specification. The analog of eigenstructure assignment, or at least eigenvalues assignment, is the Lyapunov spectrum assignment. However, this design concept, for jump parameters systems, or any other class of systems besides LTI and periodic ones, remains a challenging open problem. One way to alleviate this lack of a design procedure, to assign the Lyapunov spectrum, is to impose "similar" specifications through other means. For instance, one can require that the closed loop matrices be as close as possible, in some sense, to N desired matrices A_{d_i}. This way the resulting gains, will try to achieve a.s. stability as well as making the resulting closed loop matrices close to the desired ones in the sense specified by the design algorithm. Should the gains succeed in exactly matching the desired matrices, the closed-loop hybrid system would inherit the desired hybrid system's dynamical characteristics such as its Lyapunov spectrum.

4.1.4 The LMI Approach to Almost-Sure Stabilization

Due to its mathematical tractability, the Frobenious norm was usually used as a measure of closeness between matrices. However, the Frobenious norm is not an operator norm, and more meaningful mean to define closeness is via the induced Euclidean norm, i.e.,

$$\|A_{d_i} - (A_i - B_i K_i)\|.$$

The optimal gain achieving the minimum of the above norm is treated in the following section as a GEVP [18]. As argued earlier, this LMI approach has the advantage of providing efficient numerical means to compute the necessary gains, as well as norm scaling transformations to improve the tightness of the sufficient stability test.

GEVP-Based Design In order to compute the N gains that will achieve a.s. stabilization, given desired closed-loop matrices A_{d_i}, along with the norm scaling matrix P, one solves separately, the two optimization problems given below.

Design Algorithm (DA):

1. solve the following using $P = I$,

$$\begin{array}{l} \mathit{minimize} \quad \alpha_i \\ \quad K_i \\ \mathit{subject\ to} \\ \begin{bmatrix} -\alpha_i I & (A_{d_i} - (A_i - B_i K_i)) \\ (A_{d_i} - (A_i - B_i K_i))^T & -\alpha_i I \end{bmatrix} \\ \leq 0, \ i = 1, \cdots, N. \end{array} \tag{23}$$

2. to compute the similarity transformation, solve the following with the K_i's found in the first step, using SA.

$$\begin{array}{l} \mathit{minimize} \quad \sum_{i \in \mathcal{N}} p_i \gamma_i \\ \mathit{subject\ to} \quad (A_i - B_i K_i)^T P (A_i - B_i K_i) < \gamma_i^2 P, \\ i = 1, \cdots, N. \end{array} \tag{24}$$

Note that the outcome of the first optimization step might not yield a negative upper bound and still achieve a.s. stability. However, computing tighter bounds, as done in the second optimization step, or better the resulting Lyapunov exponent is the only way to tell if the system has been stabilized or not. However, the computation of the Lyapunov exponent is computationally much more demanding.

Numerical applications can be found in the cited references.

4.2 Convergence of Adaptive Filters

4.2.1 Background

Due to its simplicity, the LMS algorithm (or its variants: normalized LMS and sign algorithms such as clipped algorithm, pilot algorithm and zero-forcing algorithm) is widely used for adaptive signal processing. In a general context, a rigorous convergence analysis is difficult and has been hard to find.

The LMS algorithm consists in calculating recursively the parameter vector H^{opt} that optimizes, in mean square sense, the linear estimation of a measured random signal a_k, from an observation vector X_k having the same dimension d as H. It is given by:

$$H_{k+1} = H_k + \mu X_k (a_k - X_k^T H_k) \tag{25}$$

where the step size μ is a positive constant and X^T denotes the transpose of the vector X. Equivalently (1) becomes:

$$V_{k+1} = (I - \mu X_k X_k^T) V_k + \mu X_k b_k \tag{26}$$

Using the filter parameter deviation vector from the optimal ones $V_k = H_k - H^{opt}$ and the optimal filter output noise b_k.

The dynamical behavior of the algorithm is related to the transition matrix U_{kq}:

$$U_{kq} = \prod_{i=k}^{q} (I - \mu X_i X_i^T). \tag{27}$$

To overcome the complexity of this analysis due to the random nature of U_{kq}, an average approach is usually considered. It deals with the behavior study of both $E(V_k)$ and $E(V_k V_k^T)$. Since, X_k and V_k are dependent, it is difficult to establish a relation between $E(V_{k+1}V_{k+1}^T)$ and $E(V_k V_k^T)$, in fact we have :

$$\begin{aligned} E(V_{k+1}V_{k+1}^T) &= E\left((I - \mu X_k X_k^T) V_k V_k^T (I - \mu X_k X_k^T)^T\right) \\ &+ \mu^2 E(b_k^2) E(X_k X_k^T). \end{aligned} \tag{28}$$

where b_k is a second-order zero mean sequence, independent of X_k.
Different approaches are proposed to study the convergence of the LMS in the quadratic mean sense and almost sure sense. They can be classified in three categories:

- *C*1 Direct approaches: it consists in studying the behavior of the transition matrix U_{kq} .
- *C*2 Approximation approaches: it deals with stochastic approximation methods, such as ODE or perturbation expansion methods .
- *C*3 Simplified approaches: it deals with a particular model of the input sequence, e.g., ϕ-mixing model or one that decouples the distribution of X_k in a radial and angular distributions .

These approaches need generally the following assumptions:

- *H*0 Ergodicity of X_k,
- *H*1 Excitation condition of the algorithm: e.g., invertibility of the input covariance matrix $E(X_k X_k^T)$,
- *H*2 Boundedness of X_k or of its moments,
- *H*3 Independence: usually, the unrealistic but helpful assumption of the statistical independence of X_k and X_{k-1} is made . Sometimes, the more realistic M-independence assumption is used .

The assumptions *H*0 and *H*1 are fundamental in identification theory, and *H*2 is realistic in many applications. The independence assumption *H*3 is mainly a technicality which is required to simplify the convergence proofs. A rigorous analysis without this assumption has been difficult to find .

In this section, we focus our study on the convergence analysis in a general digital transmission context. In this context, all studies of the algorithm's convergence are carried out, using the previous convergence results (*C*1, *C*2 and *C*3 approaches), thus they have the same limitations. As a way to overcome these limitations, an original and rigorous approach [6], inspired from control tools[3], adapted to the finite alphabet case encountered in digital transmission contexts. Therefore, a tailored framework is made available to study applications such as speech coding, echo cancellation, channel equalization.

4.2.2 The Finite Alphabet Case

In digital transmission context, X_k remains in a finite alphabet set $A = \{W_1, W_2, ..., W_N\}$. For example in QAM modulation context with two states $\{\pm 1\}$, when the dimension of X is $d = 2$, the finite alphabet set is :

$$A \in \left\{ \begin{pmatrix} +1 \\ +1 \end{pmatrix}, \begin{pmatrix} +1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ +1 \end{pmatrix} \right\}.$$

So we suppose that $X_k = W_{\theta(k)}$, where $\{\theta(k) : k \in \mathbf{Z}^+\}$ is a discrete-time Markov chain with finite state space $\{1, 2, ...N\}$ and transition probability matrix $P = [p_{ij}]$.

To prove almost sure and quadratic mean convergences of the LMS and its variants, we consider the following assumptions:

4.2.3 Assumptions

- *A*0 $\{\theta(k) : k \in \mathbf{Z}^+\}$ is an aperiodic Markov chain.
- *A*1 The alphabet $\{W_1, W_2, ..., W_N\}$ generates the space $\Re^d$.

It is interesting to note, that trough this approach the needed assumptions are subset of the usual ones. The assumption *A*0 is similar to *H*0. It means that $\{X_k\}$ is an ergodic sequence, and *A*1 is an excitation condition similar to *H*1. The boundedness of X_k (*H*2-assumption) is a natural outcome in digital transmission contexts. Most importantly, in the finite alphabet case, no independence assumption such as the usual *H*3 is needed.

4.2.4 Almost sure and Quadratic convergences

To prove almost sure convergence of the LMS algorithm, it is possible to split V_k in two additive terms, called transient deviation V_k^t, and fluctuation deviation V_k^f. The transient deviation V_k^t is defined by:

$$\begin{aligned} V_{k+1}^t &= (I - \mu X_k X_k^T) V_k^t = \prod_{i=0}^k (I - \mu X_i X_i^T) V_0^t \\ &= \prod_{i=0}^k (I - \mu W_{\theta(i)} W_{\theta(i)}^T) V_0^t = \left(\prod_{i=0}^k A_{\theta(i)}\right) V_0^t. \end{aligned} \tag{29}$$

where $V_0^t = V_0$, and depends only on the initial conditions V_0, but not on the output noise b_k. When the algorithm works properly, V_k^t should forget the initial deviation value V_0 and decreases to zero almost surely.

Theorem 4: *If the assumptions A0 and A1 hold, then V_k^t converge to zero with probability 1 as well as in the quadratic mean sense.*

It is interesting to note that the proof necessitates only elementary algebra [6]. The condition on μ is quite simple and useful in digital transmission contexts. In the noiseless case ($b_k = 0$), V_k as V_k^t converges almost surely to zero. When b_k is almost surely bounded V_k is almost surely bounded too.

The quadratic mean convergence of the LMS in the finite alphabet case is carried through, without using the constraining independence assumption *H*3. As in the previous section, this analysis can be extended to the complex case as well as to clipped algorithms.

The convergence analysis presented deals with the real version of the LMS, i.e., a_k, X_k and H_k are real-valued. However, it can be extended to the complex case as well as to the normalized LMS, clipped algorithm and zero-forcing algorithm, without any additional difficulties.

5 Conclusion

The paper gives a review of the products of random matrices and their applications in different areas of research. It dwells mainly on two applications in control and signal processing. The one in control deals with the almost sure stabilization of jump parameter systems via the assignment of the leading Lyapunov exponent. These are only preliminary results for this important unsolved problem. The second part of the paper deals with a quite interesting application in adaptive filtering. The particularity of this problem is the finiteness of the alphabet dealt with in the filtering problem. Though this could be seen as a special case, it happens to be of great interest in digital communication. Indeed such a model is far more appropriate than classical ones. It turned out, due to the finiteness of the alphabet, that rigorous stability results can be achieved without any unrealistic independence assumptions. Both of the reviewed topics are in their infancy, especially the second one, and this paper hopes to ignite interests in both of these important and difficult research areas.

References

[1] J. Cohen, H. Kesten, and C. M. Newmann, (Eds.) *Random matrices and their applications*, AMS, Providence, RI, 1984.

[2] A. Crisanti, G. Paladin, and A. Vulpiani, Products of random matrices in statistical physics, Springer Series in Solid States Sciences, Springer Verlag, New York, 1991.

[3] J. Ezzine, and A. H. Haddad, "On the controllability and observability of hybrid systems," *Int. J. Contr.*, Vol. 49, No. 6, pp. 2045-2055, 1989.

[4] J. Ezzine, and A.H. Haddad, "On the stabilization of two-form hybrid systems via averaging," *Proc 22nd Annual Conf. on Information Sciences and Systems*, pp. 579-584, Princeton, 1988.

[5] J. Ezzine, and A.H. Haddad, "Error bounds in the averaging of hybrid systems," *IEEE Trans. Contr.*, Vol. 34, No. 11, pp. 1188-1192, November 1989.

[6] H. Besbes, M. Jaidane-Saidane and J. Ezzine, " Convergence des filtres adaptatifs dans le contexte alphabet fini." To be apper in GRETSI 1997, France.

[7] J. Ezzine, "On a practical stopping rule for the numerical computation of the Lyapunov spectrum," *Proc. IEEE 11th ACC,* pp. 1057-1059, Chicago, IL, June 1992.

[8] J. Ezzine, "An entropic approach to the numerical computation of the Lyapunov spectrum," *14th IMACS World Congress,* Georgia Tech., Atlanta, GA, 1994.

[9] H. Furstenberg, and R. Kesten, "Products of random matrices," *Ann. Math. Stats.*, Vol. 31, pp. 457-469, 1960.

[10] A.M. Lyapunov, "Probléme général de la stabilité du mouvement," 1982, Reprint *Ann. of Math. Studies*, Vol. 17, Princeton : Princeton Univ. Press, 1949.

[11] V.I. Oseledec, "A multiplicative ergodic theorem. Lyapunov characteristic numbers for dynamical systems," *Trans. Moscow Math. Soc.*, Vol. 19, pp. 197-231, 1968.

[12] B. Derrida, and E. Gardner, "Lyapunov exponent of the one dimentional Anderson model: weak disorder expansions," *J. Physique*, No. 45, pp.1283-1295, 1984.

[13] B. DErrida, K. Mecheri, and J. L. Pichard, "Lyapunov exponents of products of random matrices: weak disorder expansion. Application to locolisation," *J. Physique*, No. 48, pp. 733-740, 1987.

[14] V. D. Blondel and J. N. Tsitsiklis, "A survey of computational complexity results in systemsn and control," *Automatica*, No. 36, pp. 1249-1274, 2000.

[15] J. M. Deutsch, and G. Paladin, "Product of random matrices in a microcanonical ensemble," *Physical Review Letters*, Vo. 62, No. 7, pp. 695-699, 1989.

[16] R. Mainieri, "Cycle expansion for the Lyapunov exponent of a product of random matrices," *Chaos,* Vol. 2, No. 1, pp. 91-97, 1992.

[17] R. Mainieri, "Zeta function for the Lyapunov exponent of a Product of random matrices ," *Physical Review Letters*, Vo. 68, No. 13, pp. 1965-1968, 1992.

[18] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in systems and control theory,* Philadelphia, Siam Studies 15, 1994.

Towards the Realization of Multimedia Instrumentation based on Soft Computing Technology

Kaoru Hirota
Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku Yokohama
226-8502 Japan
hirota@hrt.dis.titech.ac.jp

Noboru Iwamatsu
Fujitsu Laboratories
Japan

Yasufumi Takama
Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku Yokohama
226-8502 Japan
takama@hrt.dis.titech.ac.jp

Abstract—**A system that infers the internal state of target objects such as cans based on the objects' multi-modal response against a given stroke and fuzzy inference is proposed. A proposed system has a hammer, by which the target object is struck, and its response against the stroke is observed with both of a microphone for auditory information and an accelerometer attached to the hammer. Internal state inference algorithm is based on the fuzzy matching using fuzzy membership data stored in advance. Experimental results show the proposed system can discriminate a plastic can from a metal can with 100%, while 3 types of internal state (amount of water; empty, half, full) can be recognized with 80% of accuracy. This paper gives the foundation of low-cost and non-destructive automatic diagnosis systems.**

Keywords— Fuzzy, Instrumentation, Multimedia, Soft Computing.

I. INTRODUCTION

A system that infers the internal state of target objects, such as cans and concrete walls, based on the object's multi-modal response against a given stroke is proposed. Internal-sate diagnosis by striking and observing target objects has been employed by experts in various fields in real world, such as palpation in hospitals, judgments of matureness of fruits like watermelons, inspection of the wall of tunnels, and the test of products in canning plants. The accuracy of these diagnoses, however, highly depends on the experience of the experts, and their know-how is mainly based on so-called implicit knowledge and difficult to transfer from the experts to novices. Although the device using echography and radiography can achieve more accurate diagnosis than human experts in some fields, their cost and difficult usage prevent them from extensive employment in various fields. Compared with the diagnosis based on such expensive and bulky devices, the diagnosis based on the fusion of multi-modal sensory information as human experts do may realize the less expensive device, which is expected to play an important role in various application fields in the real world, such the distance medical service, a diagnosis at dangerous places, and the field suffering from the shortage of human experts.

Cans filled with water are employed as target objects for the proposed system, where the quantity of water is supposed to be unknown for the system. The system has a hammer, by which the target object is struck, and its response against the stroke is observed with both of a microphone for auditory information and an accelerometer for tactile information. In the case of diagnosis by human experts, they hit the objects several times with varying the strength of hitting through the object's feedback response. The proposed system can provide the information based on which the user determine whether the additional experiment is required or not, leading to the improvement of the system's accuracy.

A configuration of a proposed system is described in Section II, while the multi-modal features required for internal-state inference are defined through the preliminary experiments in Section III. An inference algorithm is proposed in Section IV, followed by the experimental results with cans of 6 different types, which are shown in Section V.

II. SYSTEM CONFIGURATION OF INTERNAL-STATE INFERENCE SYSTEM

A configuration of a proposed internal state inference system is depicted in Fig. 1. The system is composed of mainly 3 parts, an auditory sensor part, a tactile sensor part, and an inference part. Fig. 2 shows the developed system.

In the tactile sensor part a hammer strikes the target object and the object's vibration is observed by a tactile sensor. An accelerometer MA3-04Aa (MicroStone Inc.) is employed as a tactile sensor, which is attached directly to

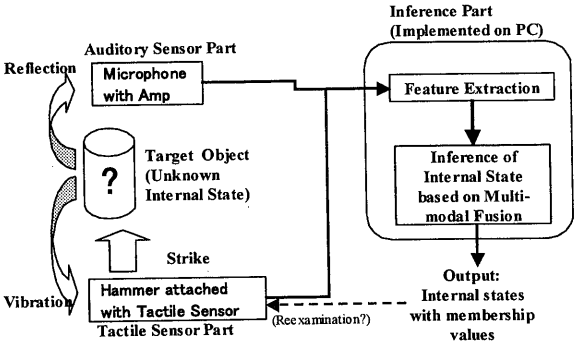


Fig. 1. System Configuration of Internal-state Inference System.

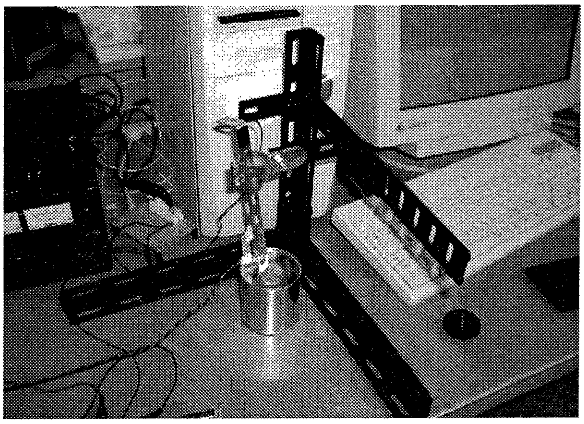


Fig. 2. Developed Internal-state Inference System.

the hammer to measure the hammer's acceleration in a vertical direction through the stroke. Its detection range, detection sensitivity, and response frequency are ±4(G), 50±10%(mV/G), and 0.8 – 1000 (Hz), respectively.

The auditory sensor part observes the sound of reflection when the hammer strikes the target object. A sound wave with less directionality than that of ultrasound is suitable for the purpose of discriminating the specific gravity or the quality of materials. The condenser microphone (maximum input sound pressure is 120dB) and the amplifier (gain is 40dB) are used in the developed system.

Both auditory and tactile information obtained in sensor parts are sent to the inference part, where a personal computer is used with an A/D converter. Several features are extracted from each of sensory information, and they are used as the input data for the inference of internal state of the target object. Since the behavior of the observed sensory information is nonlinear and complicated as shown in section III, a fuzzy rule based inference is employed. The prototype system is designed to discriminate the cans of 6 different types, by the combination with their materials (steel or plastic) and the amount of contained water (empty, half, or full). The output of the inference part is the targets object's material and the membership values of each state regarding the amount of contained water. The membership values from the system reflect the degree of confidence about the inference result, based on which the user can determine whether an additional examination (striking the object) should be needed or not. This kind of feedback is essential for the human expert-like diagnosis based on multi-modal sensory information.

III. FEATURE EXTRACTION FROM MULTI-MODAL RESPONSE AGAINST STROKE

The preliminary experiments are performed with the developed systems, in order to consider the characteristics of auditory and tactile information obtained from the target objects by giving them a stroke.

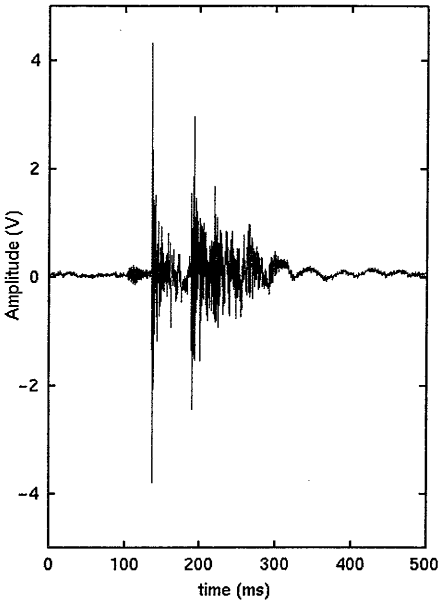


Fig. 3. Observed Sound Wave from Plastic Can with Empty.

A. Characteristics of Auditory Information

Fig. 3 - 8 show examples of sound waves observed by the auditory sensor with sampling frequency of 32KHz. The sound wave in Fig. 3 - 5 are obtained when the material of the target object is plastic with empty, half, and full

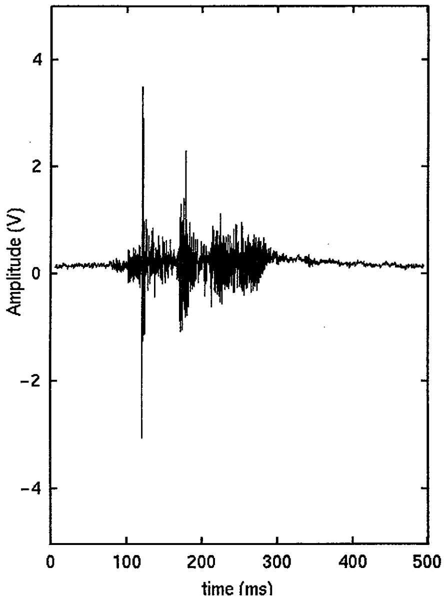


Fig. 4. Observed Sound Wave from Plastic Can with Half Amount of Water.

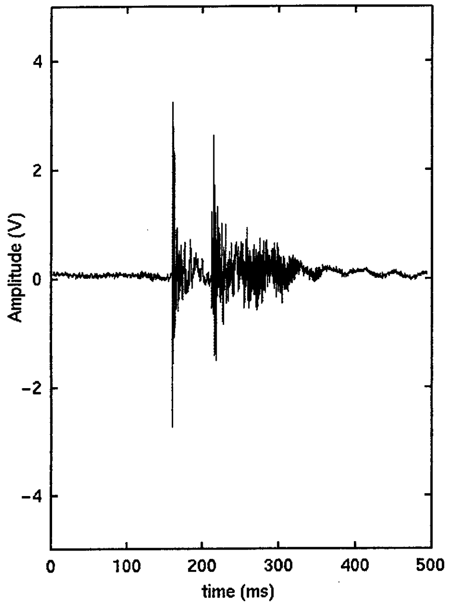


Fig. 5. Observed Sound Wave from Plastic Can with Full Amount of Water.

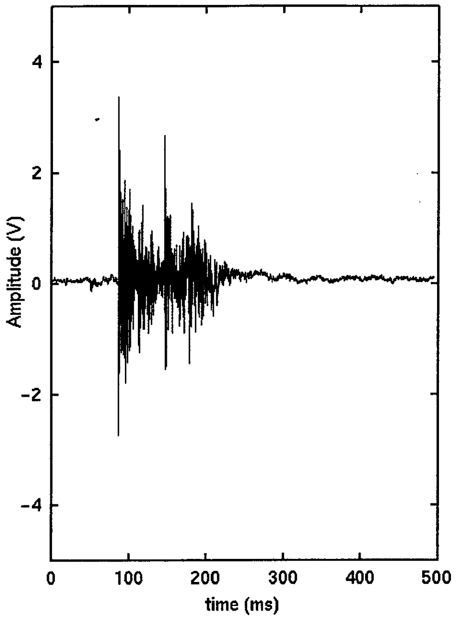


Fig. 7. Observed Sound Wave from Steel Can with Half Amount of Water.

amount of contained water, respectively.

Fig. 6 - 8 are the results when the target object is made of steel, with empty, half, and full amount of water, respectively. It can be seen from these figures that the different sound waves are observed according to the internal state (material and the amount of contained water) of the object, and its behavior is nonlinear and complicated.

When one stroke is given to the target object, the sound wave of the reflection is mainly composed of two sweeping waves [1]. The primary sweeping wave is caused by the hammer stroke given to the target object, followed by the secondary one caused by the vibration of the object's internal water. The maximum amplitude of the primary sweeping wave (A_1) and that of the secondary one (A_2), the time interval (T) between A_1 and A_2, and the decay time are varied according to both the internal state of the target object and the strength of a given stroke by hammer. A human expert can perceive the variation of these values as that of loudness and tone.

Unfortunately, the developed system gives several

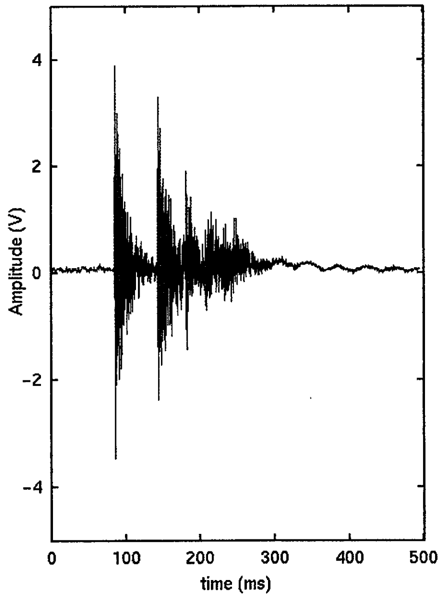


Fig. 6. Observed Sound Wave from Steel Can with Empty.

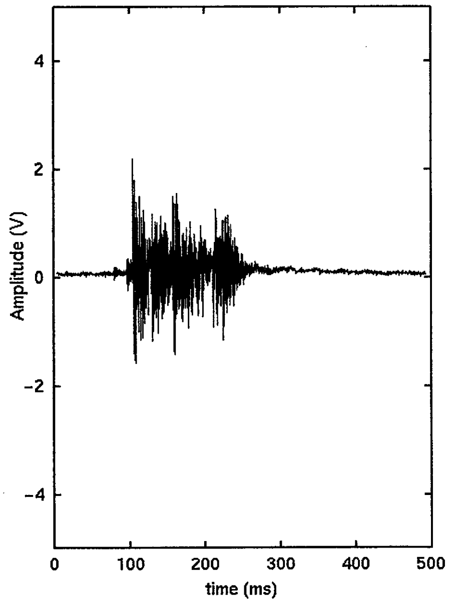


Fig. 8. Observed Sound Wave from Steel Can with Full Amount of Water.

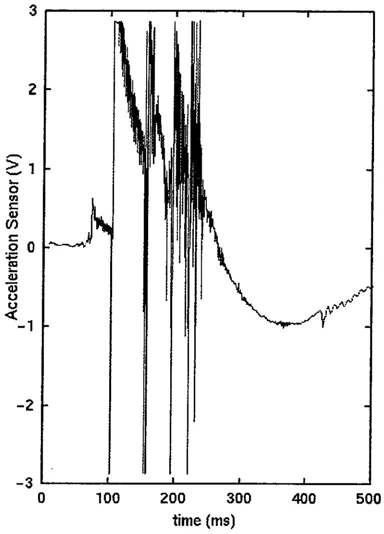


Fig. 9. Tactile Information from Accelerometer.

strokes to the target object in one trial, because the stroke is given by dropping the hammer to the target object, and the rebounded hammer gives subsequent strokes. Therefore, as shown in Figure 3 and 4, the actually observed sound wave consists of several reflections, which makes it difficult to measure the delay time of the first stroke correctly.

It is observed from Fig. 3 - 8 that when the hammer is dropped from the same height, A_1 decreases monotonously according to the amount of contained water, regardless the material of the target object. This result can be explained by the fact that the object with lower density generates a stronger vibration than another one against the stroke of the same strength.

As for the material of the target object, it is observed that the decay time of a plastic can is much shorter than that of a steel can.

B. Characteristics of Tactile Information

Fig. 9 shows an example of the tactile information that is obtained by the accelerometer attached to the hammer. The sampling frequency is 2KHz. The developed system employs the high-sensitivity accelerometer, and its obtained data are often saturated at the moment the stroke is given to the target object. Although the strength of the stroke and that of the reaction cannot be measured, the stroke point can be measured by detecting the inversion of the acceleration.

It is observed through the several experiments that the time interval between the first stroke and the second one is approximately constant when a stroke of the same strength is given to the target object with the same internal state.

The stroke point obtained from the tactile information can also be useful for the correct measurement of the auditory features from the sound wave.

C. Feature Extraction based on Multi-modal Sensory Fusion

Considering the facts obtained through the preliminary experiments, the following four features, as shown in Fig. 10, are employed as a basis for inferring the internal state of the target object.

- The maximum amplitude (**Amp_{max}**).
- The time interval (**T**) from the first stroke to the second stroke. It is obtained from the tactile information.
- The time interval (**T_2**) from when the amplitude equals Amp_{max} to the last time (within the interval T) the amplitude gets higher than the quarter of Amp_{max}.
- The time interval (**T_1**) from when the amplitude equals Amp_{max} to the last time (within the interval T) the amplitude gets higher than the half of Amp_{max}.

Only the feature **T** is extracted from tactile information. When the hammer hits the target object, the output of the tactile sensor varies from +3V to –3V, based on which the striking point can be measured correctly. Considering the noise, the point when the difference of the output values is equals to or greater than +4.5V is detected as the stroke point.

The Amp_{max}, T_1, and T_2 are extracted from auditory information. The tactile information is also used to specify

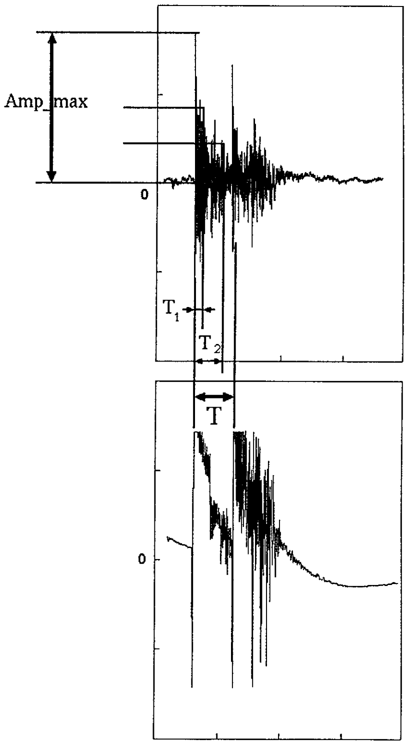


Fig. 10. Feature extraction from multi-modal sensory information: auditory information (upper) and tactile information (lower).

the stroke point. That is, the maximum amplitude is found around the stroke point, as detected from the tactile information. Considering the difference of the sampling frequency between the auditory sensor (32KHz) and the tactile one (2KHz), 512 sampling data before and after the stroke point (specified by the tactile information), total 1024 sampling data, are examined to extract $\mathrm{Amp}_{\max}$. Table 1 shows the average values of the extractet features through 100 trials. The internal-state of the target object is indicated by the following six labels.

- EP ... plastic can with empty
- HP ... plastic can with half amount of water
- FP ... plastic can with full amount of water
- ES ... steel can with empty
- HS ... steel can with half amount of water
- FS ... steel can with full amount of water

It can be seen from Table I that the value of $\mathrm{Amp}_{\max}$ and T decreases as the amount of contained water increases, while the T_1 and T_2 of the plastic can are smaller than those of the steel can. However, the variance of the obtained values is so large that it makes difficult to infer the internal state of the target object based on only a single feature.

IV. INTERNAL-STATE INFERENCE ALGORITHM

The internal state of the target object is inferred based on four features extracted from auditory and tactile sensor information, as described in Section III-C. The inference process is as follows.

1. Obtain 3 input variables from 4 features.
2. Calculate membership values for 6 internal states.
3. Identify the material of the target object based on the membership values regarding the decay time.
4. Discriminate the amount of water based on the membership values for the maximum amplitude.

A. Input variables for the Inference Engine

The following three variables are obtained from four extracted features.

$$m = \mathrm{Amp}_{\max},\ t_1 = \mathrm{T}_1/\mathrm{T},\ t_2 = \mathrm{T}_2/\mathrm{T}.$$

TABLE I
AVERAGE VALUES OF EXTRACTED FEATURES

State	$\mathrm{Amp}_{\max}$(V)	T_1 (ms)	T_2 (ms)	T (ms)
EP	4.17	0.107	22.0	56.5
HP	3.75	0.0655	6.55	54.5
FP	3.12	0.0801	1.44	53.5
ES	3.78	13.23	38.6	57.5
HS	3.41	6.05	39.0	55.0
FS	1.78	25.6	49.0	54.5

B. Membership Functions for Internal State

Six membership functions that correspond to the internal states (EP, HP, FP, ES, HS, and FS) are defined on each input variable. $\mu_A(x)$, $A \in \{\mathrm{EP,HP,FP,ES,HS,FS}\}$, $x \in \{m, t_1, t_2\}$ indicates the membership function defining the degree of x belonging to the internal state A. The membership function employed in this paper has a triangular shape with 3 parameters (*a, b,* and *c*) as shown in Fig. 11. The parameters of all membership functions are shown in Table II, which are determined based on the values in Table I.

C. Identification of Target Material

The t_1 and t_2 are used to identify the material of the target object. If $\max\{\mu_{EP}(t_1), \mu_{HP}(t_1), \mu_{FP}(t_1)\} > \max\{\mu_{ES}(t_1), \mu_{HS}(t_1), \mu_{FS}(t_1)\}$ and $\max\{\mu_{EP}(t_2), \mu_{HP}(t_2), \mu_{FP}(t_2)\} > \max\{\mu_{ES}(t_2), \mu_{HS}(t_2), \mu_{FS}(t_2)\}$ hold, its material is identified as plastic, and otherwise are considered as steel.

D. Discrimination of The Amount of Water

The three membership values $\mu_A(m)$ of the corresponding material identified in the previous step are submitted to the user as the system's output.

V. EXPERIMENTAL RESULTS

Experiments are performed to examine the developed system. Six types of cans with two materials (plastic and steel) and three internal states (i.e., amount of water; empty, half, and full) are used as the target objects. The experiments are performed under the same condition as the preliminary experiments.

Table III shows the experimental results, in which 20 trials are done for each internal state. The values in the table show the number of trials at which the proposed system infers the internal state of the target object correctly among 20 trials. As for the amount of water, the state with the highest membership value regarding m of the identified material is adopted as the system's judgment.

Table III shows that the material of the target object can be identified completely, while 80% of accuracy are achieved regarding the amount of contained water.When the system failed to infer the target's internal state correctly, it is observed that at least one of the input variables has apparently different values from the average. It seems that the examination in this case is not performed correctly, i.e., the hammer may slip on the surface of the target, or the strength of the stroke is weaker or stronger than usual. Because the proposed system is designed to be implemented with low cost, such a failure is inevitable.

TABLE II
PARAMETERS FOR MEMBERSHIP FUNCTIONS

$\mu_A(x)$	a	b	c
$\mu_{EP}(m)$	3.81	4.17	4.69
$\mu_{HP}(m)$	3.21	3.75	4.06
$\mu_{FP}(m)$	3.05	3.12	3.55
$\mu_{ES}(m)$	3.49	3.78	4.18
$\mu_{HS}(m)$	2.33	3.41	3.74
$\mu_{FS}(m)$	1.26	1.78	2.45
$\mu_{EP}(t_1)$	0	0.0019	0.027
$\mu_{HP}(t_1)$	0	0.0012	0.0052
$\mu_{FP}(t_1)$	0	0.0015	0.019
$\mu_{ES}(t_1)$	0.04	0.23	0.36
$\mu_{HS}(t_1)$	0.05	0.11	0.28
$\mu_{HS}(t_1)$	0.17	0.47	0.67
$\mu_{EP}(t_2)$	0.12	0.39	0.47
$\mu_{HP}(t_2)$	0.04	0.12	0.39
$\mu_{FP}(t_2)$	0.01	0.27	0.41
$\mu_{ES}(t_2)$	0.35	0.67	0.70
$\mu_{HS}(t_2)$	0.36	0.71	0.79
$\mu_{HS}(t_2)$	0.71	0.90	0.94

In such a case, it is often observed there is not significant difference between three membership values as the system's output. When the inference is performed correctly, the membership value of the correct state is close to 1.0, whereas other two membership values are almost equal to 0. Furthermore, the stroke-and-observation style examination is familiar for us, we can judge whether the experiments are performed correctly or not, based on the multi-modal information such as sound.

When it is found that the experiment is failure, the user can examine the target object again, because the examination of the proposed system is low-cost and non-destructive one. This is also one of the advantages of the proposed system against other high cost diagnosis methods such as echography and radiography.

TABLE III
EXPERIMENTAL RESULTS

target's state	System's output					
	EP	HP	FP	ES	HS	FS
EP	**17**	2	1	0	0	0
HP	2	**17**	1	0	0	0
FP	0	2	**18**	0	0	0
ES	0	0	0	**19**	1	0
HS	0	0	0	2	**17**	1
FS	0	0	0	0	0	**20**

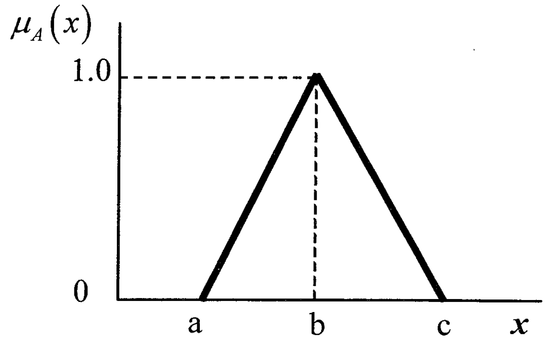


Fig. 11. Internal State Membership Functions.

VI. CONCLUSION

A system that infers the internal state of target objects based on the auditory and tactile sensor information is proposed. The proposed system requires no high cost processing such as FFT, and the fast real-time processing is possible with low cost. Experiments are performed on the cans of 6 different types by combination with 2 materials (plastic and steel) and 3 types of internal state (amount of water; empty, half , and full), and the results show that the proposed system can identify the material of targets completely, while 80% of accuracy is achieved regarding the amount of contained water. Furthermore, the stroke-and-observation style diagnosis is non-destructive and familiar for us, and the re-examination can be performed easily when the trial seems to be failure. The proposed system will be useful as a basis for low cost non-destructive diagnosis in various application fields in the real world, such as the distance medical service and the field suffering from the shortage of the human experts.

REFERENCES

[1] N. Iwamatsu. Y. Takama, and K. Hirota, "Proposal of Internal-state Inference System based on the Fusion of Auditory and Tactile Information," Proc. of 37th Workshop on Intelligent Control (Japan Society of Fuzzy Theory and Systems) SIC2000-1, pp. 1—2, 2000 (in Japanese).
[2] K. Hirota, N. Iwamatsu, and Y. Takama, "A Proposal of an Internal-state Inference System Based on Multi-modal Sensory Fusion Method," 2nd IEEE International Workshop on Intelligent Signal Processing (WISP2001), pp. 9-14, 2001 (Hungary).
[3] K. Hirota, N. Iwamatsu, and Y. Takama, "Towards the Realization of Non-destructive Inner-state Inspection System based on Fuzzy Multi-modal Sensory Integration," 2nd International Symposium on Advanced Intelligent Systems, Vol. II, No. 2, pp. 109-113, 2001 (Korea).

Scaled Vehicle Instrumentation for Intelligent Automated Highways

Lionel Lorimier
Ecole Centrale de Lille
Cité Scientifique, BP 48
59651 Villeneuve d'Ascq Cedex
France
Lorimier.Lionel@ec-lille.fr

Abdelkader El Kamel
Ecole Centrale de Lille
Cité Scientifique, BP 48
59651 Villeneuve d'Ascq Cedex
France
Abdelkader.Elkamel@ec-lille.fr

Abstract-- **Automated highways appear as a compulsory evolution to increase safety and fluidity on highways especially for heavy-duty vehicles, such as trucks, which need to circulate in a platoon mode - i.e. with a constant and reduced inter-vehicular distance. The main purpose of our project is to test in real conditions different control strategies and to determine what sensors are well adapted by introducing on purpose a redundancy for safety reasons. The 1/10 scaled prototype and the full-size platoon are in similitude so that all experimentation results led on the reduced-size platoon can be extended to the full-size platoon of vehicles.**

A four-vehicle platoon is built and the transmission issue related to the communication between vehicles in the platoon is considered in order to test the robustness of the controller. Among the controller goals are the management of the grappling and/or the unhooking of the train of vehicles, the insertion/suppression of an automated vehicle into the platoon, the fault detection and automatic reconfiguration due to failure.

Keywords-- Automated highway system, Embedded real-time control system, Multi-sensor system, Platoon of vehicles.

I. INTRODUCTION

Safety and congestion are nowadays two of the main problems on highways. The number of accidents on highways remains extremely high and several measures were considered to reduce and inflect these statistics, but their effectiveness remains questionable. A possible answer to such problem may be automation for velocity control by imposing a desired safety distance that may be smaller than the real distance in manual driving , using microprocessor performance.

The goal of automated highways is, in fact, to reduce congestion and the number of accidents due to inattention and malevolence and hence to increase safety. Many laboratories work on the automated highway topic (for example AHS in USA) and many results were derived through simulations [1, 4]. However simulations cannot substitute from real tests. Therefore a prototype including four 1:10 platoon of vehicles is developed and enables us to validate theoretical results before considering rather more expensive 1:1 tests.

The paper is organized as follow: Section 2 deals with the similitude principle associated with the physical model of a vehicle. Section 3 presents the multi-sensor system embarked in vehicles. The transmission systems are presented in Section 4 and Section 5 describes the controller. Different control laws are described in Section 6 before concluding the paper with the future work.

II. SIMILITUDE PRINCIPLE

To compare and apply results derived from a scale platoon to a full-size vehicle, it is necessary to make the experimentation in similitude – that means that we have to respect some static, cinematic and dynamic rules. These rules are extracted from the non-linear equations that govern the evolution of the platoon. Figure 1 details the process of similitude from full-size vehicle characteristics to reduced prototypes up to test results extrapolation from reduced platoon to a full-scaled platoon.

So far, a model of real vehicle was developed [2] from non-linear mechanical equations of dynamics, afterwards this model was linearized in order to develop control laws. In fact, to control a platoon of vehicles in a highway, we have to determine precisely the lateral and the longitudinal dynamics.

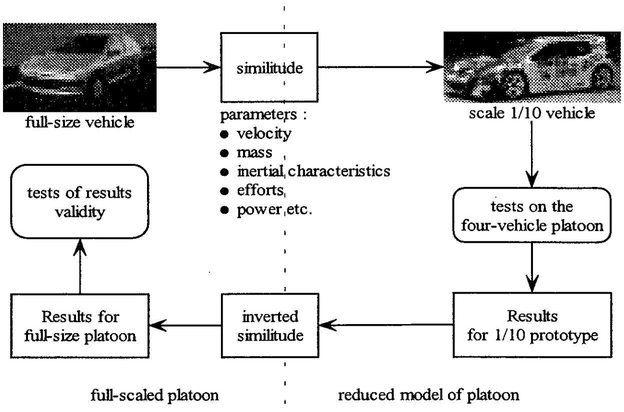


Figure 1 : the similitude principle

The dynamic similitude is applied on the non-linear model of vehicle to get the similitude parameters. The similitude parameters are ratios (e.g. the Reynolds number for a similitude in fluid mechanics) that have to be identical for the prototype and the full-size vehicles to guaranty results validity on a full-size platoon of vehicles.

This work was partly supported by ANVAR and INRETS, France

Let y represent the vehicle lateral position we obtain from the non-linear model analysis:

$$y=F(m,I_z,V,l_1,l_2,\gamma,C_{\alpha f},C_{\alpha r})$$

where m is the mass of the vehicle, I_z the moment of inertia on the vertical axis, V the velocity of the vehicle, l_1 and l_2 are respectively the length from the Center of Gravity to front axle and rear axle, γ is the steering angle, and $C_{\alpha f}$ (respectively $C_{\alpha r}$) is the cornering stiffness of front tires (respectively of rear tires)

As the set Mass m– Distance l- Velocity V is a primary system, all physical units can be expressed according to these units. The expression of the parameters in the default (mass, length, time) system and primary system can be seen in table 1. Then we recast as dimensionless all the other parameters using a dimensional analysis.

As an example, the dimensional equation for Time in the primary system is:

$$T\approx\frac{L}{V}$$

So replacing the time by this expression in other dimensional equations, we obtain the dimensional equations of each parameters in the primary system:

$$C_\alpha\approx MLT^{-2}=ML\left[\frac{L}{V}\right]^{-2}=ML^{-1}V^2$$

therefore the ratio $\frac{C_\alpha}{ML^{-1}V^2}$ is dimensionless.

Following this step, we can reduce the number of parameters using Federmann's Theorem. It yields a new expression of the lateral position function depending on parameters called similitude parameters.

$$\frac{y}{L}=F\left(\frac{I_z}{mL^2},\frac{l_1}{L},\frac{l_2}{L},\gamma,\frac{C_{\alpha f}L}{mV^2},\frac{C_{\alpha r}L}{mV^2}\right)$$

Table 1 : expression of similitude parameters in the default (m, L, T) system

	Mass	Length	Time	Mass m	Length L	Velocity V
M	1	0	0	1	0	0
L	0	1	0	0	1	0
V	0	1	-1	0	0	1
L_1	0	1	0	0	1	0
L_2	0	1	0	0	1	0
γ	0	0	0	0	0	0
$C_{\alpha f}$	1	1	-2	1	-1	2
$C_{\alpha r}$	1	1	-2	1	-1	2
I_z	1	2	0	1	2	0

Each similitude parameter has a physical meaning. The first ratio represents the mass distribution and repartition in the vehicle, the second two ratios symbolize the geometric reduction of the vehicle – since we impose a geometric reduction, it is not necessary to consider parameters such as tire radius, etc. The fourth parameter is the steering angle, and finally the two last parameters characterize ratios of available steering moment to kinetic energy.

Using the same principle, we can determine a function to express the inter-vehicular distance using the longitudinal control law. The similitude parameters are quite different, and take into account the characteristics of the electric motor of the vehicle.

$$d_{i-1\to i}=F(m,I_z,V_{i-1},a_{i-1},V_i,l_1,l_2,f_{tire\to road},\tau_{motor})$$

where $f_{tire\to road}$ characterize the contact between the road and the tires (friction), and τ_{motor} represent the dynamic of the motor, V_{i-1} and a_{i-1} are carried out to take into account the behavior of the preceding vehicle, but do not implies new similitude parameters because all the vehicles are in dynamic similitude.

Step 1 : Determination of the physical parameters of the model

We impose a physical similitude on the lateral and longitudinal dynamics of the vehicle. By applying such a similitude on the model, we obtain a representation of the prototype dynamics, which have to be adapted for several reasons:

First, the inertia of the vehicle is very important for the lateral control, and the 1:10 model cannot be built under similitude conditions. Its dynamics are faster, and the inertia is rather weak. Furthermore, the contact between tires and ground is not the same and the steering system is simplified. It implies that we have to determine experimentally the dynamics of the lateral behavior of each vehicle so as to adapt the control laws.

The scale reduction has a consequence on time as well. There is in fact a similitude factor between time for a real vehicle and time for the prototype. The relation between distance and velocity and the similitude factors on these primary parameters impose this factor. Indeed, prototype reference time requires rapid measures of distance otherwise there is not enough information on distance to keep a good stability of the platoon. As the infrared telemeter needs a 70ms period for each distance measure, artificial methods must be developed to obtain more information. Section 6 presents one of them: the inter-vehicular distance estimator.

The 1:10 scale imposes some restrictions to the choice of the sensors too. In fact, some sensors such as camera, radar, laser telemeter are not well adapted to our model for different reasons: On one hand the camera requires too many calculations for our processor, and on the other the radar and the laser telemeter are too big to be installed in these vehicles.

The last influence of scale factor is on magnetic field measurement. The plots are small and so is their magnetic field (2 gauss at 3.5 cm on symmetric axis). Consequently, the magnetometers must not be perturbed by Earth magnetic field.

Step 2: Tests on 1/10-scaled vehicles
The controllers are determined for the reduced models using Matlab by simulation. After that, the tests are performed with the automated highway prototype and we can observe the results in real time, and also retrieve the data on a computer to plot the temporal evolution of parameters.

Step 3: Extrapolation of results to verify coherence
Finally, once the so obtained results are satisfactory, we transpose them to full-scale vehicles with the similitude parameters and reduction ratios.

III. MULTI-SENSOR SYSTEM

In order to perform the control of a vehicle in a platoon configuration, we have to use two classes of sensors. The first class concerns the lateral control, and the second class includes sensors for the longitudinal control. Transmission systems have to be used in order to improve the stability of the platoon. In fact, we do not communicate the velocity and the acceleration of the leading vehicle to all the other vehicles in the platoon; but only between one vehicle and the following one. This communication system will be presented in Section 4.

A. *Sensors for lateral control*

1) *Magnetometers*

Taking into account that the magnetic field delivered by magnetic plots is weak, we have decided to use magnetometers from Honeywell. These sensors give an output voltage signal proportional to the magnetic field in which they operate. The maximum measurable field is about 2.2 gauss ($2.2 10^{-4}$ T) and two special straps are available: the first one to reset the magnetometer in order to eliminate disturbances due to an external perturbing magnetic field such as Earth field. The second one enables to set up the offset. The response of magnetometer is highly sensitive and linear. Three magnetometers are used: two in the front side of the vehicle, and one in the rear side. The optimal vertical distance between plots and magnetometers was evaluated first using a finite-element modelling, and then via an experimental test. It yields a distance of 4 cm by modelling, and a real distance of 3.5 cm after experimentation.

Figure 2 defines the three-axis base system for the platoon of vehicles.

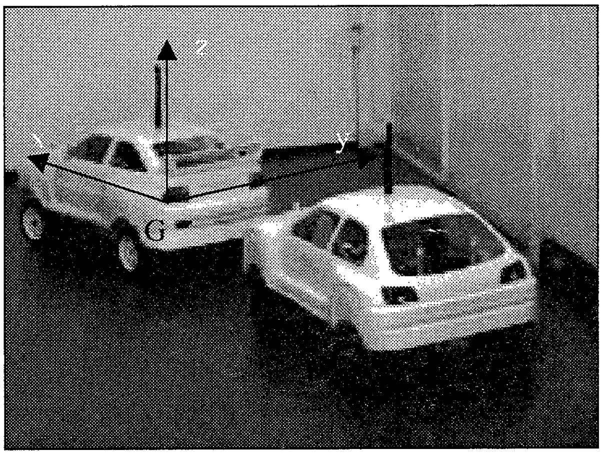


Figure 2 : definition of vehicle's axis

x : longitudinal axis, y: lateral axis, z: vertical axis

- ☐ First method to determine the lateral position: one-axis measurement

In this case we measure the vertical component of the magnetic field delivered by plots given the vertical position z=3.5 cm. The magnetic field signal on figure 3 was measured using one of the magnetometers and a numerical oscilloscope.

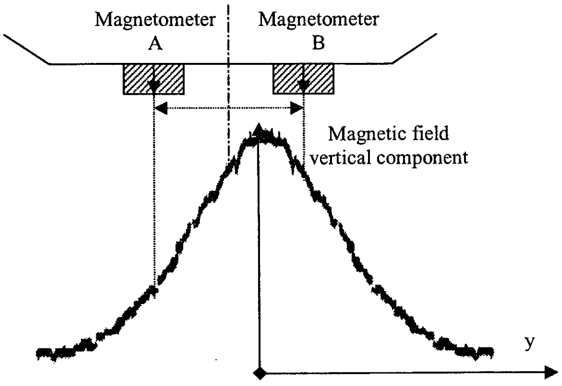


Figure 3 : Magnetic field =f (Lateral position)

The lateral distance between the two magnetometers (d_y) equals to 4cm. As we measure an absolute value of lateral position on each magnetometer, we obtain the sign of both measures knowing that the lateral shift is always inferior to d_y, and that y_A is initially positive whereas y_B is negative as indicated in figure 4.

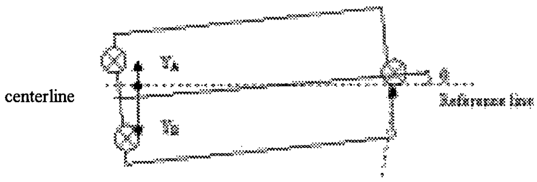


Figure 4 : Identification of the position variables

- ☐ Second method: two-axis measurement

In this case we measure the vertical and the lateral components of the magnetic field delivered by plots. The vertical component B_z gives us the absolute lateral deviation from the reference centerline. The sign of this deviation is obtained by the measure of the lateral component B_y as shown in figure 5 and 6.

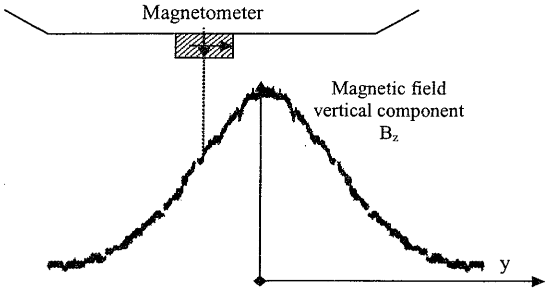


Figure 5 : vertical field measurement

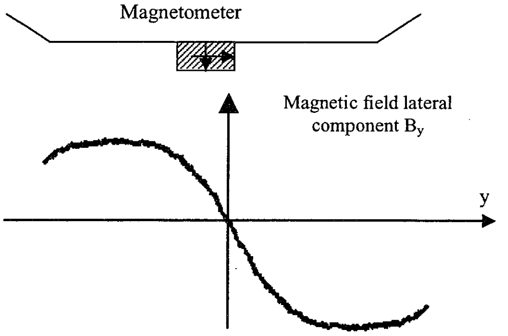


Figure 6 : lateral field measurement

2) *Sensor for real steering angle*

A linear sensor of angular rotation is used to measure the steering angle. We have to know precisely its value to optimize the control by use of a servomotor, this is used as a feedback for the lateral control.

3) *Interface between sensors and the microcontroller*

Several other electronic functions were introduced. The triggers are used to interrupt the microcontroller and capture information from magnetometers. In order to have the most reliability, Set/Reset strap of the magnetometers has to be set with a 3A-current, and an analog differential amplifier gives an output between 0 and 5V. Finally, 12-bits AD converters have been added to improve the resolution of the field measure and then of lateral position measurement since on-board AD converters have just a 10-bit resolution. Figure 7 localizes the different sensors and electronic boards for the lateral control.

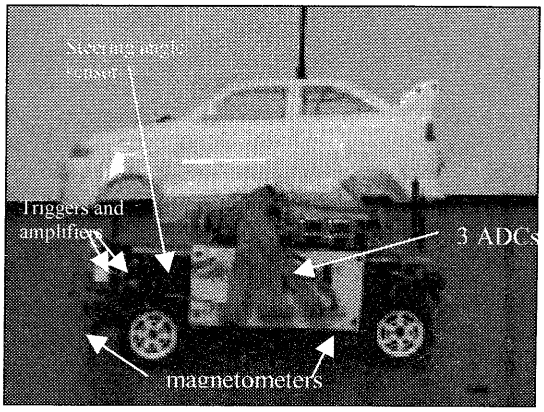


Figure 7 : sensors for lateral control

B. *Sensors for longitudinal control*

Longitudinal control is very important, mainly to avoid crashes. Accessing the distance between vehicles, and then following this information will enable us to determine whenever the i^{th} vehicle can pick up the train of vehicles or enter into collision with the preceding vehicle.

1) *Infrared Telemeter*

We can determine the distance between vehicles by measuring the shift between the outgoing infrared wave and the phototransistor that receives it after reflection on the previous vehicle or another obstacle as illustrated in figure 8.

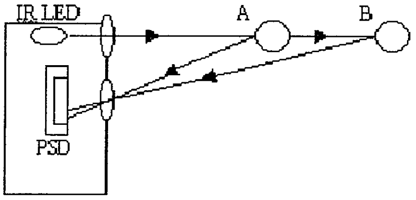


Figure 8 : Infrared sensor

This sensor delivers an 8-bit serial word, and the microcontroller has to capture this word and transform it into a distance. Each sensor have to be calibrated: after measuring each serial word delivered for known distances between vehicles, we interpolate the data with a sixth-order polynomial function. The results are plotted in figure 9.

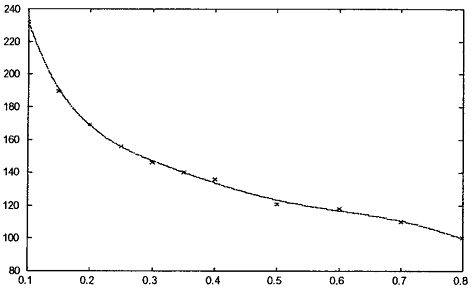


Figure 9 : correlation 8-bit word / distance

We can obtain a distance measure every 70 ms – this time includes data acquisition and treatments to obtain the serial word. In order to improve the stability of the platoon, we will show in Section 6 that an estimator has to be introduced to estimate the distance in a smaller period.

2) *Ultrasonic Telemeter*

The propagation time of a short ultrasonic impulsion is proportional to the distance between vehicles. The formula to obtain distance from measure is :

$$d = \frac{1}{2} c * t \quad c = \sqrt{\gamma * r * T}$$

c : sound velocity
t: time of propagation
γ=1.4 and r =287 J/(mol.K)
T : temperature in Kelvin

This method, using time propagation, is less precise than the deviation method used in infrared telemeter. Moreover, temperature variations can shuffle the distance measure.

3) *Optical Velocity Sensor*

Given the radius of the rear tire and rotational velocity of the drive shaft, we can access to the velocity of vehicle. The following relation yields the rotational velocity ω:

$$\omega = \frac{2\pi}{n * T_0}$$

where T_0 is measured with a timer and n is the number of features per 360°

The velocity of the vehicle results from the classical formula $v=\omega * R_{tire}$

4) *Accelerometer*

One timer of the microcontroller measures the pulse width T1, and the period T2, as shown in figure 10, is fixed with a potentiometer and a resistor. The acceleration is proportional to the pulse width of the signal.

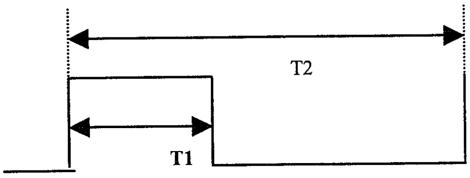


Figure 10: measure of acceleration

The relation between A and T1 is given by this formula: *A (in g) = (T1/T2 – 0.5)/12.5%*

It is interesting to note at this level that we have decided to place an infrared sensor and sonic transducers to study the interference between them, and hence to create a redundancy for measuring distance. At our scale, these sensors were the most appropriate solutions.

C. *Data Fusion*

A data fusion has to be carried out using for instance fuzzy data fusion. In fact, we have redundant information about the distance between vehicles:

- Infrared Telemeter
- Ultrasonic Telemeter
- Integration of relative velocity
- Double integration of relative acceleration

Besides we have to take into account in the fusion process the fact that each data has its own precision and confidence.

IV. TRANSMISSION SYSTEMS

A. *Transmission between vehicles*

The signal coming from the i^{th} vehicle intended to the following $(i+1)^{th}$ vehicle, should not interfere with the signal coming from the $(i+1)^{th}$ vehicle, intended to the following $(i+2)^{th}$ vehicle. In fact the $(i+2)^{th}$ vehicle needs only to consider the signals from the $(i+1)^{th}$ vehicle: velocity and acceleration.

In order to eliminate the interference phenomena, the following solution is proposed: We use one frequency bearer and then code the signal to be sent to achieve a system of identity of vehicles while also coding the information transmitted on the frequency bearer. Such a system can be applied to a platoon of many vehicles (it is not the case of a system using different frequencies to communicate). The code could be transmitted from one vehicle to another by a system of identification, which has only to send data for the vehicle behind (weak-range system). Thus, it is possible to modify the order of vehicles in platoon without generating transmission problems. The code are attributed when a vehicle enter a section of the automated highway so that there is no possibility to have the same code for two vehicles, this attribution process is managed by the section supervisor.

☐ How the transmission between vehicles works

Each vehicle has its own transmission module named ASC0 (see Embedded Control System), and this module generates the correct signal to be sent simply by transferring a word in its transmission buffer. The transmission between vehicles uses this protocol :

- 8-bit word
- one bit stop
- no parity
- max. 50 kbauds

Each vehicle has also a specific architecture of encoders and decoders described in figure 11 that enable them to send a piece of data to either a specific vehicle, or all vehicles in a platoon – this is very useful to enable the lead vehicle to send information to all the vehicles in its platoon but also the other vehicles to communicate with their following without problems of identification.

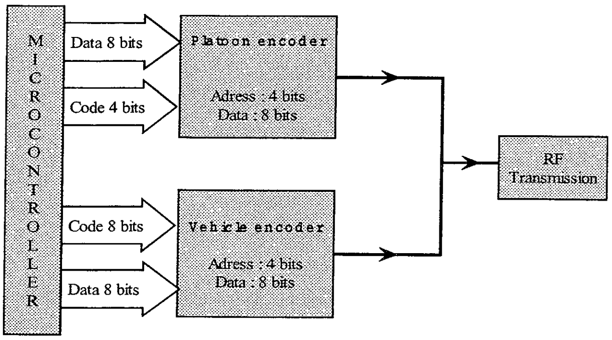


Figure 11 : transmission architecture

8-bit data are either velocity information, acceleration information or emergency information.

When a vehicle have to send data to a specific vehicle in the section, it carries it through the vehicle encoder and need to provide the a 8-bit code, and the platoon encoder is designed to send a data to every vehicle in the platoon by specifying the a 4-bit code common to each vehicle in the platoon.

Furthermore, the microcontroller has special flags (interrupts) to know whether a word is received or transmitted.

B. *Vehicle – PC supervision*

The transmission module ASC0 sends successively the data for supervision and the data about the inter-vehicular information since the baud rate of the radio transmission is sufficient. The bearer of the vehicle-to-PC transmission is 868 MHz, and the receptor is connected to the serial port of the PC through a serial RS232 driver.

The data are captured on the PC using a software developed in C Language with two possible modes:

- Capture N data and when these data are acquired in a dynamic list, the transfer to Matlab for calculations and plotting is launched.
- Each data is saved in a file with a name finishing with the number of the data. Another program under Matlab wait for the data, and then plot all the data in real time

Using such a transmission, we can follow in real time many variables such as the inter-vehicular distance, the trajectory, the velocity, etc.

V. EMBEDDED CONTROL SYSTEM USING ONE 16-BIT MICROCONTROLLER

The System architecture is defined around a 16-bit embarked microcontroller cadenced at 20 MHz with many functions such as analogue to digital converters, pulse width modulation module to command the velocity controller of the electric motor and the steering servomotor. Such an embarked system maximizes the reactivity of the vehicle and limits the delay to transfer information. Considering the complexity of calculations to carry out, a powerful, fast and low power calculator is necessary.

The capture of information coming from sensors is managed by microcontroller and we have developed a software to accomplish this goal. For example, the magnetometers trigger measurement when the vehicle goes through the magnetic field of a plot thanks to interrupts. However other information such as velocity, acceleration and inter-vehicular distance, are managed by microcontroller events that launch a capture when needed. Afterwards the microcontroller makes data fusion and computes the orders derived from control laws. It controls also transmission module for inter-vehicular and vehicle to PC transmission. Using the PWM module, we can command the velocity controller and the steering servomotor, which are isolated from microcontroller by two opto- isolators.

VI. CONTROL OF THE PLATOON

A. *Lateral control*

Several control laws (PID Anti-windup, optimal controller...) have been developed [1], and had to be adapted to our prototype according to the similitude principle explained in Section 2. Then we have implemented and coded in C Language, an appropriate control law using Matlab and Simulink. The last step is to transfer the application in the Flash memory of each microcontroller.

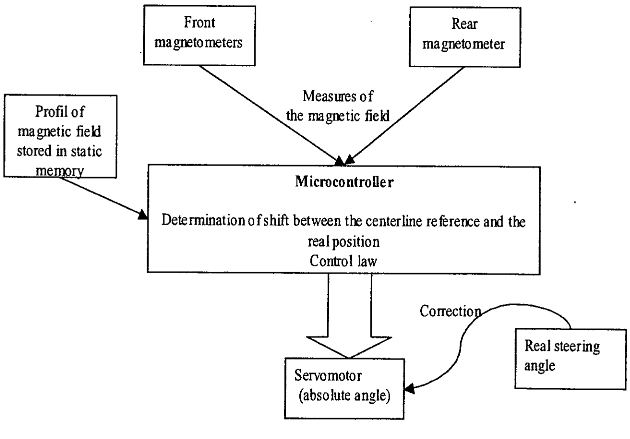


Figure 12 : How the lateral control works

Figure 12 shows the organization of sensors around the microcontroller that computes the control law in order to control the lateral behaviour.

B. Longitudinal control

Longitudinal control by exact linearization and optimal control have been developed [1]. In our case, only the control by exact linearization was adapted to our prototype following the same protocol as lateral control. The lead vehicle has to follow a velocity profile whereas the other vehicles in the platoon must follow with a constant inter-vehicular distance using the information presented in figure 13.

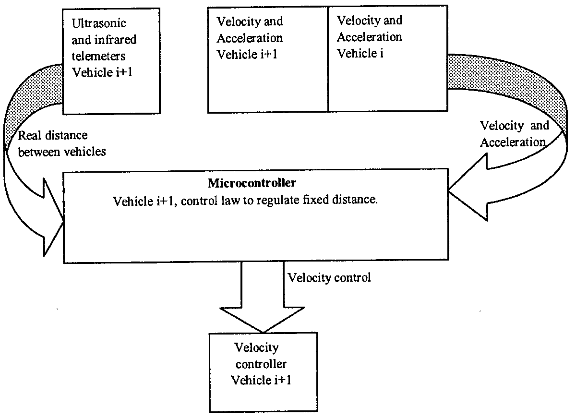


Figure 13 : How the longitudinal control works

We have faced some instability in the previous system because vehicles did not receive distance information rapidly enough. The solution was to introduce an estimator, detailed in figure 14, to estimate the inter-vehicular distance from a model of longitudinal dynamics, the relative velocity $v_{relative}=v_{i+1}-v_i$ and the relative acceleration $a_{relative}=a_{i+1}-a_i$.

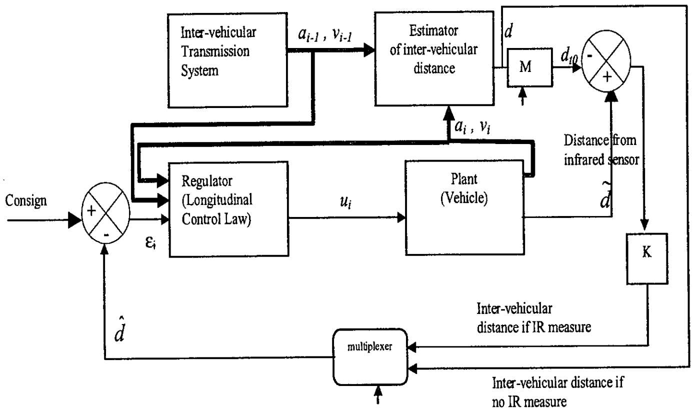


Figure 14: The estimator for the inter-vehicular distance

At t_0, a measure on the infrared sensor (IR measure) is launched. As soon as we obtain the distance measure $\tilde{d}_{t_0}$ from the infrared sensor, we update the estimated distance d_{t0+T} given by the estimator by the following formula:

$$\hat{d}_{t+T} = d_{t+T} + K(\tilde{d}_{t_0} - d_t)$$

$$d_N = d_{t0+T}$$

where K is the updating factor and T is the period of the infrared sensor. During this period, N estimations of the inter-vehicular distance are computed. At t_0+T, the distance used in the regulator is d_{t0+T}.

VII. CONCLUSION

We present in this paper an intelligent instrumentation based on an embarked microcontroller and a multi-sensor system. The principle of similitude enabled us to adapt all theoretical results to our prototype and then to extrapolate the results of the tests led on the automated highway prototype to full-size vehicles. This prototype can follow with a good precision a line defined by magnetic plots and keep a fixed inter-vehicular distance.

We plan to install a camera system for the longitudinal control so as to add another redundancy in order to enhance safety and introduce a fault detection system. Furthermore, a fault tolerant system that has to detect which sensor is deficient and hence to decide whether the vehicle can stay on the highway, will be introduced. Finally, the identification system used for the coded radio transmission of information will be extended to the supervisor, hence enabling the real-time following of every vehicle in the platoon but also to achieve some computation (predictive control, etc.) since the supervisor will keep every information it gets in a database.

REFERENCES

[1]- Ines Kammoun (supervised by Profs. P. Borne and A. El Kamel), *Approches de commande de véhicules et conception d'une plate-forme de simulation d'autoroutes automatisées* EPT–ECL 98

[2]- A. El Kamel, F. Bouslama, P. Borne, *Multimodel Multicontrol of Vehicles in automated Highways,* 4th World Automation Congress WAC'00, Hawaii, USA, Juin 2000

[3]- Karl W. Olson, *Wire-Reference Configurations in Vehicle Lateral Control.*, IEEE Trans. on Vehicular Technology, Vol. VT-26, N° 2, 1977.

[4]- R. E. Fenton, R.J. Mayhon, *Automated highway studies at the Ohio State University: an overview*, IEEE Trans. on Vehicular Technology, Vol. 40, N° 1, pp. 100-113, 1991

An Optimal Selection Problem with two Decision Makers

F. Ben Abdelaziz
Institut Supérieur de Gestion
Tunis
41, Rue de la Liberté, Le Bardo
University of Tunis
Tunisia
fabdelaz@isg.rnu.tn

S. Krichen
Institut Supérieur de Gestion
Tunis
41, Rue de la Liberté, Le Bardo
University of Tunis
Tunisia
nhnsnns@sig.run.th

Abstract— **We investigate in this paper a dynamic formulation for the bilateral optimal selection problem. The two decision makers observe a sequence of n offers, one at a time, without allowing the recall of previously inspected offers. The main concern of the two decision makers is to select jointly an offer. If either player decides to stop sampling, then the selection process is over. The no-choice option is not allowed, therefore the last offer should be accepted whatever its rank might be for both players. We propose in our definition of the problem a utility function for each player that takes into account the rank of the opponent for the selected offer. The backward recursive equations are solved for a special case of the utility and the optimal strategies of the players are then derived.**

Keywords— **Dynamic programming, Optimal selection, Optimal strategy, Bilateral problem.**

I. INTRODUCTION

We study in this paper a sequential decision problem known as the "optimal selection problem" (OSP) or the "secretary problem". It is characterized by the possibility of stopping the selection process at each stage. Examples arise in financial applications as the problem of purchasing assets in stock exchanges. Other practical sequential decision situations are to be modeled as an optimal selection problem, for example, the interview of a known number of people for a job, the selection of projects, military targets and resource alternatives (Nakai [1986]). The above examples can be good motivation for lending a special attention to this class of OSPs and the possibility of its extension to the bilateral framework.

The basic OSP is characterized by a pool of n offers that are to be evaluated sequentially, one at a time, by a single decision maker (DM) regarding only one criterion. The arrival of offers does not follow any probability distribution. Therefore, at each stage, the currently inspected offer is ranked regarding the offers so far observed. In this case, the utility of the DM should be expressed in terms of the absolute rank. In the basic OSP, the utility of the consists in choosing the best offer of the sequence. Hence, the DM is rewarded by 1 if his absolute rank of the selected offer is 1 and 0 otherwise. The DM is allowed at each stage of the process one of the following options: accept the current offer and stop sampling or refuse it and observe the next offer. Each inspected offer cannot be re-examined in later stages. As the basic problem does not allow the no-choice option, then the last offer must be accepted once inspected. Lindley [1961] and Dynkin [1963] stated and solved the basic OSP. A reference paper which capitalizes on the basic problem and some of its variations is due to Gilbert and Mosteller [1966].

A variation of the OSP consists in involving two DMs with different evaluation criteria in the selection process. The OSP with more than one DM can be considered as a kind of game involving two players. The solution to the game is then an equilibrium situation considering the problem assumptions and the degree of cooperation between the players. Previous works in the bilateral framework where each player has to select his individual offer belong to Sakaguchi [1984, 1985a, 1985b] and Enns and Ferenstein [1985]. They assume that player I is given the priority of selection. The problems were mostly zero-sum since the main concern of each player is to select an offer better than that of his opponent.

For some decision situations, it is more suitable to formulate the BOSP assuming that both players decide to select the same offer. Sakaguchi [1978] studied this problem and assumed that the offers arrive according to a known probability distribution. The utility of each player corresponds to the value of the offer. Each player in Sakaguchi's problem wishes to maximize his individual expected value of the r selected offers. An offer is selected only if both players accept it. This kind of decision is called a unanimous decision.

Salminen et al. [1996] proposed an OSP with implicit utility functions for the players. Accordingly, pairwise comparison between the offers are required to determine of the form of the utility functions. The authors assume that the group members are allowed the recall of previously examined offers while a cumulative cost of inspection is considered. To solve this version of the OSP in the group context, an interactive algorithm for negotiation to reach a compromise was outlined.

In this paper, we propose an alternative definition for the bilateral optimal selection problem (BOSP) where the two players are required to select the same offer. We assume that the offers are inspected one at a time without any knowledge about the probability distribution of their arrival. The main characterization of the problem is that each player can stop the selection process by accepting the current offer. The utility of each DM is expressed in terms of the choice of his opponent. Therefore we can say that each DM wishes to maximize the two utilities while assigning different importance. In this case, the problem is called a bivariate BOSP. We enumerate the main features of the problem then, provide its dynamic formulation. The backward recursive equations are solved for a special case of the proposed utility and the optimal strategy in then derived.

The present paper is made up of the following sections: section 2 defines the basic OSP in the single DM framework. Section 3 provides our definition of the BOSP and the utilities of the DMs to be maximized. It also analyzes the dynamic formulation of the problem and derives the optimal strategy for each player. This paper is enclosed by several extensions of the problem.

II. SINGLE DM OPTIMAL SELECTION PROBLEMS

The standard version of the OSP is defined by a sequence of a prefixed number of n offers which are available for a single DM. The DM is allowed exactly one acceptance with the object of choosing the best offer of the sequence. The DM is interested by nothing but the best offer, therefore his utility is 1 if the selected offer is the best and 0 otherwise. Mathematically the utility of the DM, expressed solely in terms of the absolute rank k, is the following:

$$v(k) = \begin{cases} 1 & \text{if } k = 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The main features that characterize the basic OSP are:

1. A sequence of a predefined number of n offers are inspected by a DM.
2. The offers are examined one by one, in a random order.
3. The arrival of offers does not follow any known probability distribution.
4. A stopping control of the process is available at each stage: accept the current offer and stop sampling or discard it and continue the drawing process.
5. No recall of previously examined offers is permitted.
6. The no-choice option is not allowed, hence if the last offer is inspected it must be accepted.
7. The aim of the DM is to pick the best offer of the sequence.

Due to the stochastic nature of the OSP, the solution is expressed as a decision strategy which is derived (Lindley [1961], Dynkin [1963] and Gilbert and Mosteller [1966]) from the following backward recursive equations at any stage i ($i = 1, \ldots, n-1$), when the ith offer has a relative rank k:

$$Ev(i,k) = \begin{cases} max\{Ev_s(i,k), Ev_c(i)\} & \text{if } k = 1 \\ Ev_c(i) & \text{if } k = 2, \ldots, i \end{cases} \tag{2}$$

The backward recursive equations express the stopping control at each stage. So, according to the expected utilities, the DM should decide whether to stop or continue the choosing process. This option is available only if the relative rank of the ith offer is 1. Otherwise, he automatically continues sampling.

$Ev_s(i,1)$ is the expected utility of the DM if he decides to stop by accepting the ith offer. It is as well the probability that the ith offer is the best among the n offers such that it is the best among the i first offers already presented, that is:

$$Ev_s(i,1) = P\{(n,1)/(i,1)\} = \frac{i}{n} \tag{3}$$

$Ev_c(i)$ denotes the expected utility of the DM when he decides to discard the ith offer and continue sampling by following the optimal strategy:

$$Ev_c(i) = \frac{1}{i+1} \sum_{j=1}^{i+1} Ev(i+1, j) \tag{4}$$

By solving the above dynamic equations, the optimal strategy of the standard OSP is derived: the optimal policy consists in rejecting the first $(i^* - 1)$ offers such that i^* is the least such offer satisfying the following inequality:

$$\frac{1}{i^*} + \ldots + \frac{1}{n-1} < 1 < \frac{1}{i^*-1} + \ldots + \frac{1}{n-1} \tag{5}$$

and accepting the first offer thereafter with relative rank 1.

Several variations of the problem were studied in the literature while modifying some of the properties of the basic version. Pressman and Sonin [1972] assumed that the number of offers is random. Smith [1975] and Tamaki [1991] studied the case where the offers can be available with a fixed probability. The arrival of offers can follow a known probability distribution as in Albright [1977], Bruss and Ferguson [1993], Cowan and Zabczyk [1978], Bruss [1987], Gnedin [1996], Degroot [1968] and Albright [1977]. Chow et al. [1964] considered an alternative utility function for the DM that consists in the minimization of the expected rank of the selected offer. For review papers one can see Freeman [1983] and Rose [1984].

In the next section we propose the bilateral extension of the problem and provide a dynamic formulation that yields the optimal strategy of the DM.

III. NOTATION

The following notation accounts for the development of the backward recursive equations in the subsequent sections:

- n: the total number of offers to be inspected.
- i: the stage number ($i = 1, \ldots, n$).
- k_p: the relative rank of the current offer according to player p ($p = 1, 2$).
- $k_{p'}$: the relative rank of the current offer according to player p' ($p' = 1, 2$), the opponent of player p.
- $Ev_s^p(i, k_p, k_{p'})$: the expected utility of player p when stopping at stage i and selecting the current offer with relative ranks k_p and $k_{p'}$.
- $Ev_c^p(i)$: the expected utility of player p when postponing the decision to the next stage.
- $Ev^p(i, k_p, k_{p'})$: the maximum expected utility of player p by following the optimal strategy when the ith offer, with relative ranks k_p and $k_{p'}$, is being examined.
- $\psi_i = \sum_{j=i}^{n-1} \frac{1}{j}$.
- $d^p(i) = \frac{E_c^p(i)}{i}$.

IV. A BILATERAL OPTIMAL SELECTION PROBLEM

A. Problem assumptions

Many decision situations require more than a single DM. This class of OSPs including more than one DM in the selection process brings a great deal of complexity. Such a situation gives rise to conflicting situations whenever the group members don't agree with the same decision. Several formulations account for the OSP with more than a single DM. Sakaguchi [1978] proposed a formulation of the BOSP assuming that both players choose the same offer knowing the probability distribution of the arrival of offers. The priority of decision is given to player I. Sakaguchi solved this problem by developing and solving the backward recursive equations for each player. In contrast to the dynamic approach, Salminen et al. [1996] proposed an interactive algorithm that included a negotiation step between the players. They assumed that the players have to select the same offer. In their formulation they supposed that the utilities are implicit and that only a partial order of the offers can be made. We investigate in what follows a new class of BOSPs. In our formulation of the problem we assume that both players are required to choose the same offer, but without any knowledge about the probability distribution of the arrival of offers. Each player has his own criterion for the evaluation of the offers. Consequently, each inspected offer has a bidimentional relative ranking. The complexity of the BOSP follows from the divergence of the criteria considered by the players.

We investigate in this part the BOSP in which two players with different evaluation criteria are required to select the same offer from a stream of n offers arriving sequentially one by one in a random order. We shall be interested in the present section by the BOSP characterized by the fact that the selection process is over if either player wishes to stop. Arriving offers are firstly examined by player I, if player I accepts the offer, the game is over. However, if player I decides to reject it then it is player II's turn to exercise one of his options: either to accept or refuse the current offer. If player II accepts the current offer, then the play is stopped, otherwise the next offer is examined as mentioned earlier.

Regarding the previous description, our BOSP can be well described through the following features:

- two players with different evaluation criteria observe a sequence of n offers, one by one, sequentially and randomly.
- the two players select the same offer.
- there does not exist any information about the probability distribution of the arrival of offers.
- each player aims to maximize his utility function expressed in terms of the preferences of his opponent.
- if either player decides to accept a currently examined offer, then the overall decision should be to accept it.
- the priority of decision is assigned to player I.
- no recall of previously examined offers is allowed.
- if the last offer is inspected it should be selected whatever its rank might be for both players.

It shall be noted that the stopping rule is firstly managed by player I. Hence, if player I decides to stop, player II will stop sampling with the offer selected by player I. So, the utility of player I should be conditioned by player II's possible consequences. In the next section, we propose a utility for each player that takes into account the opponent's possible positions for the selected offer. Then, we develop the dynamic formulation of the BOSP regarding this utility and the above problem assumptions.

B. Game formulation

We propose for the above definition of the BOSP a utility function for each player p ($p = 1, 2$) that takes into account the position of the opponent regarding the selected offer:

$$\begin{array}{cccc} & & \multicolumn{2}{c}{\text{Player II}} \\ & & \textit{Best} & \overline{\textit{Best}} \\ \cline{3-4} & \textit{Best} & (1,1) & (\alpha, \beta) \\ \text{Player I} & & & \\ & \overline{\textit{Best}} & (\beta, \alpha) & (\text{-}1,\text{-}1) \\ \cline{3-4} \end{array} \qquad (6)$$

Where $0 \le \alpha \le 1$ and $-1 \le \beta \le 0$.

The above payoff matrix provides the gain of each player when selecting an offer. If the selected offer is the best for both players, then each player is rewarded by 1. This payoff is expressed by the couple (1,1) in the matrix. If the selected offer is the best according to one player and not according to the other player, then first player is rewarded by α and the second player is rewarded by β. Finally, if the selected offer is not the best according to both players, then each player is rewarded by -1.

Regarding the above payoff matrix (6), each player is interested by the best offer according to his evaluation criterion. So, if the relative rank of the current offer is 1, the corresponding player can compare his expected payoffs when stopping and continuing. However, if the relative rank exceeds 1, then the decision will be automatically to continue sampling. For instance, the utility of each player is expressed in terms of the absolute rank of the opponent i.e. whether it is the best or not. Accordingly, the backward recursive equations for each player must be developed for the following cases:

1. $k_p = 1$ and $k_{p'} = 1$
2. $k_p = 1$ and $k_{p'} > 1$

It should be noticed that the above two cases are to be considered only in the stopping decision of player p. On the other hand, if player p decides to continue sampling, the computation of his expected utility is independent of the relative ranks of the currently inspected offer. It is expressed in terms of the maximal expected rewards in subsequent stages.

We develop in what follows the optimality equations for the general form of utility proposed formerly, then study the case where $\alpha = 1$ and $\beta = -1$. We derive the optimal strategy for this special case using a direct algebraic approach and a markov chain approach.

1) The general case

We consider the utility proposed in the previous section when $\alpha \in [0,1]$ and $\beta \in [-1,0]$. The optimality equation for player p ($p = 1,2$), at stage i ($i = 1,\dots,n-1$), is the following:

$$Ev^p(i,k_p,k_{p'}) = \begin{cases} max\{Ev_s^p(i,1,k_{p'}), Ev_c^p(i)\} & \text{if } k_p = 1 \\ Ev_c^p(i) & \text{Elsewhere} \end{cases} \tag{7}$$

Equation (7) expresses the decision of player p at stage i in terms of his relative rank and that of his opponent. If his relative rank of the selected offer is 1, then he should compare his expected utilities when stopping and continuing the selection process. However, if his relative rank for the selected offer exceeds 1, he will be rewarded by a negative payoff. So, a rational decision is to continue sampling by following the optimal strategy. Therefore, if the relative rank of the ith offer exceeds 1, the corresponding player should automatically continue sampling.

We analyze the expected payoff of player p when he decides to stop sampling by selecting the ith offer in each of the above cases, then derive its general form.

a) $k_p = 1$ and $k_{p'} = 1$. Since the utilities of the players are expressed in terms of absolute ranks and during the selection process only relative ranks are available, then we should compute the conditional probability that the ith offer has an absolute rank 1 knowing that its relative rank at the ith draw is 1 as below:

$$P\{(n,1)/(i,1)\} = \frac{P\{(n,1) \text{ and } (i,1)\}}{P\{(i,1)\}} = \frac{\frac{1}{n}}{\frac{1}{i}} = \frac{i}{n} \tag{8}$$

Where $P\{(i,j)\}$ denotes the probability that the current offer is ranked j among the i first offers.

Consequently, if the relative rank of the ith offer at the ith stage is 1 for player p, then its absolute rank is $\begin{cases} 1 & \text{with probability } \frac{i}{n} \\ > 1 & \text{with probability } 1 - \frac{i}{n} \end{cases}$

If player p decides to stop sampling by selecting the ith offer with relative rank 1, he should consider his opponent's relative rank $k_{p'}$ that can be either 1 or > 1. So, the expected utility of the corresponding player should take into account the stage number i, the relative rank 1 of the corresponding player and that of his opponent. Consequently, if for both players the relative rank of the selected offer is 1, then:

$$\begin{aligned} Ev_s^p(i,1,1) = \; & \tfrac{i}{n} \times \tfrac{i}{n} \times 1 + \tfrac{i}{n} \times (1 - \tfrac{i}{n}) \times \alpha \\ & + (1 - \tfrac{i}{n}) \times \tfrac{i}{n} \times \beta \\ & + (1 - \tfrac{i}{n}) \times (1 - \tfrac{i}{n}) \times (-1) \end{aligned} \tag{9}$$

The expectation of equation (9) is detailed in table I through the possible events that might occur. B means best and $\overline{B}$ means not the best. The first component in the couple (.,.) belongs to the evaluation of the current offer according to player p and the second component belongs to that of the opponent.

E	(B,B)	$(B,\overline{B})$	$(\overline{B},B)$	$(\overline{B},\overline{B})$
$P(E)$	$\frac{i}{n} \times \frac{i}{n}$	$\frac{i}{n} \times (1 - \frac{i}{n})$	$(1 - \frac{i}{n}) \times \frac{i}{n}$	$(1 - \frac{i}{n})^2$
Payoff	1	α	β	(-1)

TABLE I

EXPECTED PAYOFF OF PLAYER p WHEN THE RELATIVE RANK OF THE iTH OFFER FOR EACH PLAYER IS 1

b) $k_p = 1$ and $k_{p'} = 1$. When the relative rank of player p the relative rank is 1 and the opponent's relative rank exceeds 1, then the computation of the expected utility when stopping for player p requires the finding of the probabilities that the current offer is best or not for to each player. For player p the probability that the current offer has an absolute rank 1 given that its relative rank is 1 at the ith stage is $\frac{i}{n}$ and its probability to be not the best is $1 - \frac{i}{n}$ as previously shown.

For player p', the opponent of player p, it is clear that if the relative rank of the current offer exceeds 1, then the probability that its absolute rank is 1, is 0. In fact, the relative rank can only increase from stage to stage or remains the same. Indeed, if the relative rank of the current offer exceeds 1, its absolute rank exceeds 1 with probability 1. Then, if the relative rank of the ith offer is 1 at the ith stage, then its absolute rank is

$$\begin{cases} 1 & \text{with probability } 0 \\ > 1 & \text{with probability } 1 \end{cases}$$

If for player p stops the sampling process at the ith stage by selecting the ith offer, given that his relative rank for the selected offer is 1 and that of his opponent exceeds 1 is the following:

$$\begin{aligned} Ev_s^p(i,1,k) = & \tfrac{i}{n} \times 0 \times 1 + \tfrac{i}{n} \times 1 \times \alpha \\ & +(1-\tfrac{i}{n}) \times 0 \times \beta \\ & +(1-\tfrac{i}{n}) \times 1 \times (-1) \end{aligned} \tag{10}$$

Equation (10) expresses the expected payoff of player p if he decides to stop sampling by accepting the ith offer. The possible cases that might occur for player p when selecting the current offer are summarized in table II.

E	(B,B)	$(B,\overline{B})$	$(\overline{B},B)$	$(\overline{B},\overline{B})$
$P(E)$	$\frac{i}{n} \times 0$	$\frac{i}{n} \times 1$	$(1-\frac{i}{n}) \times 0$	$(1-\frac{i}{n}) \times 1$
Payoff	1	α	β	(-1)

TABLE II

EXPECTED PAYOFF WHEN THE RELATIVE RANK FOR PLAYER p IS 1 AND THAT OF THE OPPONENT'S EXCEEDS 1

Hence, the general form of the expected utility of player p if he decides to stop sampling by selection the best offer is the following:

$$Ev_s^p(i,1,k_{p'}) = \begin{cases} 2\frac{i}{n} - 1 + \frac{i}{n}(1-\frac{i}{n})(\alpha+\beta) & \text{if } k_{p'} = 1 \\ \frac{i}{n}(\alpha+1) - 1 & \text{if } k_{p'} > 1 \end{cases} \tag{11}$$

The second option available for player p is to discard the current offer and postpone the decision to the next stage. In this case, the relative rank of the next offer at the next stage will vary from 1 to $i+1$ with equal probabilities $\frac{1}{i+1}$. Then his expected payoff will be:

$$Ev_c^p(i) = \frac{1}{i+1} \sum_{j=1}^{i+1} Ev^p(i+1,j) \tag{12}$$

In the next section we rewrite the backward recursive equations for player p when $\alpha = 1$ and $\beta = -1$. Then, we derive the optimal strategy of the dynamic process using alternative approaches.

2) The case $\alpha = 1$ and $\beta = -1$:

If we rewrite the payoff matrix (6), we see that player p is rewarded by 1 if he selects the best offer and the selected offer is the best or not the best for player p'. Indeed, he's rewarded by (-1) if he does not select the best offer and the selected offer is the best or not the best according to player p'. Equivalently, we can say that his payoff becomes independent from the choice of his opponent. We shall rewrite the utility of player p as follows:

$$\begin{cases} 1 & \text{if the selected offer is the best for player } p \\ \text{-1} & \text{if the selected offer is not the best for player } p \end{cases}$$

Accordingly, the optimality equations of player p can be expressed in this special case only in terms of the stage number and the relative rank of the currently inspected offer regardless that of the opponent. In the remaining of this section we shall adopt two alternative approaches to show the same optimal strategy for player p. We firstly use a direct algebraic approach to develop the backward recursive equations and derive the corresponding optimal strategy.

Then, we propose an alternative representation of the problem using the markov chain approach.

At stage i, the current offer can have a relative rank varying from 1 to i. If the relative rank of the ith offer is 1, then player p should compare the expected probabilities when accepting and refusing the ith offer. However, if the relative rank of the current offer belongs to the set $\{2,\ldots,n\}$ then the overall decision should be to refuse the current offer. In this case, the backward recursive equation of player p is:

$$Ev^p(i,k_p) = \begin{cases} max\{Ev_s^p(i,1), Ev_c^p(i)\} & \text{if } k_p = 1 \\ Ev_c^p(i) & \text{Elsewhere} \end{cases} \tag{13}$$

Where

$$Ev_s^p(i,1) = 2\frac{i}{n} - 1 \tag{14}$$

since $\alpha + \beta = 0$ and $\alpha + 1 = 2$. We notice that in the stopping decision, player p obtains the same utility whether the selected offer is the best or not for player p'. If player p decides to postpone the decision to the next stage according to the optimal strategy, then:

$$Ev_c^p(i) = \frac{1}{i+1} \sum_{j=1}^{i+1} Ev^p(i+1,j) \tag{15}$$

Following Lindley [1961] for his proof of the basic version of the OSP with a single DM and Sakaguchi [1984] in which each player tries to choose his own offer, we propose a mathematical proof to derive the optimal strategy of BOSP for each player p ($p = 1,2$).

At stage i ($i = 1,\ldots,n-1$), the relative rank k_p of player p can be:

- $k_p = 1$:

$Ev^p(i,1) = max\{2\frac{i}{n} - 1, \frac{1}{i+1}\sum_{j=1}^{i+1} Ev^p(i+1,j)\}$

- $k_p > 1$: the decision should be postponed to the next stage, then:

$$\begin{aligned} Ev^p(i,k_p) = \; & Ev_c^p(i) \\ & = \tfrac{1}{i+1} Ev^p(i+1,1) + \tfrac{i}{i+1} Ev_c^p(i+1) \\ & = \tfrac{1}{i+1}[Ev^p(i+1,1) + iEv_c^p(i+1)] \end{aligned} \tag{16}$$

Let us consider the acceptance area of player p. If the ith offer is accepted then $\forall i^{'} > i$ with relative rank 1 is also accepted. Accordingly,

$$Ev^p(i^{'},1) = 2\frac{i^{'}}{n} - 1 \quad \forall i^{'} \geq i \tag{17}$$

So, from equations (16) and (17):

$$\begin{aligned} Ev_c^p(i) = \; & \tfrac{1}{i+1}[2\tfrac{i+1}{n} - 1 + iEv_c^p(i+1)]\tfrac{2}{n} \\ & -\tfrac{1}{i+1} + \tfrac{i}{i+1}Ev_c^p(i+1) \end{aligned} \tag{18}$$

Hence,

$$Ev_c^p(i) = \frac{2}{n} + \frac{1}{i+1}[Ev_c^p(i+1)i - 1] \tag{19}$$

Let $d^p(i) = \frac{Ev_c^p(i)}{i}$
We can write,

$$d^p(i) = \frac{2}{ni} + d^p(i+1) - \frac{1}{i(i+1)} \tag{20}$$

$$d^p(i+1) = \tfrac{2}{n(i+1)} - \tfrac{1}{(i+1)(i+2)} + d^p(i+2)$$

$$\vdots$$

$$d^p(n-1) = \tfrac{2}{n(n-1)} - \tfrac{1}{(n-1)n} + d^p(n)$$

From the above equations:

$$d^p(i) = \frac{2}{n}\sum_{j=i}^{n-1}\frac{1}{j} + \frac{1}{n} - \frac{1}{i} \tag{21}$$

Player p accepts the ith offer when his expected utility when accepting the ith offer exceeds his expected utility when discarding it. Hence, the ith offer is accepted if it fulfills the following condition:

$$2\frac{i}{n} - 1 \geq i(\frac{2}{n}\sum_{j=i}^{n-1}\frac{1}{j} + \frac{1}{n} - \frac{1}{i}) \tag{22}$$

Let $\psi_i = \sum_{j=i}^{n-1}\frac{1}{j}$. Equation (22) can be written as: $\psi_i \leq \frac{1}{2}$

The optimal strategy of player p is to reject the first $i^* - 1$ such that $\psi_i < \frac{1}{2} \leq \psi_{t-1}$, and to select the first offer with relative rank 1 thereafter.

V. CONCLUSION

We proposed in this paper a dynamic formulation for the BOSP. Our definition of the problem assumes that the two players are required to select the same offer. Indeed, each player is allowed to stop the selection process by accepting an offer. The proposed utility shows that each player aggregates the two utilities differently. We formulated the backward recursive equations for the general utility and derived the optimal strategy for a special case of the BOSP.
An alternative formulation of the BOSP can be of interest when an offer is selected only if it is accepted by both players. Further research includes solving this BOSP and testing it on several problem sizes.

REFERENCES

[1] Albright S. C. (1977), *A baysian approach to a generalized house selling problem*, Management Science, vol 21, no 1, pp. 60-67.

[2] Bruss F. T. (1987), *On an optimal selection problem of Cowan and Zabczyk*, Journal of Applied Probability, vol 24, pp. 918-928.

[3] Bruss F. T. and Ferguson T. S. (1993), *Minimizing the expected rank with full information*, Journal of Applied Probability, vol 30, pp. 616-626.

[4] Chow Y. S., Moriguttì S., Robbins H. and Samuels S. M. (1964), *Optimal selection based on relative rank (The "secretary" problem)*, Israel Journal of Mathematics, vol 2, pp. 81-90.

[5] "Cowan R. and Zabczyk", *An optimal selection problem associated with the poisson process*, Theory of Probability and its Applications, vol XXIII, no 3, pp. 584-592, 1978.

[6] "DeGroot M. H." (1968), *Some problems of optimal stopping*, Journal R Statist Soc, vol B30, pp. 108-122.

[7] Dynkin E. B. (1963), *The optimum choice of the instant for stopping a markov process*, Soviet Mathematical Dokl., vol 4, pp. 627-629.

[8] Enns E. G. and Ferenstein E. (1985), *The horse game*, Journal of the Operations Research Society of Japan, vol 28, no 1, pp. 51-62.

[9] Freeman P. R. (1983), *The secretary problem and its extensions: A review*, International Statistical Review, vol 51, pp. 189-206.

[10] Gilbert J. and Mosteller F. (1966), *Recognizing the maximum of a sequence*, American statistical Association Journal, vol 61, pp. 35-73.

[11] Gnedin A. V. (1996), *On the full information best-choice problem*, Journal of Applied Probability, vol 33, pp. 678-687.

[12] Gusein-Zade S. M. (1966), *The problem of choice and the optimal stopping rule for a sequence of independent trials*, Theory of Probability and its Applications, vol 11, pp. 472-476.

[13] Lindley D. V. (1961), *Dynamic programming and decision theory*, Applied statistics, vol 10, pp. 39-51.

[14] Nakai T. (1986), *An optimal selection problem for a sequence with a random number of applicants per period*, Operations Research, vol 14, no 3, pp. 478-485.

[15] Presman E. L. and Sonin I. M. (1972), *The best choice problem for a random number of objects*, Theory of Probability and its Applications, vol 17, no 4, pp. 657-668.

[16] Rose J. S. (1984), *Twenty years of secretary problems: A survey of developments in the theory of optimal choice*, Advances in management studies, vol 1, no 1, 53-64.

[17] Sakaguchi M. (1978), *A bilateral sequential game for sums of bivariate random variables*, Journal of the Operations Research Society of Japan, vol 21, no 4, pp. 486-507.

[18] Sakaguchi M. (1984), *Bilateral sequential games related to the no-information secretary problem*, Mathematica Japonica, vol 29, no 6, pp. 961-973.

[19] Sakaguchi M. (1985a), *The bilateral secretary problem with a random population size*, Mathematica Japonica, vol 30, no 3, pp. 457-469.

[20] Sakaguchi M. (1985b), *Non-zero-sum games for some generalized secretary problems*, Mathematica Japonica, vol 30, no 4, pp. 585-603.

[21] Salminen P, Teich J. E. and Wallenius J. (1996), *A decision model for the multiple criteria group secretary problem: Theoretical considerations*, Journal of the Operational Research Society, vol 47,no 3, pp. 1046-1053.

[22] Smith M. H. (1975), *A secretary problem with uncertain employment*, Journal of Applied Probability, vol 12, pp. 620-624.

[23] Tamaki M. (1991), *A secretary problem with uncertain employment and best choice of available candidates*, Operations Research, vol 39, no 2, pp. 274-284.

Human Centered Interface at IT Revolution

Toru Yamaguchi
PRESTO, Japan Science and
Technology Corporation (JST)
Department of Electronic Systems
Engineering, Tokyo Metropolitan
Institute of Technology
Asahigaoka 6-6, Hino City, Tokyo
Japan
Email : yamachan@fml.ec.tmit.ac.jp

Fumio Harashima
President of Tokyo Metropolitan
Institute of Technology
Asahigaoka 6-6, Hino City, Tokyo
Japan
Email : f.harashima@ieee.org

Hiroki Murakami
Department of Electronic Systems
Engineering, Tokyo Metropolitan
Institute of Technology
Yamaguchi lab, Asahigaoka
6-6, Hino City, Tokyo
Japan
Email : muraka@fml.ec.tmit.ac.jp

Abstract—**The method which composes the intelligent agent system which recognizes human intention is shown in this paper. The ontology for human of each agent is build by the neural network in this collaborative agent system. Then, a difference of the intention expression by the individual is accepted by holding the ontology in common between the agents.**

Keywords—Ontology, Intelligent agent, Intelligent Interface

Ⅰ. INTRODUCTION

Recently, the maintenance of the information infrastructure is proceeding rapidly. Though, it has been a great problem for many people using information infrastructure, taking long time to get accustomed to keyboard and mouse.

Especially, Japan became high aged society so radically, which is unprecedent in the world. The aged have time and money for Internet. Therefore, they are being expected to become the biggest user group. A general interface to the user in place of the mouse and the keyboard is indispensable for the people who aren't accustomed to use the computer such as the aged and infants. It is considered to use human movement and voice as a Natural Interface.

The method of intention recognition is indicated in this paper. By using movement of hands and face, an intelligent agent builds the ontology for intention.

And, as the way to accept different expression by the individual in the intention recognition, the technique of holding common ontology between the agents is shown.

Ⅱ. THE CONCRUTION OF THE AGENT SYSTEM

The composition of the system is shown in Fig.1. This is based on the class model of Rasmussen. Each agent gets movement of face and hands from the picture. And then, it gets an instance from the pattern of an inputted data. The ontology which coped with human intention is made by using Conceptual Fuzzy Set (CFS) composed by Bidirecitional Associative Memory (BAM). Furthermore, build the system for each agent to hold it's ontology in common.

Ⅲ.INPUT OF INFORMATION

The input information for intention recognition is the human movement. Especially, for input information, the time series data of the face position, face direction and hand position are used.

To detect the movement, first, CCD camera is used to get the color information from the picture. Second, find the position of hand, face and lip from the picture. Third, calculate center gravity to specify each position. Tracking by using the color information is shown in Fig.2. Calculated coordinate value is measured as the data on time series in Fig.3. Then, the extreme value is decided as the characteristics point.

For each other part, two data of the X direction and the Y direction are held. By picking up to position of the lip and the face, it is possible to get the direction of the face. Moreover, it is normalized, that a relative position of the hand and the foot is measured with the position of the face. Therefore, it is possible to extract the character's movement of the person without getting closer to the camera.

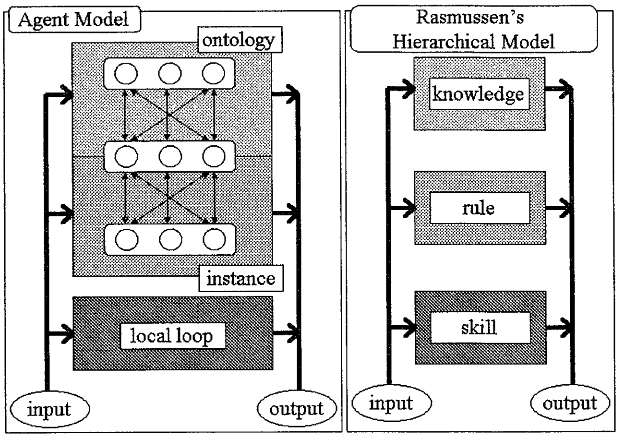


Fig. 1. Composition of system

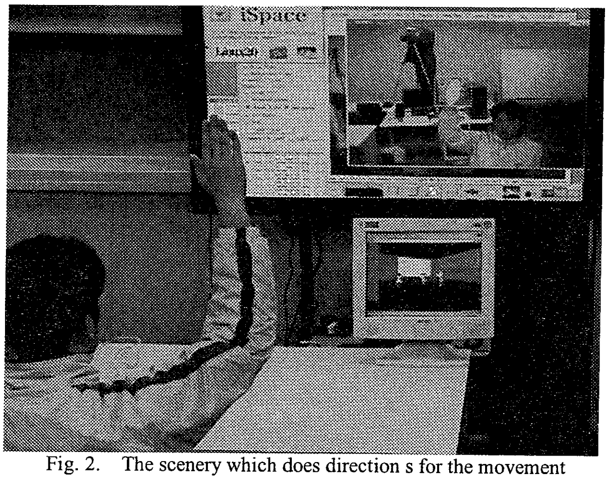


Fig. 2. The scenery which does direction s for the movement

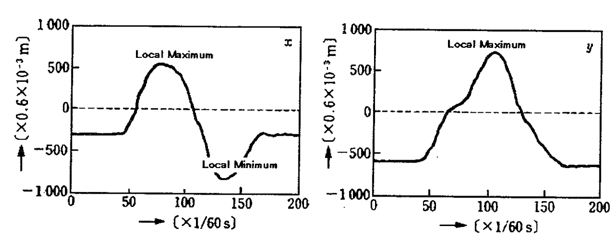


Fig. 3. Time series data

Ⅳ. INTENSION RECOGNITION USING ONTOROGY

As for the ontology, abstract general idea structure was defined. The ontology is expressed by the relations with more than one concrete case (instance).

Because speed and size aren't fixed for each person's movement, robust nature becomes necessary to reduce the unstable noise, for the intention recognition. The unstable noise is dissolved by using CFS composed by BAM. An ontology is implemented by this CFS. An easy case is shown in Fig.4.

A human movement is inputted as an instance. An ontology recognize when spread treatment is done in the network which composes CFS. Then, intention is recognized (Fig.5).

When this method was used for recognizing the sign language, the result of the recognition precision was 96%. Also, driver's intention recognition was 86%. As for the intention recognition by the ontology, it is understood that it is fully useful from these results.

Ⅴ. COOPERATIVE ONTOLOGY

When a certain agent does recognition and an ontology wasn't remembered, the joint ownership of the ontology is done by the method that uses CFS of other agents trying to recognize again. It is judged by the size of the

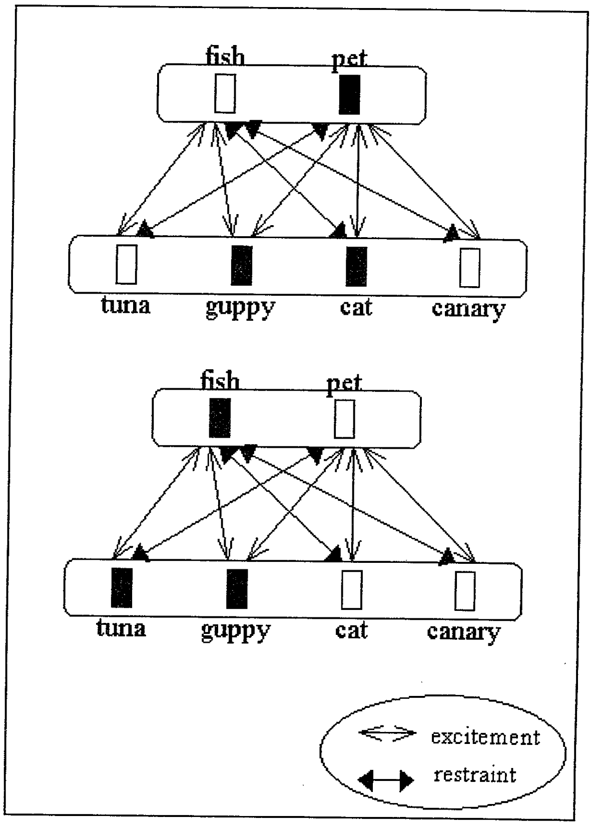


Fig. 4. Example of CFS

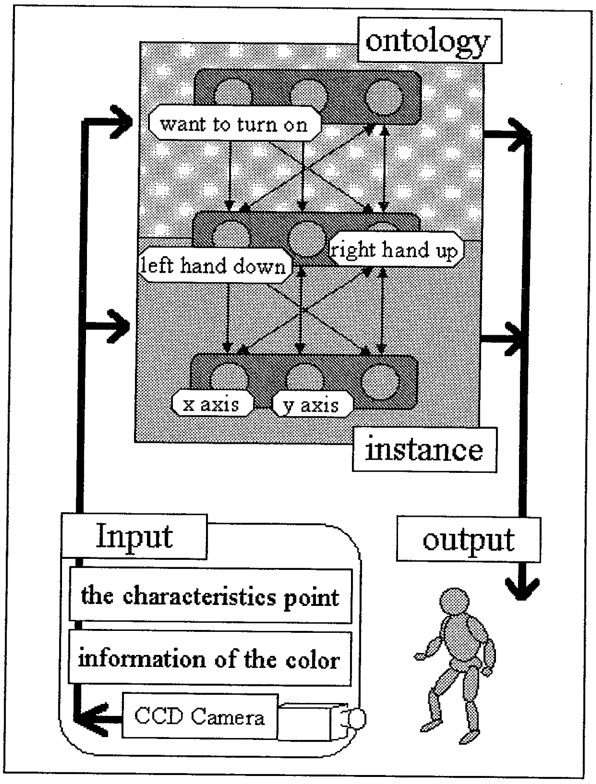


Fig. 5.

active value whether recognition is tried again in the same way as time to remember it. This is thought to be effective from controlling the degree of "fuzzy" by changing the number of times of the spread treatment (the number of the association steps) in the BAM.

A concrete process is shown by flow chart in Fig 6. A certain agent does association reasons by his CFS at first. If an active value is higher than the constant value, and ontology was remembered, intention was decided to be recognized. When active value is small, there is a possibility of error by the difference in the intention expression. Therefore, an active value is looked for by CFS of other agents, and a judgment is done in the method which is the same as former times.

In case that even active value of certain agent is small, the others get big active value existent. Ontology uses bigger value to edit and get new information against learning. Moreover, it is considered making a new ontology, too. When an instance was remembered with more than one agent from the active value of the ontology, the association reasoning is done by using both directions of BAM in a certain instance which could get the same active value.

To show it in Fig.7, human support robot shows validity with the human being by the collaboration of the intellectual agent.

Ⅵ. EXPERIMENT RESULT

A. HOW TO MAKE AN EXPERIMENT

An interface was made a human hand, and intention recognition was done in last experiment based on the position coordinate of the hand of the human being inputted from the CCD camera.

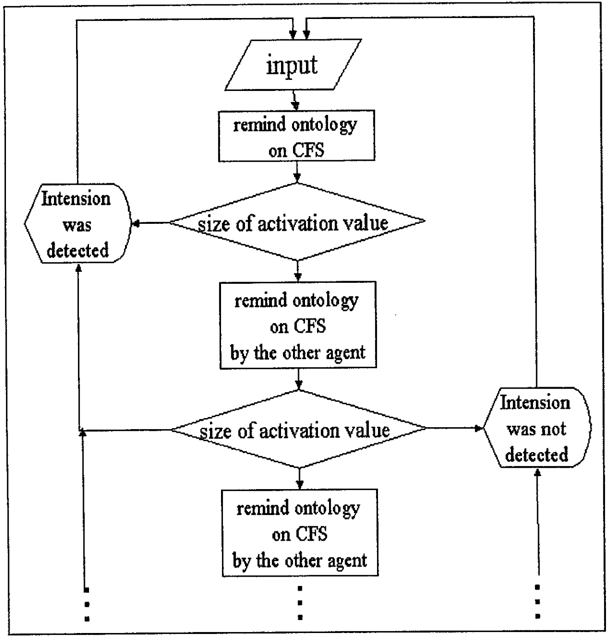


Fig.6. The flow of the information joint ownership

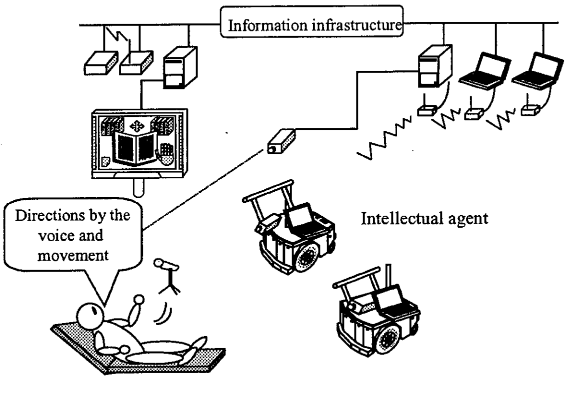


Fig.7.

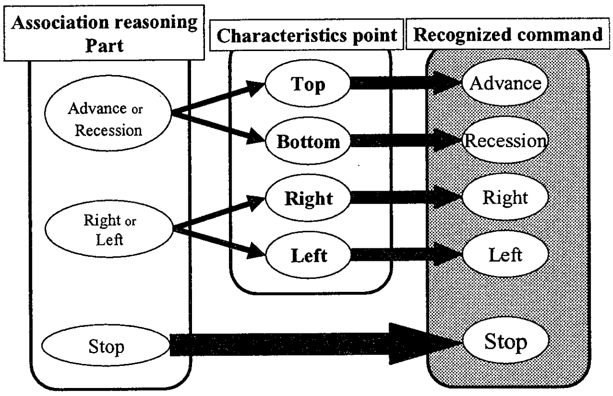


Fig. 8. Command recognition part

It was done by A, B two testes.

A line says both intention recognition to the agent who has the rule of A first.

Next, intention recognitions is done to be the same toward the agent of A made to share information.

A movement recognition part judges the command of a human movement which is finally from the reasoning result of the coordinate data. The first characteristics point from the movement input part, the association reasoning part, face of a client, and the order which corresponded with the command is sent.

The method of the judgments is shown in Fig .8.

It is limited by three of the forward recession, turning right, turning left or the stopping from the conclusion of the association reasoning part first.

A stop is judged by the command where to be stopped.

When it turns at the forward recession, turning right and left, the coordinate date of the characteristics point of the beginning which the data from the characteristics point extraction next are used.

It judges an advanced or recession whether to be in the bottom if that characteristics point is in the top in comparison with the position of a hand to start a movement.

Fig.9. Experiment scenery

TABLE Ⅰ
THE RECOGNITION RATE OF A IN THE AGENT OF A

Advance	Recession	Right	Left	Stop
86%	87%	95%	96%	99%

TABLE Ⅱ
THE RECOGNITION RATE OF B IN THE AGENT OF B

Advance	Recession	Right	Left	Stop
88%	86%	92%	96%	96%

A characteristics point is in the right, or in the left, or turning left, turning right are judged in the same way.

It was decided that output in this experiment moved the CG space. Experiment scenery is shown in Fig.9, Fig.10.

B. EXPERIMENT RESULT

An intention recognition rate is shown in Table Ⅰ, Table Ⅱ in this experiment.

The rule of the stop in A, B is different.

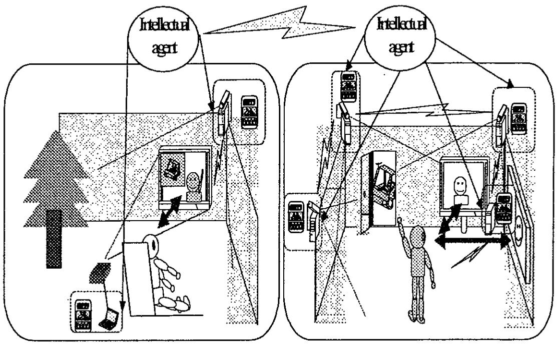
When the room of A which shows in Fig.9, was replaced.

1. The average of the recognition rate when information joint ownership isn't is 82%.
2. The average of the recognition rate after joint ownership was done was 92%.

What we knows from the upper result is probably the point whose recognition rate of the advance and the recession is bad first.

It is raised that the influence of the light is taken with a command's coming off as a reason from the human fundamental movement.

A rule except for the stop of A, B was about the same in last experiment. An experiment was done by the setup that the rule of the stop was wrong.



room A room B

Fig.10. Information joint ownership

An agent could be able to do the intention recognition of two users at the recognition rate which was same by doing joint ownership.

Ⅶ. CONCLUSION

When an agent recognizes human intention, it was shown how to hold to an ontology by the Conceptual Fuzzy Set in common as one technique that the robust enhanced in this paper.

This system is used in the intelligent transport system and welfare robot is applied to the use that a human being is exactly supported.

From now on, the validity of the system will be shown in the concrete conditions, and furthermore building multi modal interface by combining it with the voice and so on will be considered.

References

[1] T.Takagi, A.Imura, H.Ushida, T.Yamaguchi, "Conceptual Fuzzy Sets as a Meaning Representation and their Inductive Construction" *International Journal of Intelligent Systems*, Vol.10, 929-45, 1995.

[2] T.Takagi, A.Imura. H.Ushida, T.Yamaguchi, "Multi Layered Reasoning by Means of Conceptual Fuzzy Sets" *International Journal of Intelligent Systems*, Vol.11, 97-111, 1996.

[3] T.Tanihashi, M.Hagiwara, T.Yamaguchi, "Neural Network and Fuzzy Signal Treatment" *Corona inc*, 1998.

[4] T.Takagi, T.Yamaguchi, M.Sato, "Multi-Modal Information Integration by Conceptual Fuzzy Set for Interactive System" *1998 IEEE World Congress on Computational Intelligence*, 738-743, 1998.

[5] T.Takagi, S.Kasuya, M.Mukaidono, T.Yamaguchi, T.Kokubo, "Realization of Sound-scape Agent by the Fusion of Conceptual Fuzzy Sets and Ontology" *8th International Conference on Fuzzy Systems FUZZ-IEE'99, Ⅱ*, 801-806, 1999

[6] T.Takagi, S.Kasuya, M.Mukaidono, T.Yamaguchi, "Conceptual Matching and its Application to Selection of TV Programs and BGMs", *IEEE International Conference on Systems, Man and Cybernetics SMC'99, Ⅲ*, 269-273,*1999*.

[7] T.Takagi, K,Kawase, K.Otsuka, T.Yamaguchi, "Data retieval Using Conceptual Fuzzy Sets" *9th International Conference on Fuzzy Systems FUZZ-IEEE 2000, Ⅱ*, 94-99, 2000.

Conventional and Non Conventional Models of the Handwriting Process: Differential, Neural and Neuro-Fuzzy Approaches

M. Benrejeb
LARA, Ecole Nationale
d'Ingénieurs de Tunis,
BP 37, Belvédère 1002,
Tunis, Tunisie.
mohamed.benrejeb@enit.rnu.tn

M. Sano
Faculty of Information
Sciences,
Hiroshima City University,
Hiroshima, Japan.
syaka@cs.hiroshima-cu.ac.jp

A. El Abed-Abdelkrim
LARA, Ecole Nationale
d'Ingénieurs de Tunis,
BP 37, Belvédère 1002,
Tunis, Tunisie.
afef_abdelkrim@yahoo.fr

Abstract -- **A survey of the handwriting process, seen from different angles, is given in this paper. First, the handwriting process is analysed as a physical process represented by differential equations. Second, it is modelled by an artificial neural network. These results make possible the integration of a human expertise to control this process by fuzzy logic controllers. Finally, the handwriting process is considered as a neuro-fuzzy system and a new model is proposed to represent it.**

Keywords -- Handwriting process, Differential model, Neural model, Neuro-fuzzy model.

I. INTRODUCTION

THE handwriting process is one of the most important biological process of the human body. In fact, handwriting movements, not only characterise people and could explain their attitude but also play a preponderant role in communication. That's why it was studied in many surveys and researches, [1]. In 1962, Van der Gon proposed a first order linear system [2]. In 1975, Yasuhara added the friction of the pen and surface to the previous model, [3]. In 1987, Edelman and Flash, [4], proposed a model based on the trajectories of the hand.

In this paper, a neural network model of the handwriting process is proposed then an integration of the human expertise is added to control the process with fuzzy logic controllers. Finally a new neuro-fuzzy model of the handwriting process is developed. The simulation results are applied to the cursive Arabic letters and Piece of arabic Word (PAW).

II. THE HANDWRITING PROCESS AS A DIFFERENTIAL SYSTEM

Several models of the handwriting process were proposed in the litterature. The first one, proposed by Van Der Gon, is refined by Yasuhara and leads to a new model which is described by the following differential system :

$$\begin{cases} \ddot{x}(t) = F_x(t) - (4.7 + 0.5\dfrac{1.5}{(\dot{x}(t)^2 + \dot{y}(t)^2)^{1/2}})\dot{x}(t) \\ \ddot{y}(t) = F_y(t) - (4.7 + 0.5\dfrac{1.5}{(\dot{x}(t)^2 + \dot{y}(t)^2)^{1/2}})\dot{y}(t) \end{cases} \qquad (1)$$

For the inputs F_x and F_y, figures 1 and 2, corresponding to elementary motions of the hand to write the arabic letter "ﻌ", the obtained output, figure 3, is too different than the expected written regularly one.

III. THE HANDWRITING PROCESS AS AN ARTIFICIAL NEURAL NETWORK

In this part, an artificial neural network (ANN) is developed to study and modelise the behaviour of the handwriting process, figure 4.

The neural modelling approach, [5,6,7], consists of training a neural network to identify a discrete model given by (2) :

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} = \Phi(Fx_k, Fy_k, x_{k-1}, x_{k-2}, y_{k-1}, y_{k-2}) \qquad (2)$$

with : $\Phi : \Re^6 \to \Re^2$.

Sample values of x_k, y_k, Fx_k and Fy_k are taken by simulation of the model (1) so as to constitute a set of patterns and targets to the learning of the neural network. We create a three-layer network with 100 hidden nodes, [8]. The hidden layer activation function is hyperbolic tangent sigmoïd function and the output one is linear.

To generate an estimation of muscular stimuli applied to the hand, an indirect ANN is proposed. The inverse modelling consists of training a neural network to identify a discrete model given by (3) :

$$\begin{pmatrix} Fx_k \\ Fy_k \end{pmatrix} = \Psi(x_k, x_{k-1}, x_{k-2}, y_k, y_{k-1}, y_{k-2}) \qquad (3)$$

with : $\Psi : \Re^6 \to \Re^2$

The comparison between the muscular efforts generated by the inverse network and those used for the direct network shows great similarities, [9], figure 5.

IV. INTEGRATION OF THE HUMAN EXPERTISE TO CONTROL THE HANDWRITING PROCESS

A. First proposed approach

To integrate a human expertise to control the handwriting process, two Mamdani fuzzy logic controllers of the hand physical model, [10], figure 6, are developped. The used membership fonctions are given by the figures 7 and 8. The two rule tables 1 and 2 are tested, [11]. Simulation results done for writing the arabic letter "ع", figures 9 and 10, are compared to those of the physical model controlled with a theorical PID controller, figure 11.

B. Second proposed approach

Iguider and Yasuhara showed that the shapes of characters are coded only in time, [12], i.e. by the duration and not by the magnitude of the applied muscle force.

Given $\Delta x = x_{i+1} - x_i$ and $\Delta y = y_{i+1} - y_i$ with $x_i = x(t_i)$ and $y_i = y(t_i)$, the positions at t_i of the pencil point in the x and y axis repectively.

The signs of Δx and Δy give an idea about the sens and direction of the pen when it is writing :

- If $\Delta x > 0$ then $F_x = 1$.
- If $\Delta x = 0$ then $F_x = 0$.
- If $\Delta x < 0$ then $F_x = -1$.
- If $\Delta y > 0$ then $F_y = 1$.
- If $\Delta y = 0$ then $F_y = 0$.
- If $\Delta y < 0$ then $F_y = -1$.

Another fuzzy logic controller (based on segments), [13], that consider this kind of control law is developed and the handwriting process for the Piece of Arabic Word (PAW) "محمد" is simulated.

The rule table is as follows :

If (Erreur) is (NG) then (Commande) is (N).

If (Erreur) is (EZ) then (Commande) is (Z).

If (Erreur) is (PG) then (Commande) is (P).

The simulation results are given by figures 12 and 13.

V. THE HANDWRITING PROCESS AS A NEURO FUZZY SYSTEM

Finally, a neuro-fuzzy model of the human handwriting process is proposed. Indeed, the developed direct neural network is controlled with the developed fuzzy logic controller (second approach) then the responses are simulated as seen in figure 14. The input of the developed model are given by the figure 12.

VI. CONCLUSION

The addition of handwriting process fuzzy logic controllers has shown a very interesting results. In fact, it corresponds to an integration of the human expertise, which is very important, in the control approach of the handwriting process. However, the neuro fuzzy model needs to be refined and some of its parameters must be adjusted so as to give better responses.

REFERENCES

[1] Bouslama F. and Benrejeb M., "Exploring the human handwriting process", *Int. J. Appl. Math. Comput. Sci.*, Vol. 10, N° 4, 2000, pages 877-904.

[2] Van Der Gon D., Thuring P. and Strackee J., "A handwriting similator" , *physiol. Med. Biol.* , Vol.6, 1962, pages 407-414.

[3] Yasuhara M., "Experimental studies of handwriting process", *Rep. Univ. Electro-comm. 25-2*, Japan, 1975, pages 233-254.

[4] Edelman S. and Flash T., "A model of handwriting", *Biological Cybernetics*, 57, Springer Verlag, 1987, pages 25-36.

[5] Ikonen E. and Kortela U., "Intelligent on-line modelling of non-linear processes", *Proc. of 3rd Europ. Cont. Conf.*, Rome, Italy, Sept. 1995, pages 2414 -2419.

[6] Liu Kadirkamanathan G.P. and Billings S.A., "Identification of nonlinear dynamical systems by growing RBF networks", *Proc. of 3rd Europ. Cont. Conf.*, Rome, Italie, Sept. 1995, pages 2408 -2413.

[7] Tanaka M., "Non linear system identification by the combination of self-organizing feature map and radial basis function network", *Proc. of 3rd Europ. Cont. Conf.*, Rome, Italy, Sept. 1995, pages 1580-1585.

[8] Abdelkrim A., Benrejeb M., Bouslama F. and Sano M., "Approches neuronales proposées pour l'étude du système d'écriture à la main", *JSFT'2000*, Monastir, Oct. 2000.

[9] Abdelkrim A., Benrejeb M. and Sano M., "PAW handwriting neural system", *ICCCP'01*, IEEE/IEE Conference , Oman, 2001, pages 207-211.

[10] Bühler H., *Réglage par logique floue*, Presses polytechniques et universitaires romandes, Lausanne.

[11] El Abed-Abdelkrim A., Gasmi M., Benrejeb M. and Sano M., "Intégration de l'expertise humaine dans la synthèse d'une commande floue d'un système d'écriture à la main", *JS'2001*, Ecole de l'Aviation de Borj El Amri, Tunisia, May 2001, pages 129-133.

[12] Iguider Y. and Yasuhara M., "Extracting control pulses of handwriting movement", *Trans. of the Soc. Inst. and Cont. Eng.*, Vol.31, N°8, Japan, 1995, pages 1175-1184.

[13] Sugeno M., "On stability of fuzzy systems expressed by fuzzy rules with singleton consequents", *IEEE Trans. On Fuzzy Systems*, 7(2), pages 201-223, 1999.

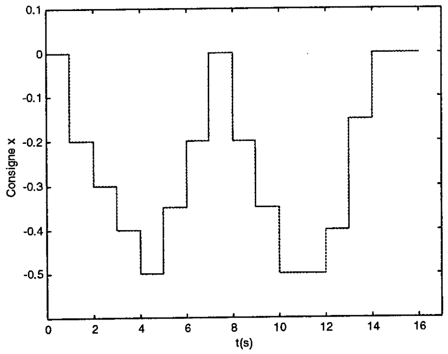


Figure 1. X position input to write the letter "ع"

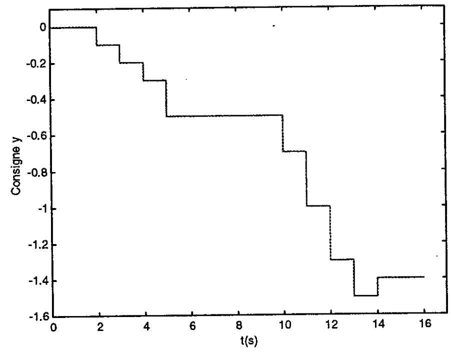


Figure 2. Y position input to write the letter "ع"

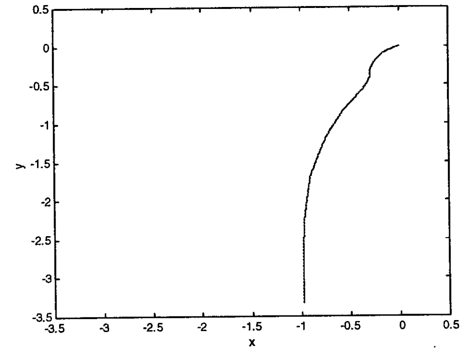


Figure 3. Letter "ع" written with the physical model of the hand-pen association.

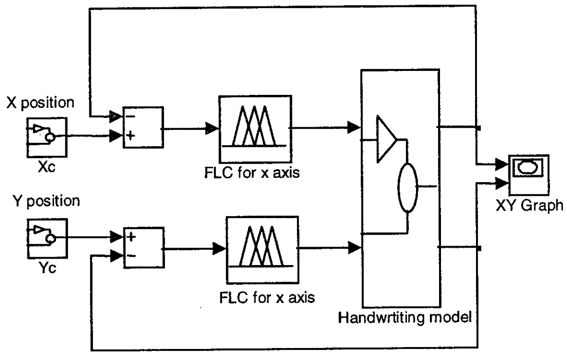


Figure 6. Handwriting system controlled by a fuzzy controller

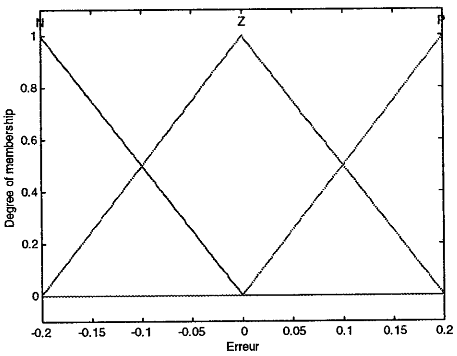


Figure 7. Membership fonctions of the error (Erreur) and the error variation (Var_Erreur)

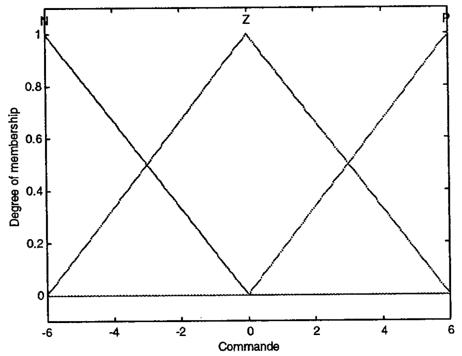
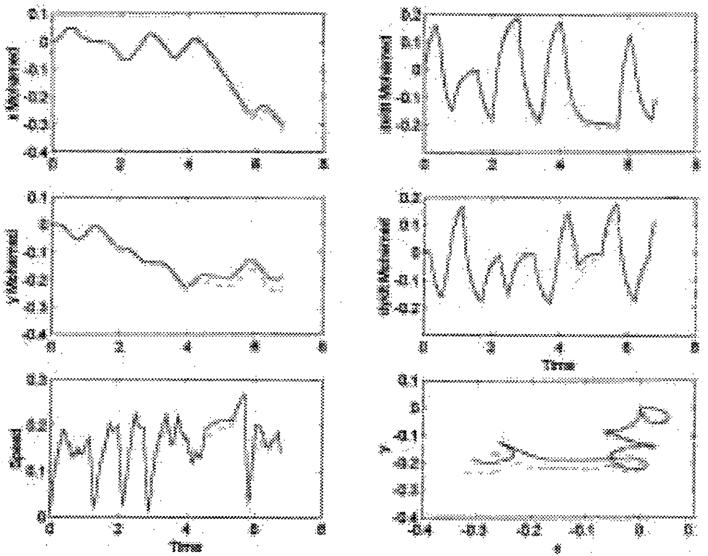


Figure 8. Membership fonctions of the (Commande)



(__) : The responses of the differential system (2).

(----) : The responses of the neural network.

Figure 4. Neural network and differential system responses.

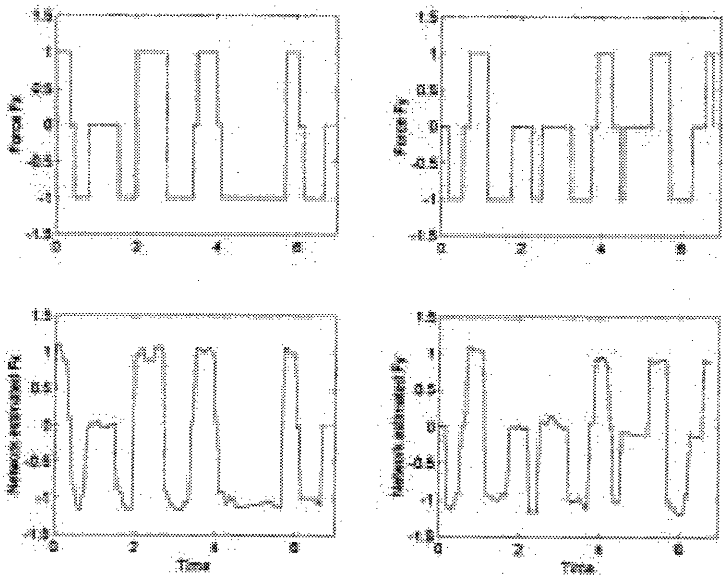


Figure 5. Control pulses used for direct NN and generated by INN

Table 1. Rule table 1

Erreur Var_Erreur	N	Z	P
N	N	N	Z
Z	N	Z	P
P	Z	P	P

Table 2. Rule table 2

Erreur Var_Erreur	N	Z	P
N		N	
Z	N	Z	P
P		P	

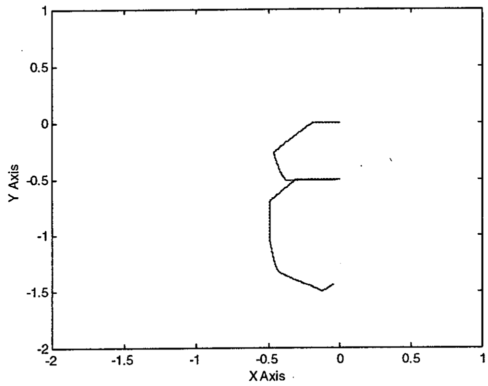


Figure 9. The letter "ع" written with the fuzzy logic controller of the rule table1.

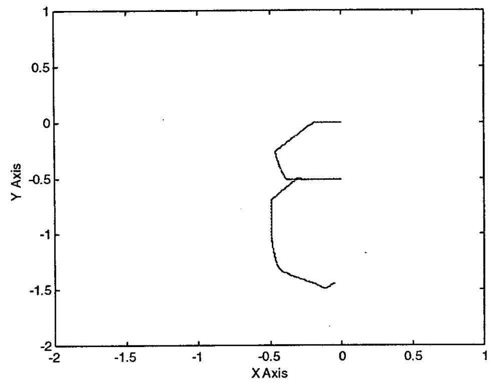


Figure 10. The letter "ع" written with the fuzzy logic controller of the rule table2.

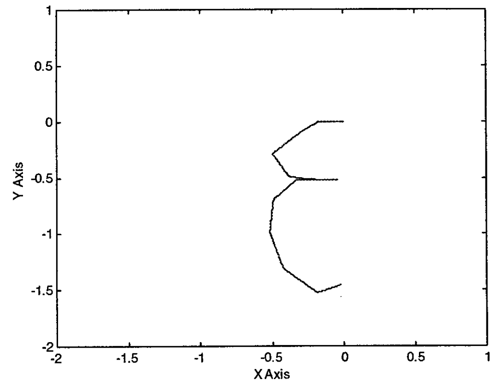


Figure 11. The letter "ع" written with a PID controller (K_P=10, K_I=1,K_D=2)

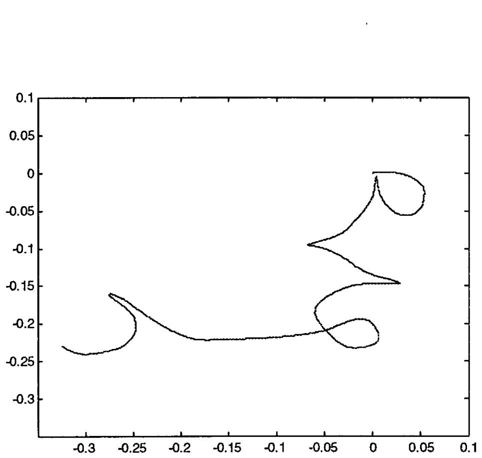


Figure 12. The input to the handwriting process controlled with the fuzzy logic controller

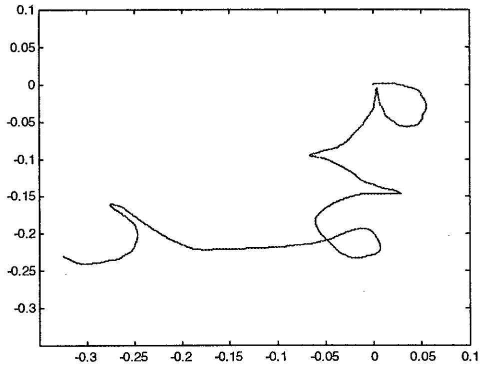


Figure 13. Response of the handwriting system controlled with the fuzzy logic controller

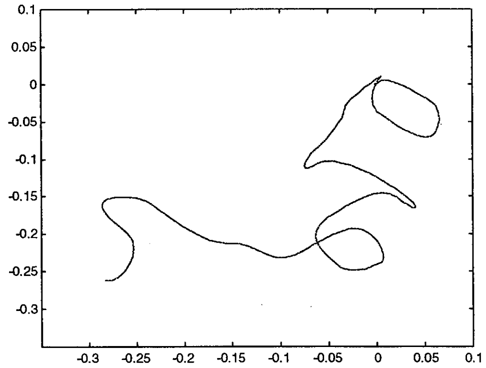


Figure 14. The response of the neural model of the handwriting process controlled with a fuzzy logic controller.

Adaptive and Predictive Controllers
Some real applications

Mekki Ksouri
INSAT, National Institute for
Applied Science and Technology
BP 676, 1080 Tunis Cedex
Tunisia
Mekki.Ksouri@isetr.rnu.tn

Ridha Ben Abdennour
ENIG, National School of
Engineers of Gabès
Tunisia
Ridha.Benabdennour@enig.rnu.tn

Faouzi M'Sahli
ENIG, National School of
Engineers of Gabès
Tunisia
Faouzi.msahli@isetkh.rnu.tn

Abstract-- **This paper gives a general presentation of some successful real applications of adaptive and predictive control theory. In each application we use numerical real time control via a personal computer, A/D and D/A converters.**

The first application has been done on a cement plant to improve the energy consumption. We showed that an adaptive control of the clinker gate cooler leads to an energy gain that exceeds 6%.

The second process considered is a semi batch reactor where esterification of crude acid of olive oil is performed. In this case we show that a constrained adaptive generalized predictive controller gives a perfect tracking of the set point and can highly reduce disturbance effects.

In the third process we have controlled the level of a liquid in a tank, it was shown that fuzzy supervision of generalized predictive controller improves the performances in terms of rapidity, of the degree of stability and of variances of the control and output signals.

Keywords—Adaptive controller, predictive controller, fuzzy supervision, GPC, Real time.

I. IDENTIFICATION AND ADAPTIVE CONTROL OF A CLINKER GRATE COOLER OF A CEMENT FACTORY

THE production of cement is very expensive in term of energy. In this work our main objective is to reduce the energy consumption by recuperation. The clinker grate cooler presented in figure 1 is used essentially to cool clinker from 1200 °C to ambient. It uses a system of perforated metallic plates.

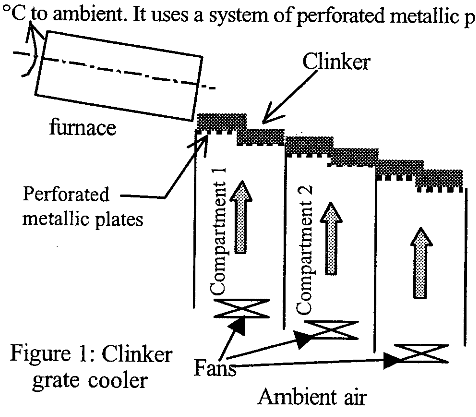


Figure 1: Clinker grate cooler

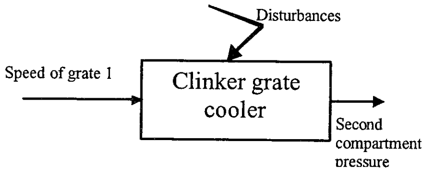


Figure 2 : Block diagram

The cooling operation is insured by six fans installed in compartments under the grate. The moving speed of plate number 1 is a very important control input of the flow rate of the secondary air combustion [1, 2]. It is very easy to see that for zero speed, the flow rate is zero, and for very high one the flow rate is very high and the cooling rate is very low. Classically, in the case of cement plant, a linear analog PID controller is used for a pressure control in the second compartment by the speed of the first grate. The block diagram of the single input single output system is illustrated in figure 2. We suggest a structural and parametric identification for the considered system. An ARMAX model is established and validated practically. The used analog PID controller is taken away and replaced by a numerical control. We elaborate an adaptive control of the real process. The chosen criteria are based on a self-tuning control strategy that minimizes both the output variance and the control energy [2]. The real application of the retained control strategy to a cement plant located near Gabès in the south of Tunisia has given good results and better performances compared to the analog PID controller.

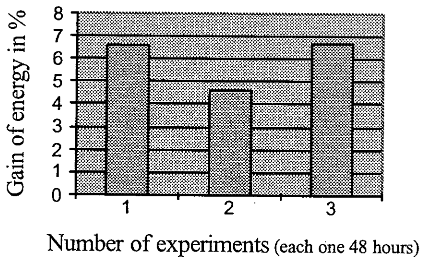


Figure 3 :Improvement of energy consumption

The values given by figure 3 are obtained for three real experiments on the cement plant; they show that adaptive control strategy brought in important gain energy relatively to a classical control.

II. AN ADAPTIVE PREDICTIVE CONTROLLER FOR A SEMI-BATCH REACTOR

Batch and semi-batch reactors are used for the production and the treatment of low volume and high value added chemical products. Tunisia is well known by its olive oil and we are interested in the reaction carried in the semi-batch reactors for the esterification of crude acid of olive oil as shown below :

Acid + Alcohol   Ester + Water

The product with very high added value is the obtained ester. To get the maximum quantity of it, it's important to control the temperature of the reactor. This is a very hard task party because the reaction is highly exothermic.

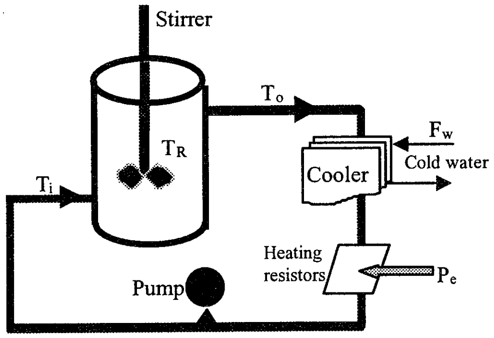


Figure 4 : Experimental scheme for the semi-batch reactor

We give in figure 4 a simplified scheme of the experimental setup installed in the National School of Engineers of Gabès (ENIG). The temperature T_R is supposed to be the same in the whole volume of the reactor by using a vertical stirrer continuously rotated at sufficient speed. External water circulating through a surrounding jacket is used to regulate T_R. We measure inlet temperature T_i and outlet temperature T_o.

Depending on whether the reactor temperature has to be raised or lowered, the regulating water is either heated or cooled. It can be heated by a set of resistors whose electric power P_e can be varied from 0 to 3500 W and it can be cooled by using a tubular cooler whose cooling rate is changed by varying the flow F_w from 0 to 1200 l/h. The reaction is carried at atmospheric pressure and some precautions are considered to recycle all evaporated solvent.

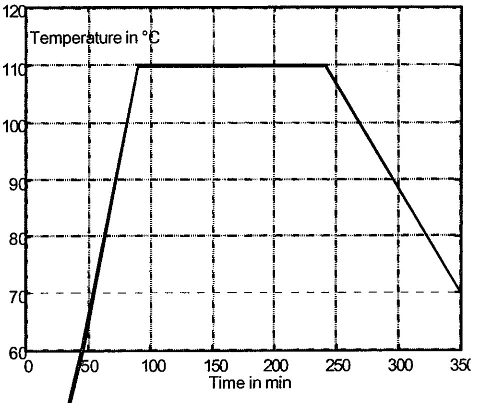


Figure 5 : Desired output

The typical temperature profile considered for the reactor is given in figure 5. We notice the existence of three regions :

- Preheating, T_R increases from ambient to 110 °C during 90 min.
- Maintain T_R to 110 °C for 150 min.
- Decreasing T_R to ambient according to a predefined temperature linear profile during 90 min.

A preliminary experiment has been performed using a discrete PID based control structure. As shown in figures 6a and 6b, the results are very unsatisfactory. In fact, the reactor temperature presents important deviations with regard to the reference trajectory and the control variable is often varying. A disturbance on F_w made between 150 and 160 min is not sufficiently carried out and T_R presents an important undershoot.

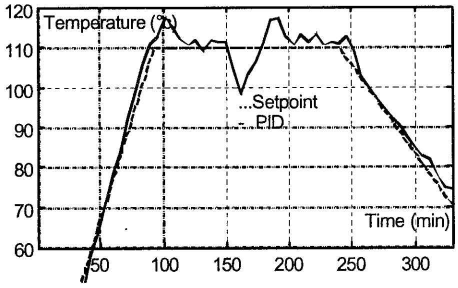


Fig.6a. Mixed temperature with the discrete PID control structure (with disturbance)

To solve this problem, we proposed a constrained adaptive generalized predictive controller (CAGPC) [3, 4]. The purpose of the control algorithm is to select future control moves that will minimize a performance function based on the desired output trajectory over a prediction horizon subject to constraints on the output and the control variable.

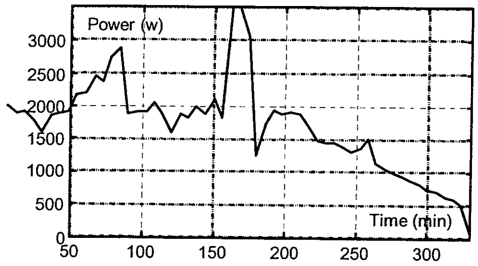


Figure 6b Electric power supply with the discrete PID control structure (with disturbance)

The used model is a reduced linear dynamic one which is obtained from input output measurements.

An IBM PC Pentium processor is used as shown in figure 7 for process monitoring and control. CAGPC is performed in compiled software using a real time environment with a closed loop control-sampling rate of 30 s [5].

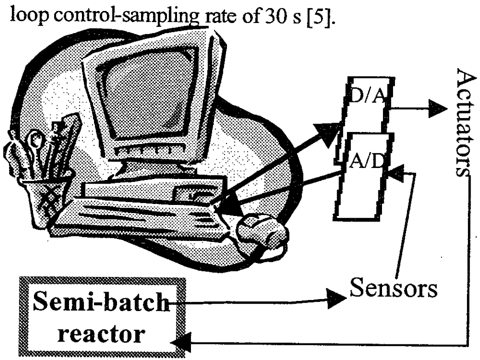


Figure 7 : Process monitoring

The reactant temperature and the electric power supply variables are shown in figure 8a and 8b respectively. It is clear that the CAGPC based system shows a closer correspondence to reference trajectory, an asymptotic perfect tracking of the set point and T_R, the absence of oscillations in T_R and signal stability.

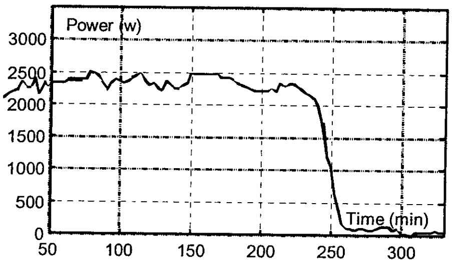


Figure 8b Electric power supply for the CAGPC (with disturbance).

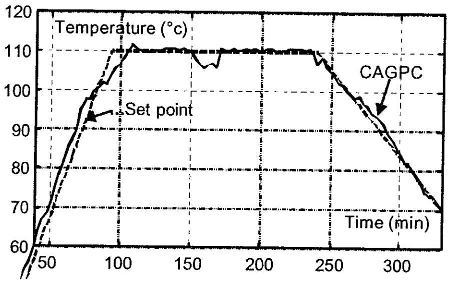


Figure 8a : Closed loop response for the CAGPC (with the disturbance)

III. FUZZY SUPERVISION OF THE GPC IN THE CASE OF THE LEVEL CONTROL OF LIQUID IN A TANK.

The results that we present here are detailed in reference [6]. The main idea is to use fuzzy logic for the on-line adjustment of the parameters of a GPC. It is well known that GPC is more efficient than the popular PID controller. But, the problem is that GPC requires expert engineers to choose the values of design parameters such as the horizons of initialization, of prediction and of control (HI, HP and HC), or the weighting coefficient of the control action increments (λ). The insertion of a fuzzy supervisor can solve the problem and replace the expert [6, 7]. Hence the supervisor will be in charge of guaranteeing the desired performances and reasonable conditioning parameters in regards of the dynamic behavior of the process.

During the design of a control law, the desired performances include often three types of pecifications, namely precision, rapidity and good degree of stability. In addition to these specifications, it is always desirable to have the smallest possible control variance, which means a long conservation of the control drive device. It is also very important, especially in the case of the GPC algorithm to take into consideration the numerical problems to deal with real time applications. Table 1 summarizes the retained performances criteria [6] :

- TMON represents the rise time and give information about rapidity.
- VARU is the control variance calculated recursively on a sliding window.
- DEGSTA evaluates the degree of stability by calculating the minimal distance between the Nyquist locus of compensated system's transfer function and the critical point (-1, 0).
- TRACE is the value $\mathrm{trace}(Q^TQ)$ and is used as criterion to take into account the numerical problems, the matrix Q is constructed from the parameters of the model which are estimated at each sampling period. This matrix intervenes in the calculation of the optimal control increments.

I. PERFORMANCE	Rapidity	Control variance	Degree of stability	Conditioning of computations
Index	TMON	VARU	DEGSTA	TRACE

Table 1 : The retained performances criteria

The considered process represented in figure 9 consists of a tank, a centrifugal pump that feeds the liquid through a proportional valve, two transmitters one for the control variable of the valve and the other for the measured level of fluid in the tank, two on/off draining valves are used to generate disturbances. The process is connected to a computer and a 12 bits A/D and D/A converters are used. The main objective of the control system is to maintain, by manipulating the proportional valve, the tank liquid level at a specified set point.

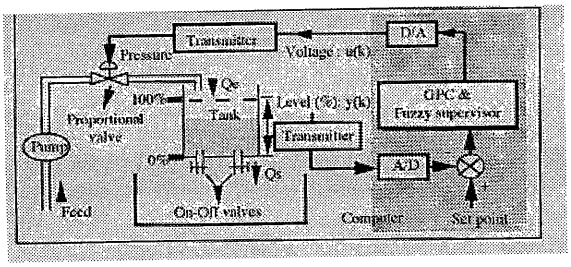


Figure 9 : Schematic of the real process n°3

Figures 10 and 11 show that the obtained results in presence of the fuzzy supervisor are much better compared to the case where the supervisor is off-line [6, 8]. The variances of the control and output signals are sensibly reduced.

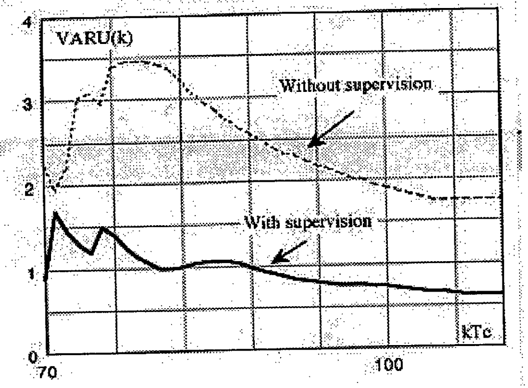


Figure 10 : The variance of the control

I. CONCLUSION

Three real applications have been considered and we show in each case that the introduction of numerical real time control using predictive or adaptive controller improves the system behavior greatly by comparison to the classical PID controllers.

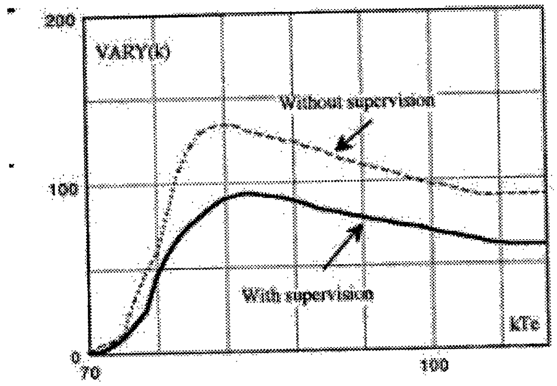


Figure 11 : The variance of real output

REFERENCES

[1] MSAHLI F., BEN ABDENNOUR R. & KSOURI M."Identification structurelle et paramétrique d'un refroidisseur à grilles de clinker d'une cimenterie". First international conference on electronics and automatic control, Tizi-ouzou-Algerie, 4-6 Mai 1992, pp. 58-66.

[2] MSAHLI F., BEN ABDENNOUR R. & KSOURI M. , "Identification et commande adaptative d'un refroidisseur industriel à grilles de clinker d'une cimenterie". Les Annales Maghrébines de l'Ingénieur, vol. 9, N° 1, Avril 1995, pp. 37-46.

[3] BOUANI F., BEN ABDENNOUR R., KSOURI M. & FAVIER G., "La commande prédictive généralisée: Validation d'une base des règles sur un procédé réel". $2^{ème}$ Conférence Maghrébine sur l'Automatique, l'Electrotechnique et l'Electronique industrielle, Alger, 35 Décembre 1996, Vol. 1, pp. 304-309.

[4] BOUANI F. , BEN ABDENNOUR R., KSOURI M. & FAVIER G., "Evaluation des performances en fonction des paramètres de synthèse d'une commande prédictive généralisée". Mediterranean Conference on Electronics and Automatic, Grenoble, 13-15 Septembre 1995.

[5] MSAHLI F., FAYECHE Ch. & BEN ABDENNOUR R., "Application of adaptive controllers for the temperature control of a semi-batch reactor", International Journal of Computational Engineering Science, accepted.

[6] BEN ABDENNOUR R., KSOURI M. & FAVIER G., "Application of fuzzy logic to the on-line adjustment of the parameter of a generalized predictive controller". Journal: Intelligent Automation and Soft Computing,vol.4, Num3, pp.197-214, 1998.

[7] BEN ABDENNOUR R., FAVIER G. & KSOURI M., " Fuzzy trace identification algorithm for non-stationary systems". Journal: Intelligent & Fuzzy Systems,6 (1998).

[8] BOUANI F., BEN ABDENNOUR R., KSOURI M. , "Commande prédictive d'un procédé chimique: Régulation de niveau". COMAGEP'96, Djerba-Gabès (Tunisie), Avril 1996.

Application of Soft Computing Techniques to Human Centered Systems

Takehisa Onisawa
Institute of Engineering Mechanics and Systems
The University of Tsukuba
1-1-1, Tennodai, Tsukuba, 305-8573 Japan
onisawa@esys.tsukuba.ac.jp

Abstract—**The definition of human centered systems is given as systems in which human subjective judgment, evaluation, recognition and emotion play central roles. Information dealt with in human centered systems has subjectivity, ambiguity, vagueness in the sense of diversity of sense of values, and situation dependence. Therefore, outputs of human centered systems are not necessarily unique for same inputs. This paper also discusses that the soft computing techniques including fuzzy set theory, a neural network model, genetic algorithms, etc. is appropriate for the approach to human centered systems. Finally, two study examples, the facial recognition system considering situations and automatic music composition system, are introduced. The former employs a neural network model and the latter uses the interactive genetic algorithms.**

Keywords—human centered systems, soft computing techniques, facial recognition, music composition

I. INTRODUCTION

What are human centered systems? Human centered systems have various sides. Roughly speaking, there are some concepts similar to human centered systems, e.g., human friendly systems [1], human centered design [2], humanistic systems [3]. Human friendly systems are defined as systems that support human behavior and thinking activities when human is faced with complex and difficult real problems [1]. The concept of the human centered design comes from the field of human interface. Manufactured goods were designed and produced so as to have good function and performance. However, there has been reconsideration whether manufactured goods with good function and performance improve the quality of our life or not. Therefore, the concept of usefulness of goods, that is, usability in the broad sense, is proposed. Furthermore, the concept of human centered design is recently proposed, which considers the satisfaction at the use of goods including attachment to the goods and sensuous acceptance whether the goods are accepted or not, as well as good function, good performance and usability. International standard ISO13407 (Human Centered Design for Interactive Systems) considering usability, performance and satisfaction is established [2]. The humanistic systems are proposed by Prof.Zadeh [3] known as a proposer of fuzzy sets [4]. The humanistic systems are systems in which human judgment, recognition, and emotion play central roles. These three concepts are dependent on each other. In this paper human centered systems are defined as subjective systems in which human judgment, evaluation, recognition and emotion play central roles as the humanistic systems.

This paper describes the characteristic of human centered systems. Information dealt with in human centered systems has subjectivity, ambiguity, vagueness and situation dependence rather than objectivity, uniqueness, universality and reappearance [5], where ambiguity means diversity of sense of values. This paper also mentions that the soft computing techniques [6] are appropriate for information processing in human centered systems. Finally this paper shows some examples of human centered systems implemented by the use of the soft computing techniques.

II. HUMAN CENTERED SYSTEMS THINKING

A. Fuzzy Thinking

Considering human centered systems as subjective systems in which human judgment, evaluation, recognition and emotion play central roles, information dealt with in human centered systems is mainly subjective information including Kansei information [7] rather than objective one. Subjective information has subjectivity, ambiguity, vagueness and situation dependence, where ambiguity means diversity of sense of values. These properties are opposite to the properties of information dealt with in conventional natural science, i.e., objectivity, uniqueness, universality and reappearance [5]. Therefore, the interpretation of information in human centered systems has diversity and its diversity should be accepted in the analysis and the design of human centered systems. That is, for the same input, human centered systems are permitted to have various outputs.

However, if human centered systems have not consistency with respect to input-output relation to some extent, human centered systems merely have random outputs, and then human centered systems are not systems. Human centered systems should have not crisp consistency, but fuzzy consistency with respect to input-output relation. Therefore, human centered systems dealing with information that has subjectivity, ambiguity, vagueness, and situation dependence, and having fuzzy input-output relation, are considered as fuzzy systems. These systems should be analyzed and/or designed based on individual or group sense of values. On the other hand the conventional system analysis and/or design are performed based on objective evaluation or sense of values accepted by almost all people. Therefore, as a matter of course, for same inputs the system has same outputs. From the viewpoint of the system design the latter concept is based on mass production of small kinds of goods, that is, goods are produced based on some standards. On the other hand, the former concept leads to a little production of many kinds of goods. This means that the number of goods to be designed is the same as the number of sense of values and the goods fitting an individual are the standards for the person.

B. Methodology

Which methodology is more appropriate for the analysis and the design of human centered systems? A flexible computation algorithm is more appropriate for information processing in human centered systems rather than a crisp one since information has subjectivity, ambiguity, vagueness and situation dependence and human centered systems have fuzzy input-output relation. Recently, the soft computing techniques [6] are proposed as the complementary way of an advantage and a disadvantage of variety of approaches such as fuzzy theory, a neural network approach, genetic algorithms [8]. The advantage and the disadvantage have the following meanings. Although fuzzy theory [9] is applicable to express fuzziness with respect to subjectivity, it is unsuitable for acquisition of fuzzy information and knowledge. On the other hand, a neural network model [10] and genetic algorithms [11] are applicable to acquisitions of them. Unfortunately, however, these approaches are unsuitable for representations of acquired information and knowledge. That is, each approach has merits and demerits [8]. The soft computing techniques are whole techniques as complementary ways of the merits and the demerits. The soft computing techniques are appropriate for the processing of information having subjectivity, ambiguity, vagueness and situation dependence.

By the way, it is necessary to know the response of human centered systems for some inputs. It is also necessary to express the input-output relation by some ways for the analysis and the design of human centered systems. The former means learning and knowledge acquisition and the latter means the expression of a system behavior. Although the complex system behavior is often difficult to be expressed by differential equations, human experts know the system behavior by their experience and the system behavior can be often expressed in a plain language based on their experience and knowledge. Fuzzy theory is useful for the expression of the system behavior in the language form, i.e., the if-then form. Its applications are observed in the fuzzy control system [12]. On the other hand knowledge acquisition is studied from various points of view. For example, the model acquisition of an ordinary system is performed using some evaluation functions. A system is usually identified by the least mean square error between the model output and the system output. However, usual standard evaluation functions are not useful for the acquisition of the model of human centered systems since individual subjectivity and diversity of sense of values are not reflected by standard evaluation functions well and are difficult to be expressed quantitatively. Which evaluation function should be considered for the modeling of human centered systems? Human direct evaluation is one of good evaluations for the modeling. This paper suggests two methods. One of them is the use of questionnaire. The analysis of questionnaire data has the possibility to model human centered systems reflecting individual subjectivity and diversity of sense of values. However, this method is difficult to be applied to the real time modeling of human centered systems. Therefore, the questionnaire method is applied to not real time modeling but off-line modeling. The other is the application of the interactive genetic algorithms (abbreviated as Interactive GA) [13]. In the Interactive GA conventional fitness functions are not used and human direct evaluations are embedded into the GA process. Therefore, the model obtained by the Interactive GA can reflect individual subjectivity and diversity of sense of values well. Furthermore, real time modeling is possible. The Interactive GA is applied to the artistic field as well as the engineering filed [14]. However, the Interactive GA has some problems. One of them is human fatigue [13] since human must repeat simple evaluation generation by generation. Therefore, the population size and the number of generations in the Interactive GA must be smaller than those in the conventional GA in order to decrease human load. The small population size and the small number of generations, however, lead to the GA results dependence on initial values and worse convergence. Therefore, various methods are used in order to solve the problem [13].

III. Modeling Examples of Human Centered Systems

In this paper two examples are introduced. One is the recognition model of facial expressions considering situations [15]. The model employs a neural network model and questionnaire data. The other is an automatic melody composition system reflecting a user's feeling [16]. The system employs the Interactive GA. This paper considers these studies as one of human centered systems in the sense that various outputs reflecting an individual feeling well are obtained for the same input.

A. Facial Recognition Model Considering Situations

If a computer recognizes human psychology and emotions through human facial expressions, a human-computer interaction is hoped to become a natural bi-directional communication. In this sense the recognition of human facial expressions becomes important in a human-computer interaction [17]. Of course, the face recognition is much important in areas such as security systems, identification of criminals as well as a human-computer interaction. So, there have been many approaches to the face recognition [18-22]. In most of these approaches, only the relationship between facial expressions and emotions is considered. In practice, however, for the recognition of facial expressions human uses information not only facial expressions themselves but also gesture, voice pitch, situations in which human is put. Especially, situations have an influence on facial expressions. Facial recognition considering situations is studied with respect to faces by line drawing [23]. This section describes a recognition model of emotions from real face image and situations.

1) *Basic Emotions*

This study considers the following 6 kinds of emotions, happiness, surprise, anger, disgust, sadness and fear that are well known as basic emotions and are proposed by Ekman [24].

2) *Recognition Model of Emotions*

The structure of the model is shown in Fig.1. A real face image and a situation are inputs to the model. Fourteen features values of a face image such as the size of eyes, the size of a mouth are extracted from a real face image by the image processing [25]. The features values are inputted to the first model that is composed of six neural network models. Each neural network model recognizes the degree of one of six basic emotions using only facial expressions. Information on a situation is transformed into basic emotions. For example, if surprise and sadness are felt in the situation that I get worse grades at school, the situation is transformed into surprise and sadness, where the degrees of emotions are inputted as they are.

The degrees of emotions recognized from facial expressions of real face image by the first kind of neural network model and the degrees of emotions transformed from the situation are inputted to the second neural network model. The outputs of the second neural network are degrees of basic emotions considering facial expressions and situations.

The recognition model consists of seven models, each of which recognizes the degree of one of six basic emotions or the degree of unnaturalness of the combination of a situation and a facial image, where the unnaturalness means the unnatural feeling from the presented combination of a facial expression and a situation. A set of the degrees of six kinds of emotions and unnaturalness is the output of the recognition model.

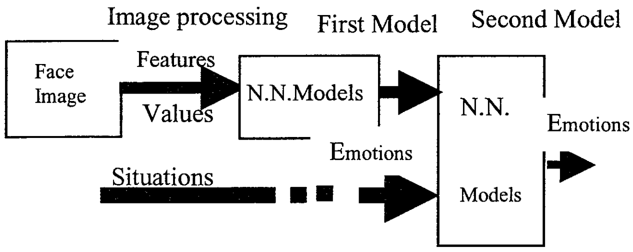


Fig.1 Structure of Recognition Model

3) *Features Values*

Features values $V_1 - V_{14}$ as shown in Fig.2 are obtained using the coordinates of features point extracted by the image processing [25], which are defined by the distance between features points. These features values are normalized by the distance between the left edge point of a right eye and the right edge point of a left eye since these positions don't change in every facial expression. Furthermore, the first normalized features values are normalized again using values in a neutral face, i.e., a wooden face, and the maximum and the minimum values of features values among all facial expressions, where the features values in a neutral face, the maximum and the minimum values are obtained in advance. Eq. (1) is used for the second normalization. The value $\tilde{V}_k\ (k=1,2,...,14)$ in Eq. (1) are used as inputs of the model about information on facial expressions.

$$\tilde{V}_k = \begin{cases} \dfrac{V_k - V_{k-neutral}}{V_{k-\max} - V_{k-neutral}} & (V_k \ge V_{k-neutral}) \\ & (k=1,2,...,14) \\ \dfrac{V_k - V_{k-neutral}}{V_{k-neutral} - V_{k-\min}} & (V_k < V_{k-neutral}) \end{cases} \tag{1}$$

where $V_{k-\max}$ and $V_{k-\min}$ are the maximum and the minimum values of V_k among all facial expressions, respectively, and $V_{k-neutral}$ are the features values of V_k in a neutral face.

4) *Questionnaire about Situations*

In this section information on a situation is transformed into emotions. Therefore, in order to obtain the relationship between situations and emotions, the questionnaires about situations are performed. Subjects are asked to answer the degree of each emotion according to the degree of each emotions felt under the presented situation with four scale evaluations, *no feeling*, *a weak feeling*, *a moderate feeling* and *a strong feeling* of six basic emotions. The questionnaire is performed by 6

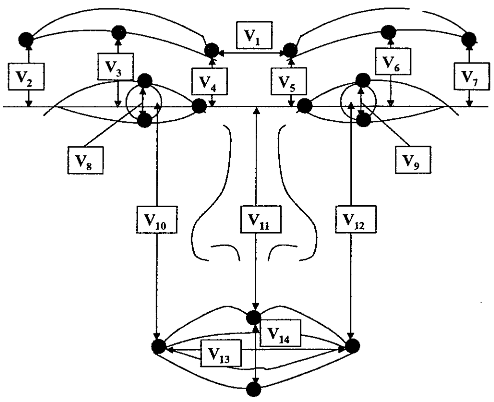


Fig. 2 Features Values

students for 30 kinds of situations such as “He passes an exam unexpectedly”, “He loses his way”.

5) Questionnaire about Facial Expressions and Situations

In this questionnaire, subjects are asked to compare a facial expressions with a neutral one and to answer the degree of each emotion felt under the presented situation with the same way as the questionnaire about situations. If subjects feel that the combination of a situation and a facial expression is unnatural, they are asked to answer the unnaturalness. The questionnaire is performed for 120 combinations that are chosen from the combinations of 20 facial expressions and 20 situations by the same students as the questionnaire about the situations.

As for the questionnaire about facial expressions, the questionnaire without a situation is employed.

6) Learning of Neural Networks

With respect to the degrees of emotions in the questionnaires about situations, and those about facial expressions and situations, natural language expressions are assumed to correspond to numerical values in [0, 1] such as *no feeling:0.00, a weak feeling:0.33, a moderate feeling:0.67*, and *a strong feeling:1.00*. If the unnaturalness is answered, the numerical value about the unnaturalness is assigned to 1.0.

Seven neural network models are obtained by the backpropagation algorithm. One set of seven neural network models is regarded as the individual recognition model of each subject’s own. Each neural network model has one input layer, one output layer and two hidden layers. Each hidden layer has 20 nodes. Six individual models are obtained since six subjects’ questionnaire data are used for the learning of neural networks separately.

7) Evaluations of Obtained Models

Fig. 4 shows examples of recognition results by subject B’s model and subject C’s model, where these combinations of facial expressions and situations are not used for the learning of neural networks. It is found that although the same facial expression is given, a variety of recognition results are obtained depending on subjects and situations. In order to verify the model performance, recognition results by the obtained models are compared with subjects questionnaire results. The six subjects are asked to answer the questionnaire about 48 combinations of facial expressions and situations, which are not used for the learning of neural networks. The degrees of emotions recognized by the models are classified into as follows.

$$\begin{aligned}
&\textit{recognition result} \in [0.0, 0.25) \Rightarrow 1.\textit{no feeling}\\
&\textit{recognition result} \in [0.25, 0.50) \Rightarrow 2.\textit{a weak feeling}\\
&\textit{recognition result} \in [0.50, 0.75) \Rightarrow 3.\textit{a moderate feeling}\\
&\textit{recognition result} \in [0.75, 1.0] \Rightarrow 4.\textit{a strong feeling}
\end{aligned} \qquad (2)$$

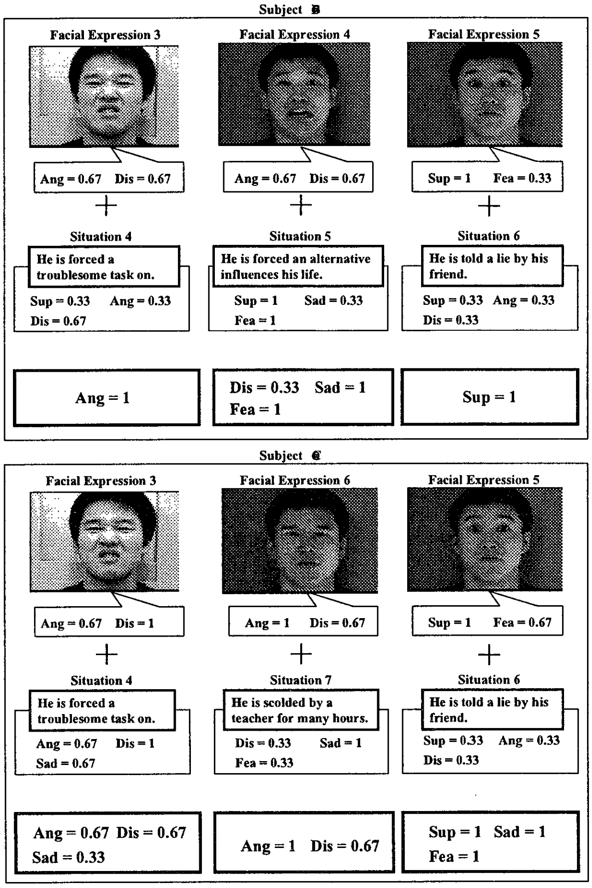


Fig. 4 Recognition Examples

Models outputs for these combinations are compared with the questionnaire results by the subjects. Let the difference of the degree between, for example, a weak feeling and a strong feeling, be 2. And let a difference scale in the combination be the total sum of the difference of each emotion.

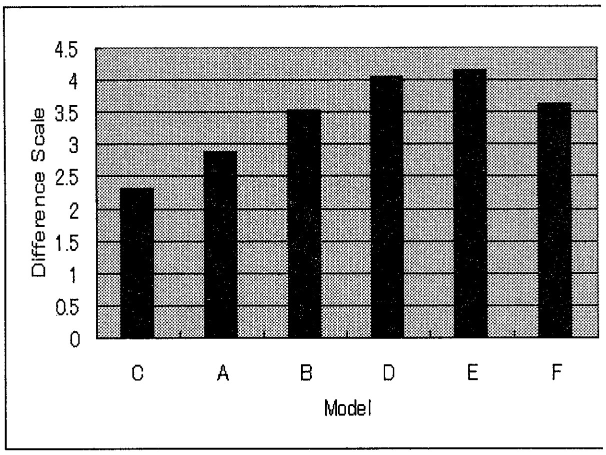


Fig.5 Example of Average Difference Scales

Table 1 Test Statistics t_0

	Hap	**Sup**	**Ang**	**Dis**	**Sad**	**Fea**	**Unnat**
A	0	1.911	1.772	0.186	0.419	0.761	-1.528
B	-1.019	-0.182	0.552	-0.757	-0.812	-0.596	-0.275
C	-0.318	0.851	$\boxed{3.288}$	0.006	-0.938	0.549	-1.663
D	1.353	-2.011	-0.785	0.166	-0.260	-2.022	0.240
E	-2.000	-0.211	0.295	$\boxed{3.181}$	0.358	-0.542	1.159
F	1.829	-0.861	0.004	-1.131	-0.004	-0.176	0.374

Fig. 5 shows an example of average difference scales between the subject C's questionnaire results and the recognition results by 6 individual models for 48 combinations of facial expressions and situations. It is found that the individual recognition model of the subject C's own has the smallest difference among the individual models of subjects' own. It is found that the individual model reflects the feeling of the subjects' own well.

Finally, the test statistics t_0 obtained using $d_{ijk} = x_{Sijk} - x_{Mijk}$ are tested, where x_{Sijk} is the subject i's questionnaire result for the j-th emotion in the k-th combination, and x_{Mijk} are recognition results by subject i's model for the j-th emotion in the k-th combination, $i = 1, 2, \ldots, 6$ (the number of the subjects), $j = 1, 2, \ldots, 7$ (the number of the emotions including unnaturalness) and $k = 1, 2, \ldots, 48$ (the number of the combinations).

Table 1 shows the test statistics t_0. The critical region is assumed to be $t(47, 0.01) = 2.685$. It is found that $t_0 > t(47, 0.01)$ for the subject C's model recognition results of anger and for the subject E's model recognition results of disgust. That is, their models have tendency to recognize these emotions rather small. Then it is not said that there is no difference between subjects' questionnaire results and models recognition results. However, it is also found that $t_0 < t(47, 0.01)$ for excepting above cases. Therefore, it is not said that there is difference between subjects' questionnaire results and models recognition results. The obtained models can recognize the emotions reflecting subjects feeling and subjectivity well.

8) Remarks

In this section emotion recognition models from the combination of real face images and situations are described as one example of human centered systems. These models are obtained using individual questionnaire data and have various recognition results depending on situations for same facial expressions and depending on subjects for same combination of facial expressions and situations. In this sense these models reflect individual feeling and subjectivity well.

B. Automatic Melody Composition System Reflecting User's Feeling

A study on automatic music composition using a computer began with the birth of computers. Recently, software package has been sold, by which even amateurs can compose music using personal computers. However, a feeling of music varies among people because of a variety of favorite music [26]. Therefore, melodies composed by some automatic composition techniques do not necessarily satisfy all listeners if a human feeling of music is not reflected in the automatic music composition. A human feeling is difficult to be expressed quantitatively, for example, by some functions, and to be embedded into computer programming since it has qualitativeness, subjectivity, ambiguity, vagueness and situation dependence. Recently, the concept of the Interactive GA is proposed and it is applied to the artistic field as well as the engineering field [14]. Human evaluation is embedded into the GA process in the Interactive GA instead of fitness functions. This section describes automatic music composition system reflecting an individual feeling of music using the interactive GA. Of course, there are studies on music composition with the GA or the interactive GA [27-29]. These studies are on finding the optimum methods for the structuring musical algorithm, on a system playing jazz session with human performer, and on a support system of composing music. In this section, human evaluations are attached greater importance to than these studies. That is, this study considers a user as a music composer in the sense that user's evaluation to music plays an important role and that a user gets a melody satisfying himself/herself [16]. Therefore, the presented system is considered as one example of human centered systems.

1) Assumptions

For the construction of an automatic music composition system this section has some assumptions: (1) Music theory is applied to music composition partially so that music is becoming to a tune and gives various impressions to listeners. (2) Composed music has one main melody part and 4 backing parts. The pattern of the backing parts is fixed, which is chosen from the patterns of the backing part of popular music [30]. The tone color of each part is chosen from the MIDI tone color database. (3) The composed work has 4 music bars with a time signature 4/4. Although this length is short comparing with ordinary music works, it is easy for a composer to evaluate short work paying attention to delicate changes of pitch notes and a composer's own liking for a melody line is reflected well in short music. (4) The tune of notes is chosen from C-Major Scales and the fundamental tune is fixed at c, i.e., do. The octave of this fundamental tune varies in every part such as c5 (c in the 5th octave), c4 and c3. In each part the interval range between the fundamental tune and each note's tune is less than 1 octave. Half steps of pitch are not used. (5) The concept of ordinary harmonic progression [31] is introduced. Chord names on four bars are C, F, G, and C. Each note in a backing part is chosen so that it suits this kinds of progression. (6) The concept of block, i.e., a set of notes, is employed in order to arrange the length of notes in the melody part. Three kinds of block types consisting of 4 beats, 2 beats, or 1 beat are defined. The number of blocks assigned to a melody in 4 bars is from 4 through 16. Eight kinds of note lengths such as 4, 3, 2, 1.5, 1, 0.75, 0.5 and 0.25 are considered. The number of notes assigned to one block is determined according to the block type and the note length.

2) *Procedure of Music Composition*

The procedure of music composition is shown in Fig. 6. (1) A composer has a feeling of music that he/she tries to compose, e.g., "cheerful music". (2) The system generates 200 chromosomes using the database based on music theory, and then presents 20 chromosomes (works) that are chosen at random. (3) A composer listens to 20 melodies one by one, and evaluates them based on his/her subjectivity. (4) The system gives the GA operations to chromosomes that are chosen at random from new 200 chromosomes. The procedures (3) and (4) are repeated until a melody reflecting a composer's feeling is composed.

3) *Coding of Music*

A music work corresponds to one chromosome as shown in Fig.7. Information on the number of chords, the number of backing patterns and backing parts, the scale number, the chord name, the number of blocks, the length of each block (1 beat, 2 beats or 4 beats), tone color, the number of notes, note pitch (the difference between the fundamental tune and a note tune), the octave of the fundamental tune in each part and note length is coded in a chromosome by integer. The melody part plays an important role to give a listener a feeling of music. Therefore, gene information on the melody part, i.e., the number of blocks, the length of each block, tone color, the number of notes, note pitch and note length, is operated by the GA operations.

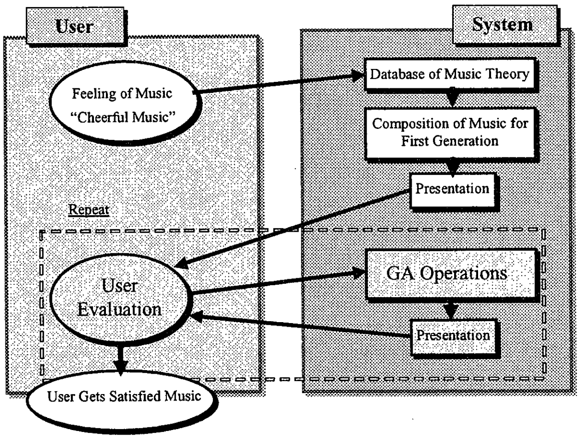


Fig. 6 Procedure of Music Composition

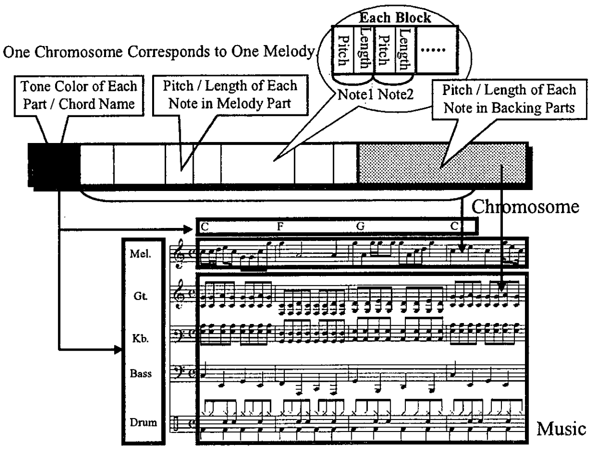


Fig. 7 Coding of Music

4) *Evaluation in GA Process*

As mentioned before, although the results of the GA process are necessary to be evaluated in order to evolve individuals, it is difficult to evaluate the results by some fitness functions in the artistic field such as music composition. The direct evaluation of a human user should be used instead of fitness functions in the field. In this section three kinds of composer's evaluations are considered.

Twenty melodies are presented through a human interface in the system. When a composer clicks the Play button of the interface, the system starts playing music. The pointer showing the playing position of the melody line moves and the pointed block color turns into yellow in order. A composer listens to music and then has a total evaluation, a partial evaluation and the choice of the best work. The total evaluation is a relative evaluation between each tune and the criterion tune explained below. The criterion is displayed on the top of the interface. A composer evaluates each tune with 7 scales evaluation (+3, +2, ..., -2, -3) to what degree it fits a feeling of music comparing with the criterion music. The partial evaluation means the evaluation of every block in a melody line. If a composer feels good for a block, he/she gives the block a good evaluation. If a composer feels bad, he/she gives it a bad evaluation. The choice of the best work means that a composer chooses the best work after the total and the partial evaluations for 20 works in the sense that it fits his/her feeling of music best in a generation. The chosen work is the criterion in the next generation. If there is no better work than the criterion, a composer can choose the criterion again.

5) *Strategy of Elitist Selection*

The best chromosome chosen by a composer is considered as an elite chromosome. The elite is left as it is, and it is presented to a composer as the criterion work in the next generation.

6) Selection of Parent Chromosomes

The composer's total evaluation is reflected to the number of next generation parent chromosomes. If the composer's evaluation for a chromosome (a work) is plus, chromosomes with the four times number of the evaluation value are copied as the next generation parent chromosomes. If it is minus or zero, the chromosomes are not chosen. If the number of parent chromosomes is less than 200, the deficiency is supplied by choosing non-presented chromosomes at random.

7) Crossover and Mutation

The crossover is defined as the copy operation from a part of a chromosome to that of another one. Four kinds of crossovers are operated between two chromosomes chosen at random from the pool of chromosomes as shown Fig.8. The probability of each type of the crossover is defined as shown in Fig.8. After the crossover, the mutation is operated to 30% of 199 chromosomes. The mutation is defined as a re-construction of some blocks of the melody part.

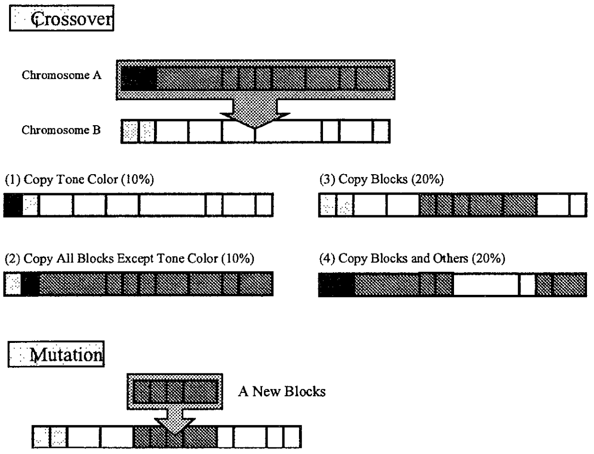


Fig. 8 Crossover and Mutation

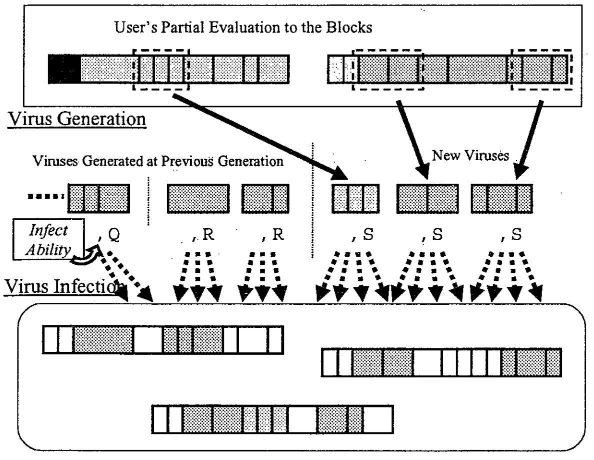


Fig.9 Virus Generation and Infection

8) Virus Generation and Infection

The composer's partial evaluation is reflected to the GA operations as the virus generation and its infection. Fig.9 shows the structure of the virus generation and its infection. The block, to which a composer gives a good evaluation, plays a virus role. After the crossover and the mutation, some chromosomes are infected with viruses. The infection is defined as the copy of the viruses to a part of chromosomes. An infect ability is given to a virus as the parameter value, which is assumed to be 4 in this section. It means that a virus is copied to 4 chromosomes. Once a new virus is generated, it infects within 4 generations. The infect ability is decreased one by one with the progress of the generations.

8) Experiments of Music Composition

Six subjects try to compose cheerful music using the present system. In the experiments, the subjects repeat 10 generations. After 10 repeats, 10 chosen works are presented at random. The subjects listen to the works and evaluate them with 10 scales evaluation (+1, +2, ..., +9, +10).

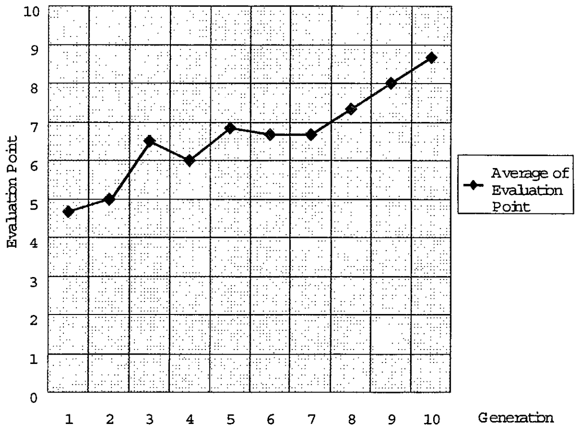


Fig. 10 Average Evaluation

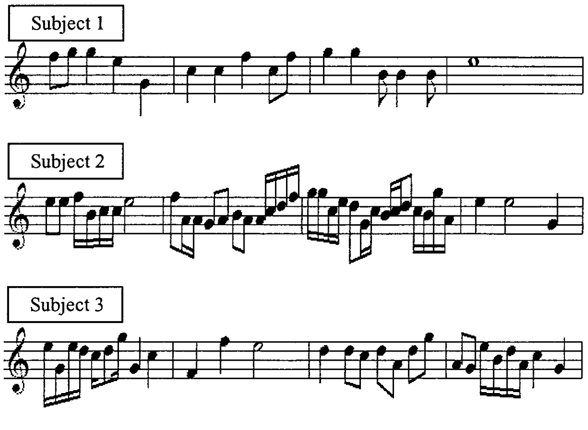



Fig. 11 Examples of Composed Scores

Fig. 10 shows the average evaluations for 10 best works among the subjects. The horizontal line expresses the generation and the vertical line expresses the average value. From Fig. 10 it is found that the average value becomes high with the progress of the generations. Fig.11 shows three subjects melody parts scores, which are chosen as the best work at the 10th generation by subjects themselves. Although the subjects try to compose a cheerful melody in the experiments, it is found that many kinds of melody lines are composed by the system and that the subjects give high evaluation to these works.

Although composers try to compose cheerful melody, that is, the system has the same input, the system generates varieties of cheerful melodies and the subjects give these melodies high evaluation. The system generates melody lines reflecting composers feeling of music well.

IV. CONCLUDING REMARKS

Human centered systems are defined as a subjective system in which human judgment, evaluation, recognition and emotion play central roles. Information dealt with in human centered systems has subjectivity, ambiguity, vagueness in the sense of diversity of sense of values and situation dependence. These properties have been excluded in conventional natural science. This paper discusses that the so-called soft computing techniques including fuzzy set theory, a neural network model, genetic algorithms, are appropriate for the analysis and the design of human centered systems. This paper also introduces two study examples as examples of human centered systems. The one is the facial recognition system considering situations. The system consists of some neural network models and gives flexible recognition results. The other is an automatic music composition system reflecting a composer's feeling of music. The interactive GA is applied to the system. The experiments results show that even if composers try to compose a cheerful melody using the system, variety of melodies are obtained and composers give them high evaluation.

The concept of human centered systems becomes important in the field of engineering. The soft computing techniques are appropriate for the approach to the human centered systems.

REFERENCES

[1] Laboratory for International Fuzzy Research ed., *Intelligent Information Processing by Fuzzy Thinking*, Computer Age, Tokyo, 1995.

[2] M.Kurosu, "Human Centered Design and Kansei", in *Kansei Engineering and Computerization Society*, M.Ohsawa ed., Morikita, Tokyo, pp.127-153, 2000.

[3] L.A.Zadeh, "Fuzzy Set Theory –A Perspective-", in *Fuzzy Automata and Decision Process*, M.M.Gupta, G.N.Saridis and B.R.Gaines eds., Elsevier, New York, pp.3-4, 1977.

[4] L.A.Zadeh, "Fuzzy Sets", *Information and Control*, Vol.3, pp.338-353, 1965.

[5] T.Onisawa, "Soft Computing Technique in Kansei (Emotional) Information Processing", in *Soft Computing and Human Centered Machines*, Z.-Q.Liu and S.Miyamoto eds., Springer-Verlag, Tokyo, pp.215-248, 2000.

[6] L.A.Zadeh, "Applications of Fuzzy Technology and Soft Computing ", *Journal of Japan Society for Fuzzy Theory and Systems*, Vol.5, No.2, pp.261-268, 1993.

[7] S.Iguchi, et al., *Kansei Information Processing*, Ohmusha, Tokyo, 1994.

[8] X.Li, D.Ruan and A.J.van der Wal, "Discussion of Soft Computing at FLINS'96", *International Journal of Intelligent Systems*, Vol.13, No.5, pp.691-745, 1998.

[9] E.Cox, *The Fuzzy Systems Handbook, Second Edition*, Academic Press, MA, 1999.

[10] L.Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*, Prentice Hall, NJ, 1994

[11] M.Mitchel, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, MA, 1996.

[12] M.Sugeno, *Fuzzy Control*, Nikkan Kogyo Shinbunsha, Tokoyo, 1988.

[13] H.Takagi, T.Unemi and T.Terano, "Perspective on Interactive Evolutionary Computing", *Journal of Japanese Society for Artificial Intelligence*, Vol.13, No.5, pp692-703, 1998

[14] H.Takagi, "Interactive Evolutionary Computation –Cooperation of Computational Intelligence and Human Kansei-", *Proc. of the 5th International Conference on Soft Computing and Information/ Intelligent Systems*, Iizuka, Fukuoka, Vol.1, pp.41-50, 1998.

[15] S.Kato and T.Onisawa, "Recognition Model of Emotions from Real Face Image and Situations", *Proc. of the 6th International Conference on Soft Computing*, Iizuka, Fukuoka, pp.157-164, 2000.

[16] M.Unehara and T.Onisawa, "Composition of Music Using Human Evaluation", *Proc. of 2001 IEEE International Conference on Fuzzy Systems*, Melbourne, Australia, 2001 (to appear).

[17] S.Akamatsu, "Computer Recognition of Human Face – A Survey", *The Transaction of The Institute of Electronics, Information and Communication Engineers*, Vol.J80-A, No.8, pp.1215-1230, 1997.

[18] K.Izumitani, T.Mikami and K.Inoue, "A Model of Expression Grade for Face Graphs Using Fuzzy Untegral", *System and Control*, Vol.28, No.10.pp590-596, 1984.

[19] H.Harashima, "Face, Expressions, and Emotion –An Approach from Video-Robot Technology", *Journal of the Japan Society of Mechanical Engineering*, Vol.95, No.833, pp.503-507, 1992.

[20] H.Ushida, T.Takagi and T.Yamaguchi, "Facial Expression Model Construction Based Associative Memories and Linguistic Instructions Learning", *Trans. IEE of Japan*, Vol.113-C, No.12, 1993.

[21] H.Yamada, "Psychlogical Models for Recognition of Facial Expressions of Emotion", *Journal of the Society of Instrument and Control Engineers*, Vol.33, No.12, pp.1063-1069, 1994..

[22] O.A.Uwechue and A.S.Pandya, *Human Face Recognition Using Third-Order Synthetic Neural Networks*, Kluwer Academic Publishers, MA, 1997.

[23] S.Kitazaki and T.Onisawa, "Inference Model of Facial Expressions and Emotions Considering Situations", *International Journal of Intelligent Automation and Soft Computing*, Vol.17, No.4, 2001 (to appear).

[24] P.Ekman and W.V.Friesen, *Facial Action Coding System (FACS): A Technique for the Measurement of Facial Action*, Consulting Psychology Press, California, 1978.

[25] D.Yang, T.Kunihiro, H.Shimoda and H.Yoshikawa, "A Study of Real-time Image Processing Method for Treating Human Emotion by Facial Expressions", *Proc. of 1999 IEEE International Conference on Systems, Man, and Cybernetics*, Tokyo, Vol.2, pp.360-364, 1999.

[26] T.Taniguchi, *Music and Emotion*, Kita-Ohji Shobo, Tokyo, 1998.

[27] G.L.Nelson, "Sonomorphs: An Application of Genetic Algorithms to Growth and Development of Musical Organisms", *Proc. of the Fourth Biennial Art & Technology Symposium*, Connecticut College, pp.155-169, 1993.

[28] J.A.Biles, "Life with GenJam: Interacting with a Musical IGA", *Proc. of 1999 IEEE International Conference on Systems, Man, and Cybernetics*, Tokyo, Vol.3, pp.652-656, 1999.

[29] T.Unemi and E.Nakada, "A Support System for Music Composition based on Simulated Breeding", *Proc. of the 28th The Society of Instrument and Control Engineers Intelligent System Symposium*, pp.141-146, 2001.
[30] H.Yahagi, *Electric Piano Backing Pattern Book*, YAMAHA Music Media Co., Tokyo, 1999.
[31] Y.Kitagawa, *Handbook of Harmonic Progression*, Rittor Music, Tokyo, 1999.

On Fuzzy and Probabilistic Control Charts

Hassen TALEB
The University of Tunis
Institut Supèrieur de Gestion
Citè bouchoucha Bardo
Tunis Tunisie
hassentaleb@yahoo.fr

Mohamed LIMAM
The University of Tunis
Institut Supèrieur de Gestion
Citè bouchoucha Bardo
Tunis Tunisie
Mohamed.Limam@isg.rnu.tn

Abstract— **In this article different procedures of constructing control charts for linguistic data, based on fuzzy and probability theory, are discussed. Three sets of membership functions, with these different degree of fuzziness, are proposed for fuzzy approaches. A comparison between fuzzy and probability approaches, based on the Average Run Length and samples under control, is conducted for a real data. Contrary to what was concluded by Raz and Wang (1990b) The choice of degree of fuzziness affected the sensitivity of control charts.**

Keywords—**Fuzzy chart, Fuzzy control, statistical quality control, Fuzzy set theory.**

I. INTRODUCTION

DIFFERENT procedures are proposed to monitor multinomial processes when products are classified into mutually exclusive categories. Marcucci (1985) proposed two procedures using Shewhart-type control charts. The first type uses the Pearson χ^2 statistic and is designed to detect changes in any of the quality proportions. The second type uses the multinomial distribution, which can be approximated by a multivariate normal distribution.

Raz and Wang (1988,1990a,b) proposed an alternative approach based on fuzzy theory. Fuzzy sets are assigned to each linguistic term, and then using rules of fuzzy arithmetic they are combined for each sample. The result is a single fuzzy set. A measure of centrality of this aggregate fuzzy set is then plotted on a Shewhart-type control chart. Kanagawa et al. (1991,1993) introduced modifications to the construction of control charts given by Raz and Wang (1990b). They presented a control chart based on the probability density function existing behind the linguistic data. These procedures are discussed by Laviolette et al. (1995).

We propose to compare all these approaches using real data. Using criteria like sensitivity and average run length (ARL), we investigate which approach performs better. Approaches proposed by Marrucci (1985) and Raz and Wang (1990a, b), based on fuzzy and probability theory, will be reviewed in the first section. Results of application of these approaches to the porcelain process will be discussed in section 2. Finally, we examine the sensitivity of control chart approaches to the degree of fuzziness. Measuring of sets fuzziness are presented by Saaty (1975).

II. FUZZY AND PROBABILITY APPROACHES

A. Marcucci Approach

In this section we consider two approaches. The first is used when quality proportions are designed to be specific values, where any change in these proportions must be detected by the monitoring procedure. The second approach allows for one-sided monitoring of quality proportions and is designed to detect only increase in all but one quality proportion.

Suppose we have a multinomial process when proportions. Let $\pi_1, \pi_2, ..., \pi_t$, where t denote these proportions, are required to be specific values for each category. A situation which is analogous to the monitoring problem as it is generally the case for the attribute chart. Here, we consider two cases. One occurs when process proportions are known a priori (Case I). The second case occurs when $\pi_1, \pi_2, ..., \pi_t$ are estimated for a base period where the process is assumed in control (case II). Suppose a case of monitoring a multinomial process we have observations X_{i1}, X_{i2},...,X_{it} of samples i, $i = 1, ..., m$, taken at different monitoring periods. Let X_{i1},X_{i2}, ...,X_{it} denote the number of observations in categories $1, 2, ..., t$, respectively, for the i^{th} monitoring period, when $i = 0$ is the period reserved for the base period, and n_i is the sample size for monitoring period i.

- case I: Proportions $\pi_1, \pi_2, ..., \pi_t$ are specified. To monitor such multinomial process under this condition, we use a standard statistical procedure which is the *Pearson goodness-of-fit* statistic, defined in Duncan (1974) as follows:

$$Y_i^2 = \sum_{j=1}^{t} \frac{(X_{ij}-n_i\pi_j)^2}{n_i\pi_j}. \qquad (1)$$

When the process is in control, the asymptotic distribution of Y_i^2 is $\chi^2_{(t-1)}$, a chi-squared distribution with (t-1) degrees of freedom.

- case II: Specific values of process proportions are not known a priori. Then, the *Pearson goodness-of-fit* statistic is not applicable. Following Marrucci (1985), an appropriate statistical procedure is a test of homogeneity of proportions between the base period (0) and each monitoring period i. This statistic is defined as follows:

$$Z_i^2 = \sum_{k=i,0} \sum_{j=1}^{t} \frac{n_k(\frac{X_{kj}}{n_k} - \frac{X_{ij}+X_{0j}}{n_i+n_0})^2}{\frac{X_{ij}+X_{0j}}{n_i+n_0}}$$

$$= n_i n_0 \sum_{j=1}^{t} \frac{(p_{ij}-p_{0j})^2}{X_{ij}+X_{0j}}, \qquad (2)$$

where $k = \{0, i\}$, $p_{kj} = X_{kj}/n_k$, $j = 1, ..., t$, are the sample proportions, and n_i is the sample size.
The construction of a control chart entails the determination of the values to be used for the centre line and the control limits .

B. Representative value for a fuzzy subset

Each observation in a sample of linguistic data is a linguistic term associated with a fuzzy subset defined on the base variable and described by a membership function. It is necessary to convert the fuzzy subsets into scalars in order to keep the standard format of control chart and to facilate plotting of samples on the chart. We shall refer to these scalars as the representative values of their respective fuzzy subsets. In the literature, there are four methods for transforming a fuzzy subset into it's representative value: fuzzy mode, fuzzy midrange, fuzzy median and fuzzy average.

C. Representative value for a Sample

A sample consists of several observations selected for inspection. Each observation is classified by a linguistic value and is associated with a known membership function. These linguistic values need to be combined in order to yield a single representative value for this sample, similar to the mean in $x$bar-chart or the fraction non-conforming in the p-chart. This combination may be done either before or after the conversion of linguistic variables into a representative value.

In the first case, fuzzy sets associated each linguistic value in a sample should be added and then devided by the number of observations in the sample. The result is a fuzzy set which may not correspond to a specific linguistic value in the term set, but it corresponds to the average quality level of this sample. Then, a single numerical value can be obtained for this sample by determining a representative value, according to one of the four transformation methods. Raz and Wang (1990b) called it "membership approach".

In the second case, representative value are obtained directly from linguistic terms associated with the observations. The representative value of a sample is calculated as the average of the representative values of the observations in sample. This approach is called by Raz and Wang (1990b) a "probabilistic approach" since the control limits are derived based on a probabilistic argument. Note that the conversion of once the linguistic observations into their representative values, is equivalent to the application of control charts for variables.

1) The membership approach

Suppose that there are t linguistic values in the term set, denoted by $L_i, i = 1, ..., t$, for each linguistic value. The corresponding fuzzy set is denoted by F_i, and described by the membership function $\mu_i(x_i)$, where x_i is a subset of the standard base variable. Now, if we have a sample S of n observations, then S can be described as

$$S = \{(F_1, k_1), (F_2, k_2), ..., (F_t, k_t)\}, \qquad (3)$$

where k_i is the number of observations assigned to the linguistic value L_i by the inspector, with $\sum_1^t k_i = n$. The fuzzy set that equal the mean of fuzzy sets in the sample is denoted by MF_s. The membership function of MF_s is denoted by $\mu_s(x_s)$. The fuzzy subset MF_j, corresponding to the mean of each sample j is obtained by the following arithmetic equation:

$$MF_j = (k_{1j}F_1 + k_{2j}F_2 + ... + k_{tj}F_t)/n. \qquad (4)$$

This is an operation on fuzzy subsets associated with linguistic terms which results also in a fuzzy subset. The membership function of MF_j is given by:

$$\mu_{MF_j}(x) = \underset{x=(k_{1j}x_1+....+k_{tj}x_t)/n}{MaxMin}[\mu_{1j}(F_1), ..., \mu_{tj}(F_t)]. \qquad (5)$$

The grand mean of the m initials available samples (GMF) is calculated as the average of the sample means:

$$GMF = \frac{\sum_{j=1}^{m} MF_j}{m}, \qquad (6)$$

which is also a fuzzy subset. Its membership function is defined as follows:

$$\mu_{GMF}(x) = \underset{x=(x_1+....+x_m)/m}{MaxMin}[\mu_{MF_1}(x_1), ..., \mu_{MF_m}(x_m)] \qquad (7)$$

The centre line, CL, is computed by transforming GMF into its representative value with one of the transformation methods. The mean deviation of the fuzzy subset GMF is calculated according to Kaufmann and Gupta (1985). Considering a fuzzy convex set A, and let the mode of its membership function be x_m. Let $x_l(\alpha)$ the left side of this membership function and $x_r(\alpha)$ the right one, where $x_l(\alpha)$ is a minimum of basic variable x such that the membership value is α, and $x_r(\alpha)$ is a maximum of basic variable x such that the membership value is α. Then $\sigma(A)$ is defined as follows:

$$\begin{aligned}\sigma_A &= \sigma_l + \sigma_r \\ &= \int_0^1 [x_m - x_l(\alpha)]d\alpha + \int_0^1 [x_r(\alpha) - x_m]d\alpha \\ &= \int_0^1 [x_r(\alpha) - x_l(\alpha)]d\alpha \end{aligned} \tag{8}$$

A multiplier k representing the number of mean deviations, $\sigma(GMF)$, that control limits are far from the centre line should be computed. Raz and Wang (1990a) proposed, for linguistic data, the use of Monte-Carlo simulation to find a value of k that yields a pre specified type I error probability. First, the empirical distribution of the linguistic terms is estimated from the initial samples. Then, an initial value for k is chosen arbitrarily, and control limits are calculated with σ(GMF). The empirical distribution is used to generate a large number of samples, and the fraction of the samples falling outside the control limits are calculated. If this fraction is different from the predefined type I error value, then the value of k is changed and the fraction is recalculated. Iteration is continued until we reach the exact value of the predefined type I error probability. The fuzzy control limits can then be expressed as follows:

$$\begin{aligned} LCL &= Max\{0, [CL - k\sigma(GMF)]\}, \\ UCL &= Min\{1, [CL + k\sigma(GMF)]\}. \end{aligned} \tag{9}$$

Sample points and control limits must be in the range of $[0, 1]$, because the representative value are drawn from a standardized base variable.

2) The probabilistic approach

An alternative to the membership approach is the probabilistic one. Fuzzy subsets F_i associated with the linguistic terms L_i are transformed into their respective representative values r_i with one of the previously described transformation methods. The sample mean M_j is calculated as the average of the sample linguistic representative values, r_i, according to the following formula:

$$M_j = (k_{1j}r_1 + k_{2j}r_2 + ... + k_{tj}r_t)/n. \tag{10}$$

For each sample j, the standard deviation SD_j is calculated as the standard deviation of the representative values of the observations in the sample:

$$SD_j = (\tfrac{1}{n-1}\textstyle\sum_{i=1}^{t} k_{ij}(r_i - M_j)^2)^{1/2}. \tag{11}$$

The centre line is calculated as the grand mean of the sample means M_j as follows:

$$CL = \frac{\sum_{j=1}^{m} M_j}{m} = \frac{\sum_{j=1}^{m}\sum_{i=1}^{t} r_i k_i j}{mn}. \tag{12}$$

The mean sample standard deviation, MSD, is calculated as the average of the standard deviation of the m samples available. Then, we have

$$MSD = \tfrac{1}{m}\textstyle\sum_{j=1}^{m} SD_j. \tag{13}$$

The control limits are determined with the formulae of the control chart by variables. The points plotted on the chart are the means of representative values, which are defined on the base variable, standardized to [0,1]. Then, control limits are given by:

$$\begin{aligned} LCL &= Max\{0, (CL - A_3MSD)\} \\ UCL &= Min\{1, (CL + A_3MSD)\}\ . \end{aligned} \tag{14}$$

$$A_3 = \tfrac{3}{c_4 n^{1/2}}, \text{ with } c_4 = (\tfrac{2}{n-1})^{1/2}\frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})}. \tag{15}$$

III. NUMERICAL EXAMPLE

Tunisie Porcelaine is a company specialized in the porcelain product (cups, plates,...). Items are classified by experts, with respect to quality, into four categories. There are no instruments used for this classification. When a product presents no default, or an invisible minor default, it is classified as standard product (SD). When it presents a visible minor default that does not affect the use of product, then it is classified as second choice (SC). If, there is a visible major default that does not affect the product use, then it is classified as third choice (TC). Finally when the use is affected the item is considered as chipped (CP). Table 1 shows data for thirty samples with different sample sizes taken every half an hour.

A. Marcucci approach

The porcelain example corresponds to the case II problem where the goodness-of-fit test is not applicable. In this example the target value is estimated for the base period where the process is assumed to be in control. Suppose that the process is in control in the period corresponding to sample 18. Then, we can estimate the sample proportions for the base period as follows:

$p_{i1} = 65\%$, $p_{i2} = 23\%$, $p_{i3} = 7\%$, $p_{i4} = 4\%$.

TABLE I

Data of the porcelain process.

Sample	SD	SC	TC	CP	Size
1	144	46	12	5	207
2	142	50	9	5	206
3	142	35	16	6	199
4	130	70	19	10	229
5	126	60	15	10	211
6	112	47	9	8	176
7	151	28	22	9	210
8	127	43	45	30	245
9	102	79	20	3	204
10	137	64	24	5	230
11	147	59	16	6	228
12	146	30	6	6	188
13	135	51	16	8	210
14	186	82	23	7	298
15	183	53	11	9	256
16	137	65	26	4	232
17	140	70	10	3	223
18	135	48	15	9	207
19	122	52	23	10	207
20	109	42	28	9	188
21	140	31	9	4	184
22	130	22	3	8	163
23	126	29	11	8	174
24	90	23	16	2	131
25	80	29	19	8	136
26	138	55	12	12	217
27	121	35	18	10	184
28	140	35	15	6	196
29	110	15	9	1	135
30	112	37	28	11	188

The statistic to be plotted in the control chart for each sample is:

$$Z_i^2 = n_i n_0 \sum_{j=1}^{3} \frac{(p_{ij} - p_{0j})^2}{X_{ij} + X_{0j}}, \qquad (16)$$

where $p_{ij} = X_{ij}/n_i$ and X_{ij} is the number of observations of the sample i classified as j products.

The resulting generalized p_i chart is illustrated in figure 1. The upper control limit is taken to be the 95th percentile of the $\chi^2(3)$ distribution which is 7.815. On four occasions the process is deemed to be out of control. If the quality of the production process is improved, the control chart shown in figure 1 is unable to detect the change in the quality level.

B. Wang and Raz approaches

In this part, the two approaches presented by Raz and Wang (1990b) are applied to the porcelain process.

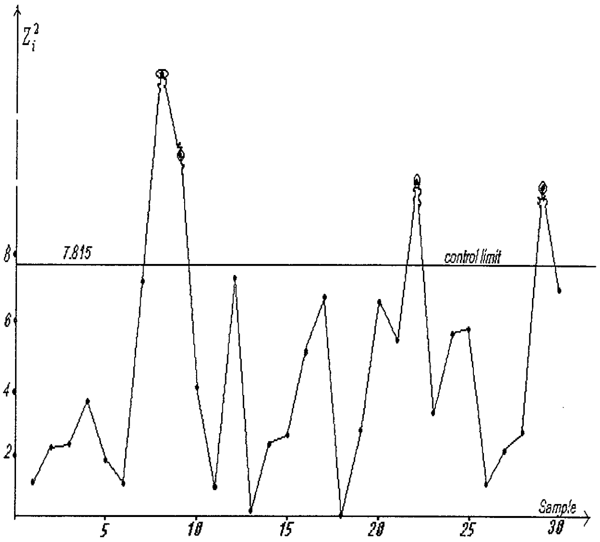


Fig. 1. Generalized p-chart for the Porcelain Process.

The term set consists of 4 terms: SD:Standard, SC:Second Choice, TC:Third Choice, CP:chipped. Each term is associated with a fuzzy subset and is described by a membership function. Currently, there is no theoretical evidence supporting any method for constructing these membership functions. To determine which importance has the constructing method on the control chart, three sets of membership functions are presented. Raz and Wang (1990b) showed that there is no theoretical evidence supporting any method for transforming a fuzzy subset into its representative value. In the following, fuzzy mode and fuzzy median are used as transformation methods. Representative value of the four fuzzy subset are given in table 2.

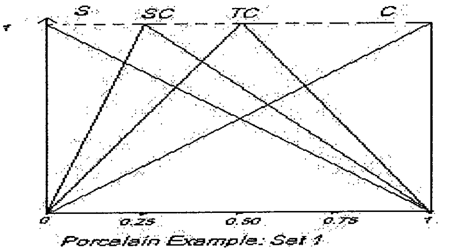
TABLE II

Representative values of linguistic terms.

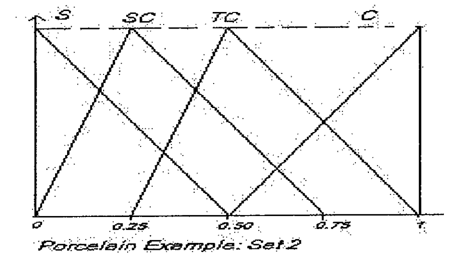
Set	1	2	3	1	2	3
SD	0	0	0	0.293	0.143	0.073
SC	0.25	0.25	0.25	0.387	0.317	0.25
TC	0.5	0.5	0.5	0.5	0.57	0.5
CP	1	1	1	0.5	0.854	0.75

1) Probabilistic Approach

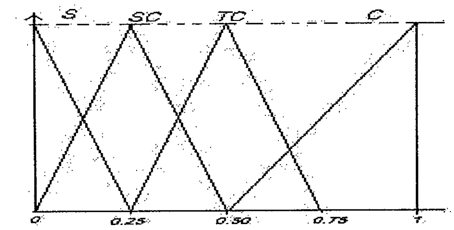
From table 2 we conclude that the shape of membership function has no effect on the representative value of fuzzy subset when fuzzy mode method is used. But when fuzzy median is used representative value of fuzzy subset changes when the shape of the membership function changes. Figure 2 illustrates these three sets of membership functions. Control charts produced by these different sets are the same when



Porcelain Example: Set 1



Porcelain Example: Set 2



Porcelain Example: Set 3

Fig. 2. Sets of Membership functions for the Porcelain Data, with term set: SD: Standard, SC: Second Choice, TC: Third Choice and CP: Chipped.

TABLE III
Results of applying Fuzzy probabilistic approach (Fuzzy mode)

Sample	M_j	SD_j	UCL	LCL
1	0:109	0:20	0:184	0:088
2	0:107	0:20	0:164	0:088
3	0:114	0:22	0:183	0:085
4	0:162	0:24	0:178	0:091
5	0:154	0:24	0:182	0:089
6	0:138	0:24	0:181	0:084
7	0:129	0:25	0:179	0:089
8	0:258	0:34	0:179	0:092
9	0:161	0:19	0:180	0:088
10	0:143	0:21	0:174	0:091
11	0:126	0:21	0:178	0:090
12	0:088	0:21	0:179	0:086
13	0:137	0:23	0:177	0:089
14	0:131	0:21	0:174	0:096
15	0:108	0:22	0:175	0:093
16	0:143	0:21	0:170	0:091
17	0:114	0:18	0:174	0:090
18	0:138	0:24	0:177	0:088
19	0:167	0:25	0:178	0:088
20	0:178	0:26	0:182	0:086
21	0:088	0:19	0:182	0:085
22	0:092	0:23	0:188	0:082
23	0:119	0:24	0:186	0:084
24	0:120	0:21	0:198	0:076
25	0:182	0:27	0:195	0:077
26	0:146	0:25	0:190	0:089
27	0:151	0:26	0:193	0:085
28	0:114	0:22	0:180	0:087
29	0:069	0:16	0:193	0:077
30	0:182	0:27	0:183	0:086
	Average	0. 136	0. 229	

fuzzy mode is used and are different when the fuzzy median is used.

For each sample j, sample mean M_j and the standard deviation SD_j, are determined. In the following, results of these values, their means, and the correspondant control limits, are given using the two methods for transforming fuzzy subsets in their representative values.

- Fuzzy Mode: Table 3 shows results of applying fuzzy mode. Control limits changes when the sample size changes. Only on two occasions the process is deemed to be out of control: samples 8 and 29. This is different from control chart for Marcucci approach which seems to be more sensitive. Samples 9 and 22 cannot be detected by control chart given here.

- Fuzzy median: Representative values obtained are different for each set, then the result is one control chart for each set. From table 4 we conclude that the shape of the membership fuctions, which describe fuzzy subsets, affect the sensitivity of control charts.

When set1 is used, the resulting control chart shows that on five occasions the process is deemed to be out of control, which are samples 8, 12, 20, 22 and 29. By using set2 and set3, the resulting control charts show only sample 8 and 29 are out of control. Then, if the degree of fuzziness of the fuzzy subsets, associated with the linguistic terms, is increased then control chart becomes more sensitive.

Results obtained, using fuzzy median transformations, are better than obtained by using fuzzy mode transformation.

2) Membership approach:

For each sample, membership function of the fuzzy subset corresponding to the mean of the sample observations is determined. The corresponding representative value is then calculated. The value k is used to

TABLE IV
Results of applying Fuzzy probabilistic approach (Fuzzy median)

set1

j	M_j	SD_j	UCL	LCL
8	0, 373*	0, 091	0, 352	0, 326
12	0, 321*	0, 058	0, 354	0, 324
20	0, 355*	0, 081	0, 354	0, 324
22	0, 320*	0, 058	0, 355	0, 323
29	0, 319*	0, 059	0, 357	0, 321

set2

j	M_j	SD_j	UCL	LCL
8	0, 339*	0, 250	0, 278	0, 212
12	0, 207	0, 151	0, 283	0, 207
20	0, 280	0, 199	0, 283	0, 207
22	0, 209	0, 167	0, 285	0, 205
29	0, 196*	0, 129	0, 289	0, 201

set3

j	M_j	SD_j	UCL	LCL
8	0, 278*	0, 268	0, 213	0, 143
12	0, 140	0, 161	0, 218	0, 138
20	0, 214	0, 209	0, 218	0, 138
22	0, 143	0, 181	0, 221	0, 135
29	0, 127*	0, 132	0, 225	0, 131

*Bold face numbers are out of control.

yield a prespecified probability of type I error. By fixing this probability at 0:0027 as in the traditionnal control chart, the value of k is then chosen sothat the fraction of sample point falling outside the control limits is equal to this probability. By applying membership approach to the porcelain example, only samples 8 and 29 shows an out of control state. The result of the use of the fuzzy mode transformation is a same sample representative values for every set of membership functions and then a same control limits, figure 3. The use of fuzzy median results on a different sample representative values and then a different control limits.

IV. Comparison Between Marcucci and Wang and Raz Approaches.

The comparison between these approaches is based on two points. First, the determination of the samples under control on each control chart and, second, the analysis of the ARL on each case. Marcucci's generalized p_j chart is compared to Raz and Wang chart because the two type were designed to monitor deviations in any category.

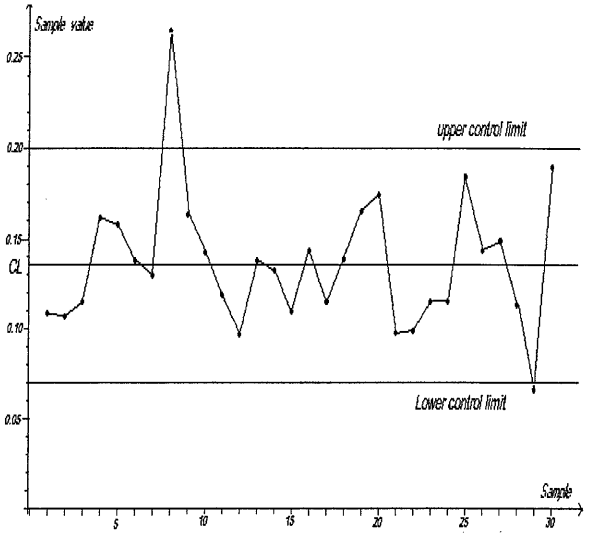


Fig. 3. Fuzzy Membership Control Chart (fuzzy mode).

A. Samples Under Control

In the Marcucci approach the out of control states are given by samples, 8; 9; 20 and 29. The out of control states in the Raz and Wang approach for all sets, are given by samples 8 and 29 when the membership approach is used with both fuzzy mode or fuzzy median transformation. When fuzzy probabilistic approach is selected the same result is obtained with fuzzy mode transformation. However, when fuzzy median is used as a transformation method the result changes when set 1 is selected. For a combination of fuzzy median transformation method, probabilistic approach and set 1, control chart shows that process is deemed to be out of control on five occasion, Samples 8,12,20,22 and 29. Sample 9 judged to be out of control by Marcucci approach, plot on the upper control limit in the fuzzy probabilistic approaches.

Then, at least in this example, some conclusions can be made:

* The degree of fuzziness of the membership functions associated with linguistic terms are significant variables in the construction of control charts.

* Fuzzy theory performs better than probability theory in monitoring multinomial process (porcelain), when probabilistic approach, fuzzy median transformation and set 1 are combined.

B. Average Run Length

The detection capability of control charts is given by the probability of detecting the shift based on the first sample taken after the shift has occured. Often the complement of that probability, which is the probability of a type II error, is used. This probability can be calculated with normal distribution for the linguistic chart and the noncentral chi-squared distribution for the generalized p-chart.

Suppose that process mean shifts from the in control value to another value (representative value of sample 29). table 5 shows all values of the ARL when the process shifts to a situation like given by sample 29. The ARL when the Marcucci's approach is used, is 2.48.

TABLE V

AVERAGE RUN LENGTH FOR CONTROL CHART APPROACHES

	Prob. Approach			Memb. Approach		
Set	1	2	3	1	2	3
F.mod	2.81	2.81	2.81	12.5	2.59	1.064
F.med	1.09	1.61	1.59	12.5	7.14	1.48

By analysing table 5, some conclusion can be given:
* The use of fuzzy mode results on a little ARL, when the membership approach is selected.
* Fuzzy median transformation leads to a little ARL, when the probabilistic approach is selected.
* Probabilistic approach leads to a more competetive ARL than membership approach, when Set 1, and fuzzy mode are used.
* The smallest value of the ARL is obtained when the membership approach, fuzzy mode and Set 3 are combined.

By setting the two key parameters: ARL and samples under control, the conclusion to be given is that fuzzy control charts leads to a better results than the generalized $p-$chart if the membership functions and the transformation method are precisely selected.

V. CONCLUSION

Raz and Wang (1990b) concluded that transformation method used to obtain the representative value and the degree of fuzziness does not affect the performance of control chart. In this paper we showed that for the porcelain example the performance of fuzzy control chart is affected by the degree of fuzziness and the transformation method.

We conclude, for the porcelain example, that fuzzy control charts perform better and are more sensitive than probabilistic charts when a combinaison of fuzzy probabilistic approach, fuzzy median and (set 1) of membership functions is used.

Future studies need to determine which shape of membership functions must be used for linguistic terms, and the appropriate degree of fuzziness for these membership functions. Exact relationship between degree of fuzziness and sensitivity of control charts needs to be investigated.

REFERENCES

[1] Kanagawa, A., Tamaki, F., and Ohta, H., "Fuzzy Control Charts for Linguistic Data," *Proc. of International Fuzzy Engineering Symposium'91.*, vol. 2, pp. 644-654, 1991.

[2] Kanagawa, A., Tamaki, F., and Ohta, H., "Control Charts for Process Average and Variability Based on linguistic Data," *International Journal of Production Research.*, vol. 2, pp. 913-922, 1993.

[3] Marrucci,M., "Monitoring Multinomial Processes," *Journal of Quality Technology.*, vol. 17, pp. 86-91, 1985.

[4] Laviolette, M., Seaman, J. W., Barrett, J. D., and Woodall, W. H., "A Probabilistic and Statistical View of Fuzzy Methods, with discussion," *Technometrics.*, vol. 37, pp. 249-292, 1995.

[5] Saaty, T. L., "Measuring the Fuzziness of Sets," *Journal of Cybernetics.*, vol. 4, pp. 53-61, 1975.

[6] Raz, T., and Wang, J., "Probabilistic and Membership Approaches in the Construction of Control Charts for Linguistic Data," *Production Planning and Control.*, vol. 1, pp. 147-157, 1990b.

[7] Raz, T., and Wang, J., "On the Construction of Control Charts Using Linguistic Variables," *International Journal of Production Research.*, vol. 28, pp. 477-487, 1990a.

[8] Raz, T., and Wang, J., "Applying Fuzzy Set Theory in the Development of Quality Control Charts," *International Industrial Engineering Conference Proceedings,* pp.30-36, 1988.

[9] Duncan, A. J., *Quality Control and Industrial Statistics,* Richard D. Irwin, Homewood,IL, 4th ed., 1974.

[10] Kaufmann, A., and Gupta, M. M., *Introduction to Fuzzy Arithmetic,* Van Nostrand Reinhold, New York., 1985.