ISSN: 1945-3116 Vol. 3, No. 7, July 2010

Scientific
Research

Journal of Software Engineering and Applications

Chief Editor : Dr. Ruben Prieto-Diaz

ISSN: 1945-3116

9 771945 311001 07

www.scirp.org/journal/jsea

Journal Editorial Board

J. Software Engineering & Applications, **2010, 3, 629-735**
Published Online July 2010 in SciRes (http://www.SciRP.org/journal/jsea/)

TABLE OF CONTENTS

Volume 3 Number 7 July 2010

Journal of Software Engineering and Applications (JSEA)

Journal Information

SUBSCRIPTIONS

The *Journal of Software Engineering and Applications* (Online at Scientific Research Publishing, www.SciRP.org) is published monthly by Scientific Research Publishing, Inc., USA.

Subscription rates:

Print: $50 per issue.
To subscribe, please contact Journals Subscriptions Department, E-mail: sub@scirp.org

SERVICES

Advertisements
Advertisement Sales Department, E-mail: service@scirp.org

Reprints (minimum quantity 100 copies)
Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.
E-mail: sub@scirp.org

COPYRIGHT

Statements and opinions expressed in the articles and communications are those of the individual contributors and not the statements and opinion of Scientific Research Publishing, Inc. We assumes no responsibility or liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained herein. We expressly disclaim any implied warranties of merchantability or fitness for a particular purpose. If expert assistance is required, the services of a competent professional person should be sought.

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:
E-mail: jsea@scirp.org

J. Software Engineering & Applications, **2010, 3, 629-643**
doi:10.4236/jsea.2010.37073 Published Online July 2010 (http://www.SciRP.org/journal/jsea)

Using an Ontology to Help Reason about the Information Content of Data

Shuang Zhu[1], Junkang Feng[2]

[1,2]Database Research Group, School of Computing, University of the West of Scotland, Paisley, UK; [2]Business College, Beijing Union University, Beijing, China.
Email: {shuang.zhu, junkang.Feng}@uws.ac.uk

Received March 25th, 2010; revised April 23rd, 2010; accepted April 25th, 2010.

ABSTRACT

We explore how an ontology may be used with a database to support reasoning about the "information content" of data whereby to reveal hidden information that would otherwise not derivable by using conventional database query languages. Our basic ideas rest with "ontology" and the notions of "information content". A public ontology, if available, would be the best choice for reliable domain knowledge. To enable an ontology to work with a database would involve, among others, certain mechanism thereby the two systems can form a coherent whole. This is achieved by means of the notion of "information content inclusion relation", IIR for short. We present what an IIR is, and how IIR can be identified from both an ontology and a database, and then reasoning about them.

Keywords: *Ontology, Information Content of Data*

1. Introduction

Data mining techniques and tools are developed for finding otherwise hidden knowledge from data, and little seems to have been done on bringing "standard" domain knowledge into such a process, which we envisage would be helpful.

Ontologies as domain knowledge have been used in many fields. We want to explore how an ontology may help find hidden information from data. In this paper, the focus is on how to link an ontology with a relation database in order to reason about informational relationships between data constructs in the database and those between domain objects captured by an ontology. This may represent an innovative approach to knowledge discovery in a database.

Ontology [1] as a term used in computer science was started in the 1990's. Compared with the development of relational databases, it is a new scientific field. Ontology offers an opportunity to give an open and standardized description of database semantics with which we can substantially improve the quality and utilization of data. That is,

Ontology + Database = (Standards + Explicit Semantics) + Database,

which leads to improved data utilization and data quality [2].

Futhermore, *semantic web* [3] is a popular topic. Through semantic web we attempt to provide users with far better machine assistance than it is available now for their queries. Semantically annotated web pages with ontologies may assist reachers to achieve this purpose [4].

Through our work, we obtain an ontology from the DAML library, which represents some additional common knowledge, and link it with an existing database. In terms of linking an ontology and a database though, in the literature, we find a few different methods in using an ontology to assist a query process.

It appears that one way to achieve this is that an ontology is invoked at the very beginning of a query process [5] as shown in **Figure 1**. That is, it is through re-writing a query in order to get more information. A

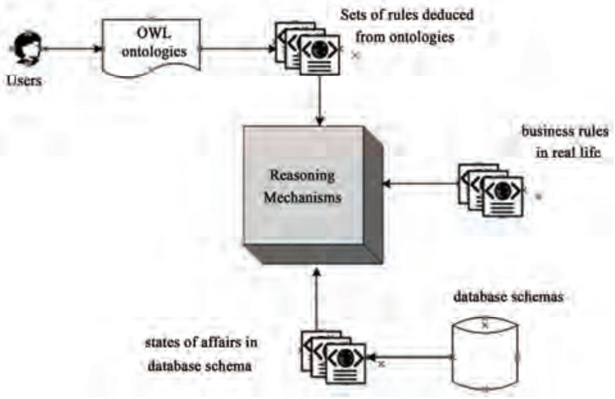

Users
OWL ontologies
Sets of rules deduced from ontologies
Reasoning Mechanisms
business rules in real life
database schemas
states of affairs in database schema

Figure 1. Invoking an ontology in the query processing

user query is translated into a set of queries with the help of the ontology, which better fits the structure of the data source. After query optimization strategies having been applied on them, the resultant transformed queries are equivalent to the submitted ones. Although seemingly a promising approach, it is not concerned explicitly with the information content of data, in which we are particularly interested and wish to explore and make use of.

Another approach that we have investigated is where an ontology is invoked in formulating a query process by Munir *et al*. [6]. In their approach, firstly, an ontology is generated based upon domain metadata including relationships between data in a relational database. Then such an ontology is enriched with domain knowledge. Secondly, ontology statements are translated into expressions in the OWL-DL language. Thirdly, the expressions are transformed into relational query statements. Finally, map the domain ontology to a relational database (as shown in **Figure 2**). Munir *et al.* [6] said little about the mapping between the created *ad hoc* ontology and the "standard" domain ontology if any, which we suspect is done intuitively. This is however one of the topics in which we are particularly interested.

We give an outline of our approach in Section 2. The key notion is informational relationship and its formalisation IIR [7]. We describe in details how IIR may be derived from a relational database and from an ontology in Section 3, which make use of inherent and *ad hoc* constraints between data constructs in a database and between concepts in an ontology. We present a full account on how our ideas are tested by using some implementation in Oracle in Section 4. Finally we give concluding remarks in Section 5.

2. Outline of our Approach

Our approach is to invoke an ontology when we work on a database. Namely, when a user submits a query, we do not change the query, but rather we involve the ontology in the reasoning process *per se* that is required for answering the query (shown in **Figure 3**). Furthermore and most importantly for us, the reasoning is carried out on the basis of the notion of "information content" of data. This notion is the work of Xu, Feng and Crowe in 2008 [8], which extends substantially Dretske's [9] definition of "information content" of a signal. In this paper, they introduce another notion called IIR, as a formulation of the notion of "information content" of data.

Xu *et al*. [8] define IIR as follows: "Let X and Y be an event respectively, there exists an IIR, from X to Y, if every possible particular of Y is in the information content of at least one particular of X". Furthermore, they define that "Let X be a event, the *information content* of X, denoted I(X), is the set of events with each of which X has an information content inclusion relation". Moreover, they present a sound and complete set of inference rules (IIR rules) for reasoning about information content of data (states of affairs, or events in general). The six inference rules are cited below.

1) **Sum**

If $Y = X_1 \cup X_2 \cdots \cup X_n$ then $I(X_i) \ni Y$ for $i = 1, \ldots, n$

This rule says if it is the disjunction of a number of events, then an event *X* is in the information content of any of the latter. A trivial case is where *X* and *Y* above are not distinct.

2) **Product**

If $X = X_1 \cap X_2 \cdots \cap X_n, Y = X_i$ for $i = 1, \ldots, n$ then $I(X) \ni Y$

This rule says that if an event *X* is the conjunction of a number of events, then any of the latter is in the information content of the former. A trivial case is where *X* and *Y* above are not distinct.

3) **Transitivity**

If $I(X) \ni Y, I(Y) \ni Z$ then $I(X) \ni Z$

This rule says that if the information content of an event *X* includes another event *Y*, and the information

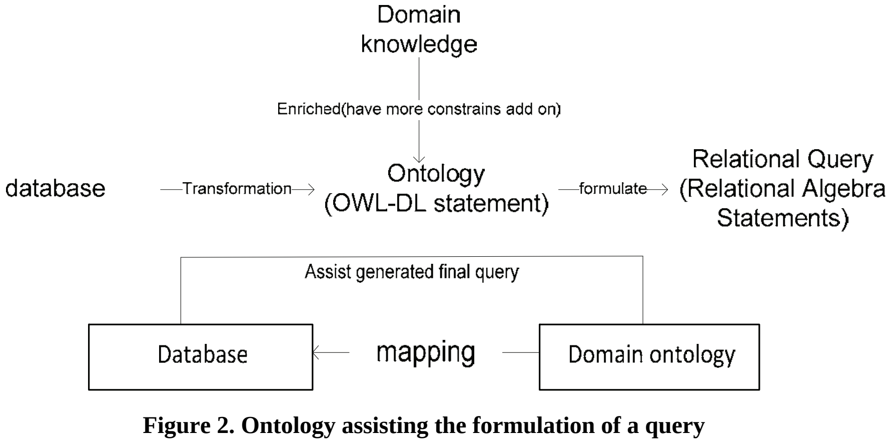

Domain knowledge
Enriched(have more constrains add on)
database
Transformation
Ontology (OWL-DL statement)
formulate
Relational Query (Relational Algebra Statements)
Assist generated final query
Database
mapping
Domain ontology

Figure 2. Ontology assisting the formulation of a query

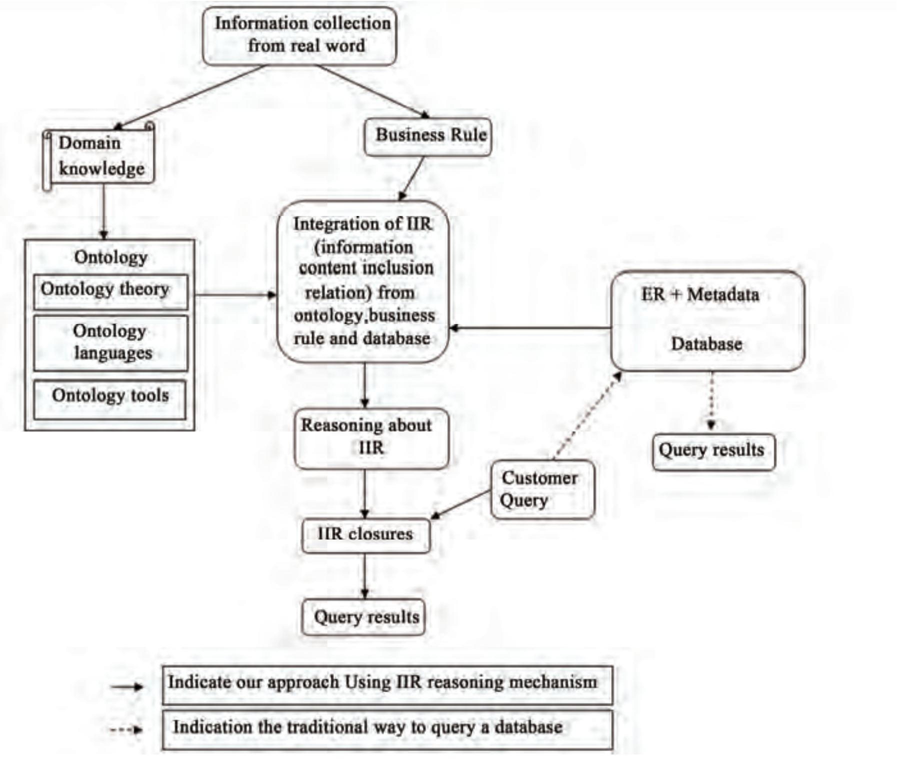

Information collection from real word
Domain knowledge
Business Rule
Ontology
Ontology theory
Ontology languages
Ontology tools
Integration of IIR (information content inclusion relation) from ontology,business rule and database
ER + Metadata
Database
Reasoning about IIR
Customer Query
Query results
IIR closures
Query results
Indicate our approach Using IIR reasoning mechanism
Indication the traditional way to query a database

Figure 3. Ontology enhances reasoning about the information content in a database

content of *Y* includes yet another event *Z*, then the information content of *X* includes *Z*.

4) **Union**

If $I(X) \ni Y, I(X) \ni Z$, then $I(X) \ni Y \cup Z$

This rule says that if the information content of an event *X* includes another two events *Y* and *Z* respectively, then the information content of *X* includes event *Y*∩*Z* that is the product of *Y* and *Z*. And it is in this sense that *Y* and *Z* are in a "union".

5) **Augmentation**

If $W = W_1 \cap W_2 \cdots \cap W_n$, *Z* is the product of a subset of $\{W_1, W_2, \cdots, W_n\}$ then $I(W \cap X) \ni Z \cap Y$

This rule says that if $W_1 \cap W_2 \cdots \cap W_n$, event *Z* is the product of a subset of $\{W_1, W_2, \cdots, W_n\}$, and the information content of event *X* includes event *Y*, then the information content of the event *W*∩*X* formed by the product of *W* and *X* includes the event *Z*∩*Y* formed by the product of *Z* and *Y*.

6) **Decomposition**

If $I(X) \ni Y \cap Z$ then $I(X) \ni Y, I(X) \ni Z$

This rule says that if the information content of event *X* includes event *Y*∩*Z* that is the product of event *Y* and event *Z*, then *Y* and *Z*, as separate events, are in the information content of *X*, respectively.

In this paper, we exploit the ideas above. That is, in a way, we translate both the ontology and the database into IIR and then reason about them as a whole. Put another way, as what matters is information and IIR captures and formulates it, so we look at both an ontology and a database from the same perspective of IIR, and this enables the two different things to work together. The overview of our approach is illustrated in **Figure 3**.

On the very top of **Figure 3**, there is a block called "information collection from the real world". From this information, knowledge about a domain of interest including explicit business rules is arrived at. Domain knowledge is then formulated as an ontology by using software tools and languages.

Two different routes are there to deal with user queries. If it is in a conventional query language then a query is handled in a normal way. The dotted line indicates this route. If that does not work, we would invoke the other route, *i.e.*, to invoke ontology and reasoning about IIR. The second solution is the primary goal of this project, which is indicated by the solid line arrow in **Figure 3** of "Customer query"→"IIR closures"→"Query results".

The only difference between these two solutions lies in the middle part of the procedure, on which we concentrate. Within the "Integration of IIR" section, there are three different resources required to derive the "IIR", indicated by three arrows from "ontology", "business rule" and "database", which are the origins of initial IIR. Then there is a reasoning mechanism implemented in PL/SQL of Oracle. The result of the reasoning is *IIR* *closures*. Given an event A, the *IIR closure* of A, denoted as A^+, is the set of all events that are in the information content of A, that is, if IIR(A, B), then B $\in A^+$. *IIR closures* are the basis of answering queries in our approach. Our work thus far shows that it is the *additional relationships between data constructs* especially "entities" that are revealed and made available through using an ontology that give us more and enlarged *IIR closures* than those that would otherwise be based on the database alone. This is how our approach makes a difference.

One of the main tasks is to derive IIR from the ontology, the database and business rule, and then integrate them as a whole. For instance, suppose that we have IIR(A, B) (meaning the information content of A includes B) and IIR(C, D) from a relational database, and IIR(E, F) and IIR(G, H) from an ontology. If we also know that A and E are equivalent, then with Transitivity, we get IIR(E, B) and IIR(A, F). Consequently A^+ and E^+ are enlarged.

We use Oracle [10] to implement this approach. An ontology in OWL [11] can be translated into relational tables [12]. Such tables do not hold data values however, if the ontology is an unpopulated one. In such a case, the involvement of an ontology results in additional objects and additional relationships between objects that are represented by data in the original relational database. This way, a query that does not have exact match with data in the database may be answered. An ontology may add an additional hierarchical structure to data in the database. Furthermore, as said earlier, we use ontologies in a special type of reasoning, *i.e.*, reasoning about the *information content* of data through a kind of special relationship between data items and between data items and real world objects, namely *informational relationships*, which is captured and formulated as IIR between *events* (in terms of probability theory). Thus, how to identify IIR from an ontology becomes a key factor in our approach.

3. Deriving IIR

An IIR is a relationship between two *states of affairs* (*i.e.*, *event*s) such that one's existence results in the certainty that the other exists, and without the former, the latter is not certain. Following Dretske 81 [9], we say that the latter is in the "information content" of the former.

It would appear that to express IIR(X, Y) must be based on and revolved around two elements. One is two individual values (two individual parts or two sets of groups) captured as X and Y, and the other is relationships between X and Y.

We use part of a "university" database and part of ontology "Academic" to present how IIR can be derived from a database and an ontology. Then the IIR are reasoned about by applying aforementioned Inference Rules. The reasoning is implemented by a program.

3.1 Deriving IIR from an Ontology

According to characteristics of ontologies, these are two different sources that may help the derivation of IIR. One is concerned with relationships between "Classes" in an ontology. The other is "ObjectProperty".

3.1.1 IIR Derived from Classes

Generally, there are two different types of relationships between classes from which IIR exist. One is "subClassOf", and the other "equivalentClass". The syntax for these two in an OWL ontology is as follows:

- A relationship between "Class" and "subClassOf",

```
<owl:Class rdf:ID="Lecturer">
    <rdfs:subClassOf rdf:resource="# Faculty">
</owl:Class>
```

- A relationship between "Class" and "equivalentClass",

```
<owl:Class rdf:ID="Teachers">
<owl:equivalentClass rdf:resource="#Faculty"/>
</owl:Class>
```

The IIR could be derived from these two relationships thusly:

- IIR(Class, subClassOf),
- IIR(Class, equivalentClass) and IIR (equivalentClass, Class).

As shown above, we have a relationship "Lecturer is a subclass of Faculty, and Teachers is an equivalent class to Faculty". Hence we have IIR(Lecturer, Faculty), IIR(Teacher, Faculty), IIR(Faculty, Teacher), and IIR(Lecturer, Teacher).

3.1.2 IIR Derived from ObjectProperty

There are four different types of ObjectProperty relationships, which capture relationships between classes in an OWL ontology. These are: "ObjectProperty", "subPropertyOf", "equivalentProperty" and "inverseOf".

As aforementioned, to create IIR needs two classes (X and Y) from the ontology. As ObjectProperty represents a relation for connecting two classes of "domain" and "range" in an OWL ontology, an ObjectProperty already contains a set of classes, which can be expressed as "ObjectProperty=('domain', 'range')". Accordingly, we obtain IIR(domain, range). That is, the IIR that can be derived from these four types of ObjectProperty is all of the form:

- IIR(domain, range)

Note that IIR must be of a many-to-one relationship (including one-to-one). How to handle many-to-many is to be addressed shortly.

The relevant syntax of OWL is as follows:

- A relationship between "ObjectProperty",

```
<owl:ObjectProperty rdf:ID="research_by ">
<rdfs:domain rdf:resource="Professors"/>
<rdfs:range rdf:resource="Projects"/>
</owl:ObjectProperty>
```

Then we have “IIR(domain, rang)”, for example, IIR(Professors, Projects).

- A relationship between “ObjectProperty” and “subPropertyOf”,

```
<owl:ObjectProperty rdf:ID=”research_in”>
<rdfs:domain rdf:resource=”Postgraduates”/>
<rdfs:range rdf:resource=”Projects”/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID=”study_in”>
<rdfs:subPropertyOf rdf:resource=”research_in”/>
<rdfs:domain rdf:resource=”Postgraduates”/>
<rdfs:range rdf:resource=”Projects”/>
</owl:ObjectProperty>
```

Then we have “IIR(domain, range)”, for example, IIR(Postgraduates, Projects).

- A relationship between “ObjectProperty” and “equivalentProperty”,

```
<owl:ObjectProperty rdf:ID=”attend_course”>
<rdfs:domain rdf:resource=”Student”/>
<rdfs:range rdf:resource=”Course”/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID=”join_course ”>
        <owl:equivalentProperty
rdf:resource=”attend_course”/>
<rdfs:domain rdf:resource=”Students”/>
<rdfs:range rdf:resource=”Course”/>
</owl:ObjectProperty>
```

Then we have “IIR(domain, range)”’, for example, IIR(Students, Course).

- A relationship between “ObjectProperty” and “inverseOf”,

```
<owl:ObjectProperty rdf:ID=”teache_of”>
<rdfs:domain rdf:resource=”Faculty”/>
<rdfs:range rdf:resource=”Course”/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID=”instruct_by”>
<owl:inverseOf rdf:resource=”teaches_of”/>
<rdfs:domain rdf:resource=”Course”/>
<rdfs:range rdf:resource=”Faculty”/>
</owl:ObjectProperty>
```

Then we have “IIR(domain, rang)”, for example, IIR(Course, Faculty).

Moreover, there are relationships between classes and “Not Null” FunctionalProperty, the syntax of which is as follows:

- A relationship between “Class” and “FunctionalProperty”,

```
<owl:Class rdf:ID=”Course”>
    <owl:DatatypeProperty rdf:ID=”courseNo”>
    <rdfs:type
rdf:resource=”&owl;FunctionalProperty”/>
<!-- NOT NULL -->
<rdfs:domain rdf:resource=”Course”/>
          <rdfs:range rdf:resource=”&xsd;short”/>
    </owl:DatatypeProperty>
```

Then we have “IIR(Class, DatatypeProperty)”, for example, IIR(Course, courseNo).

Furthermore, to handle a many-to-many relationship, we transform it into two many-to-one relationships. Consider firstly this scenario: “one course is taken by more than one students and one student takes more than one course”. This is a many-to-many relationship. We decompose such a relationship into two many-to-one by creating a new class and then they are treated in the same way as the second method for the ObjectProperty transformation. We create an intermediate table and use the ObjectProperty name as the new class name (as well as the table name which will be the transformation of this class). In details, the relationship “StudentTakeCourse” between the class “student” and the class “course” is many-to-many. We create a new class CourseLearning, which contains two ObjectProperty relationships as shown below:

Class (Student)

DatatypeProperty (studentNo domain (Student) range (xsd: short) Functional)

DatatypeProperty (studentName domain (Student) range (xsd: string))

DatatypeProperty (major domain (Student) range (xsd: string))

DatatypeProperty (enrollmentDate domain (Student) range (xsd: date))

Class (Course)

DatatypeProperty (courseNo domain Course) range (xsd: short) Functional)

DatatypeProperty (courseName domain (Course) range (xsd: string))

DatatypeProperty (creditHour domain (Course) range (xsd: integer))

Class (CourseLearning)

ObjectProperty (takenBy domain (CourseLearning) range (Student))

ObjectProperty (inv - takenBy domain (Student) range (CourseLearning) inverseOf (takenBy))

ObjectProperty (takesCourse domain (CourseLearning) range (Course))

ObjectProperty (inv - takesCourse domain (Course) range (CourseLearning) inverseOf (takesCourse))

Accordingly the IIR obtained in this process are IIR(CourseLearning, Student) and IIR(CourseLearning, Course) (**Figure 4**).

The paragraphs that follow illustrate details at the “instances” (data values) level of the above example.

The original class CourseLearning is divided into two parts takenBy and takesCourse (as ObjectProperty), either of which only shows a one-way relationship. These combined however form the relationship between students and courses. **Table 1** shows some instances.

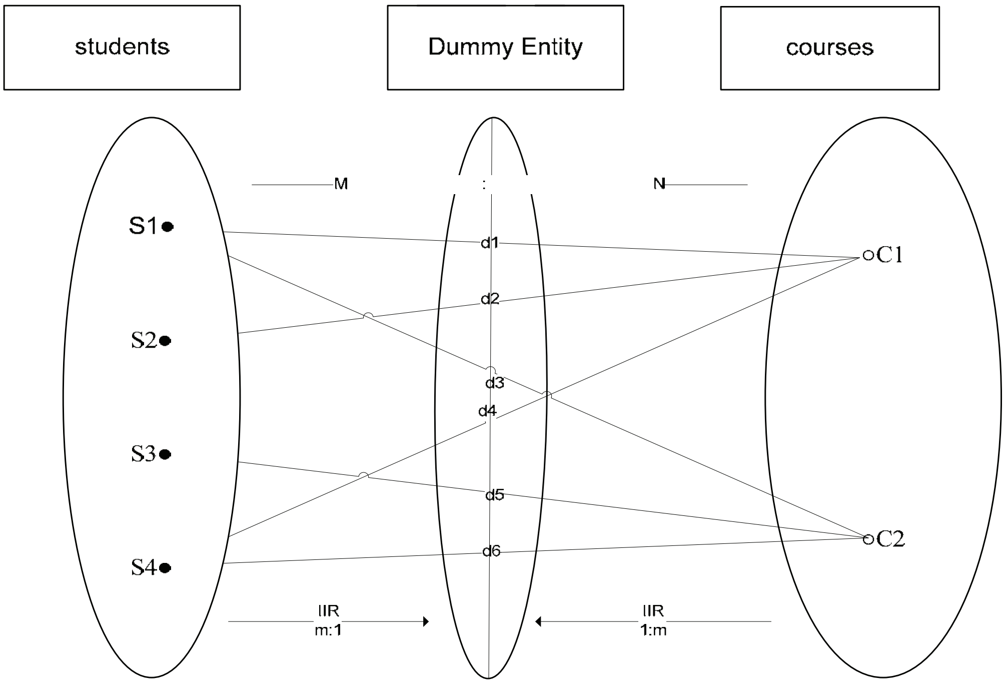


Figure 4. The overview for IIR relationship of CourseLearning

Table 1. Table transformation from CourseLearning

CourseLearning	Student	Course
d1	S1	C1
d2	S2	C1
d3	S1	C2
d4	S4	C1
d5	S3	C2
d6	S4	C2

3.2 Deriving IIR from a Database

In a database, initial IIR (*i.e.*, IIR that is not implied by others) come from three sources: relationships between tables, relationships between attributes and relationships between individual data values. Two different ways can be used to derive such IIR.

- A relationship between a “subclass” and a “super class”

Figure 5 shows part of a “university” database schema. An IIR exists between two tables if one is a super class of the other, for example, IIR(postgraduate, student).

Note that, IIR is a relationship between *event*s as we said earlier. For tables, we define events as follows: if we randomly chose a tuple from a database, that the tuple happens to be in a particular table is an event. Thus the above IIR(postgraduate, student) means that the existence of a tuple in table Postgraduate makes certain that a tuple that corresponds to the former exists in table Student.

- A many-to-one relationship between two tables

Similar to deriving IIR from an “ObjectProperty” with an ontology, we obtain IIR(table 1, table 2) if they have a many-to-one relationship for example, IIR(undergraduate, course).

- Constraints of the relational data model and business rules on data

A third source of IIR is *constraints* of a relational database and business rules on data, for example, IIR(table 1, PK) and ‘IIR(PK, attribute1). For **Figure 5**, we have IIR(courses, courseID) and IIR(courseID, courseName). The former means that the existence of a tuple of table Courses makes certain that a corresponding course ID (a value) exists. The latter means that the existence of a course ID makes certain that a corresponding course name exists.

Another type of IIR is IIR(FK1, PK1), for example:

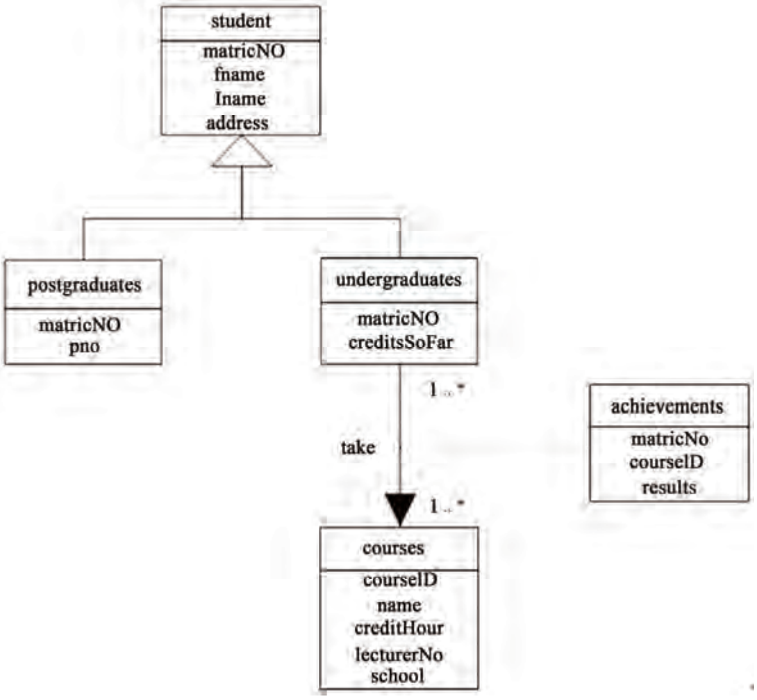


Figure 5. 6 partial EER diagram of "University" relational database

IIR(courseIDoftableStudents, courseIDoftableCourses), which means that the existence of a course ID in table Students makes certain that a corresponding course ID exists in table Courses.

A database is normally populated with data values. Inaddition to the above IIR on the "table" level and "attribute" level, there could be IIR identifiable at the "data value" level.

The information that each individual "data value" holds in a relational database comes from the semantics of the attributes to which the data value belongs. This is due to the capacity of a concept's "giving meaning to its instances" [9]. An attribute may be seen as representing a concept. For example, "student name" is seen as a concept. Relationships between entities in a database can be seen as "complex concepts" [9] and therefore also give meaning to data values that are instances of the relationships. That is, data in the relational database already hold relationships upon which there are constraints imposed.

We now use a simple example to summarise how IIR may be derived on the three levels. Suppose three tables shown in **Figure 6** in the "University" database.

Table level

According to **Figure 6**, table administration staff is a subclass of table staff, which gives the following IIR between these two tables:

IIR(administration_staff, staff)

As previously mentioned, the meaning of IIR indicates that first arguments existence results in the certainty that the other exists, and without the former, the latter is not certain. Therefore, the meaning of the two part relationship in this particular IIR, may be explained as: "if there is a member of administration staff then a corresponding member of staff must exist, otherwise the latter is not certain". In this particular case, the IIR is true because any member of administration staff is a member of staff.

Attribute level

If an attribute "A" is in a table which includes attributes "A", "B" and "C". Then, any combination of "A", "B" and "C" that includes "A" would have "A" in its information content. For example, IIR(A∩B, A), which means that if an instance, say (*a*, *b*), of "A∩B" exists, then there must be an corresponding instance of "A" existing - in this case, it is *a*. In general, this type of IIR is IIR("a set of attributes", "a subset of the attributes").

Using the values in **Figure 6**, an example is shown below. The attributes in table administration staff include sno, position, and deptNo. So the IIR are:

IIR(sno∩position∩deptNo, sno)
IIR(sno∩position∩deptNo, sno∩position)
IIR(sno∩position∩deptNo, sno∩ deptNo)
IIR(sno∩position∩deptNo, position)
IIR(sno∩position∩deptNo, position∩deptNo)
IIR(sno∩position∩deptNo, deptNo)

These IIR are derived on the attributes level, which may be seen as based on the aforementioned "Product" Rule, *i.e.*, if an event *X* is the conjunction of a number of events, then any of the latter is in the information content of the former.

Data value level

DB-staff

sno	fname	lname	sex	Address	tel	office
s02923	John	Key	M	6 Lawrence St, Glasgow	2384	E110
s02933	Julie	Lee	F	8 George St, Glasgow	2234	G203
s04885	Ann	White	F	18 Taylor St, Glasgow	5112	G133
s04995	Susan	Brand	F	28 High St, Paisley	3001	G229
s06465	Mary	Tregear	F	7 George St, Paisley	7754	F232
s06883	David	Ford	M	64 Well St, Paisley	8772	F231

DB-administration staff

sno	position	deptNo
s04885	secretary	d01
s04995	accountant	d03

DB-departments

deptID	departmentName
d01	administration
d03	finance

Figure 6. Three tables within the "University" relational database

Unlike the unpopulated ontology in use for this project, data values are a very important part in relational databases and it also the largest constituent of a relational database. Before explaining how to derive IIR on the data value level, let us re-cap the meaning of the terms we have been using, *i.e.*, "random variable" "event" and "particular of an event".

A *random variable* is an entity used mainly to describe chance and probability in a mathematical way. An *event* is a set of outcomes (a subset of the sample space) to which a probability is assigned. Typically, when the sample space is finite, any subset of the sample space is an event (*i.e.*, all elements of the power set of the sample space are defined as events) (A WorldViewer.com, 2009). Moreover, a specific event at a particular time and in a particular space is called a *particular* of an event. For example, consider the following situation. For an electric circuit, two random variables can be identified: one is "the condition of the lamp", and the other "the condition of the switch". There are two states about the lamp: "lit" and "unlit", and two for the switch: "closed" and "open". There are 2^2 events for either. Moreover, "unlit" at 10:30 am and "lit" at 10:30 pm, are two particulars.

Table 2. IIR between data values

Random variables	IIR
'sno', 'position'	iir('s04885', 'secretary')
	iir('s04995', 'accountant')
'sno', 'deptNo'	iir('s04885', 'd01')
	iir('s04995', 'd03')

Table 2 shows some random variables and associated for the university database given in **Figure 6**. In a rela--tional database, an attribute, e.g., sno, can be seen as a random variable, and then each possible data value in this column is an event. That is, randomly picking up a tuple in this column, then its value could be any one of all those that are allowed. An attribute is therefore a variable. The variable holding a particular value is an event.

3.3 Deriving IIR from Business Rules

Business rules are domain dependent, established by an individual organisation and they are *ad hoc* logical limitations on data. Business rules may be embedded in an ontology and also could be in a database. In order to derive IIR from these business rules, we treat an Object-Property in an OWL ontology as if it were a constraint in a database. Both could be represented as additional relationships. For example, in a university, there might be a rule: "Any newly recruited lecturer must hold a PhD". Then we have an IIR(newly recruited lecturer, PhD), which means that if someone is a newly recruited lecturer, then she/he must hold a PhD corresponding to him/her.

4. Testing our Idea

In this section, we show a case study that verifies our idea. We created an ontology entitled “Academic” and a relational database “University”. The program runs in Oracle using PL/SQL. This case study elucidates the different results when reasoning is based on the database in question only, and on both the database and the ontology integrated through IIR.

There are 11 tables in the “University” database shown in **Figure 7**.

And as we mentioned in previous sections, in the OWL ontology, a “Class” is transformed into a “table”. “subClassOf” is treated as a “Class”. A “DatatypeProperty” is changed to an “attribute”. An “ObjectProperty” is a relationship between classes, which are transformed into constraints upon these tables. So, we arrive at 10 tables shown in **Figure 8** from the “Academic” ontology.

Thus the schema of the “University” is substantially extended as shown in **Figure 9**, from which more IIR are derived.

Figure 10 shows the 10 tables in SQL Plus of Oracle.

4.1 Original IIR and Derived IIR

A few business rules are defined for this case study. They specify correspondences between the “Academic” OWL ontology and the “University” relational database. For example, there is a table in the “University” database called “staff”. There is a class in the “Academic” OWL ontology named “Person”, and “staff” is a subclass of “Person”. Other 18 business rules are concerned with equivalent classes between ontology and the database at class level. There is one on the ObjectProperty.

The original IIR derived from the ontology and the ‘database and business rules are shown in **Table 3**.

Applying the IIR inference rules listed earlier to the IIR identified, more IIR are derived.

4.2 Implementation and Results

As we previously mentioned, firstly, we created tables for the relational database (named 0db.sql), the ontology (named 0onto.sql) and IIR (named 0IIR.sql), used for storing both original IIR from the ontology and the database and 0IIR_DB.sql is used for storing the original IIR from the relational database alone.

Secondly data values are inserted into the database tables and all the original IIR are entered into the IIR tables. Then all single attributes and class names and attribute names from the ontology and the database are obtained and inserted into the attributes table with duplicate components removed.

Thirdly, three intermediate tables are created. The tables named fo1 and fo2 store the former and the latter part of original IIR respectively, and the table t1 stores intermediate results.

Fourthly, a procedure is invoked. For instance, when a user asks a question, relevant IIR closures will be looked at. They embody relevant information for the query. There is a string match function in this procedure.

In order to find out the difference that the ontology makes, we compare the two results. One was obtained by using both the “Academic” OWL ontology and the “University” database, and the other obtained using the database alone. They are shown in **Table 4**.

As **Table 4** shows, the first column “Attributes” indicate all attributes that are extracted from “Academic” OWL ontology and the “University” database. The column “Closures from both ONTO and DB” shows the IIR closures that we derive by running our prototype when the “Academic” OWL ontology is involved, The column “Closures from DB only” consists of the IIR closures that we derive by running our prototype when only the “University” database is involved.

We use the same five questions in the testing. As **Table 4** shows, the attributes that are included in the results are ticked. When the database alone is used, a query for “sno” gives 12 results and a query for “matricNo” gives 7 results. When both the ontology and the database are used, the same query for “sno” gets 14 results and the same query for “matricNo’ gives 9 results. That is, two more attributes are found to be included in the respective IIR closures when the ontology is involved, which means that more information is made available due to the ontology.

5. Conclusions

We have described how an ontology may be linked with database in order to derive hidden information. A prototype in Oracle was developed to verify our ideas. We use the notion of IIR (Information content Inclusion relation) and inference rules for IIR.

We have found that if we do not invoke a relevant ontology, a query may be unanswerable. After invoking an ontology, more relationships between objects become available, and therefore more elements can connect to one another, and as a result, a query may become answerable, and as a result, more information can be derived from data in a database. To achieve this, a key is to beable to identify IIR from both a database and an ontology. We have presented a way of doing so.

More work need to be done in the future, for instance, to display correspondences between a query and the answers in a more accurate and specific way, *i.e.*, not just listing the answers. One issue that is not aesthetic is how to achieve semantic alignment between an ontology and a database, on which we are currently working.

sno	fname	lname	sex	Address	tel
s02923	John	Key	M	6 Lawrence St, Glasgow	2384
s02933	Julie	Lee	F	8 George St, Glasgow	2234
s04885	Ann	White	F	18 Taylor St, Glasgow	5112
s04995	Susan	Brand	F	28 High St, Paisley	3001
s06465	Mary	Tregear	F	7 George St, Paisley	7754
s06883	David	Ford	M	64 Well St, Paisley	8772

1. DB-staff

sno	title	school
s02923	lecturer	computing
s02933	professor	business

2. DB-faculty

sno	school	pno
s06465	business	p00203
s06883	engineering	p00334

3. DB-researcher

sno	position	deptNo
s04885	secretary	d01
s04995	accountant	d03

4. DB-administration staff

matricNo	pno
ts030283	p00334
tm051083	p00203

6. DB-postgraduates

matricNo	fname	lname	sex	Address
ts030283	Tony	Shaw	M	20 George St, Paisley
tm051083	Tina	Murphy	F	16 George St, Paisley
rn050385	Robert	Nielson	M	11 George St. Paisley
hf151186	Henry	Ford	M	7 Well St. Paisley
jw010483	John	White	M	5 Novar Dr, Glasgow
sb210682	Susan	Brand	F	2 Manor Rd, Glasgow
cp020381	Chris	Paul	M	6 Lawrence St, Glasgow

5. DB-student

matricNo	creditsSoFar	matricNo
rn050385	155	M050385

8. DB-projects

matricNo	creditsSoFar
rn050385	155
hf151186	65

7. DB-undergraduates

courseID	courseName	creditHour	lecturerNo	school
c0054	Oracle Development	24	s02923	computing
c0021	International Finance Planning	24	s06465	business
c0154	Advanced Oracle Development	24	s02923	computing
c0155	Networking Principles	16	s06883	computing
c0220	Software Development	24	s06883	computing

9. DB-courses

matricNo	courseID	results
rn050385	c0054	A
hf151186	c0021	C1
cp020381	c0154	B1
cp020381	c0155	C2
sb210682	c0220	B2

10. DB-achievements

deptID	departmentName
d01	administration
d03	finance

11. DB-departments

Figure 7. Tables in the "University" database

name	age	sex

1. onto-Person

specialty	educationDegree	back-ground

2. onto-Worker

school	title	background

3. onto-Faculty

department	Position	background

4. onto-Administration staff

school	background

5. onto-Assistants

supervisor

7. onto-Postgraduates

credits

8. onto-Undergraduates

studentNo (PK)	studentName	major	address	E-mail	sex

6. onto-Student

projectNo (PK)	projectName

9. onto-Projects

courseNo (PK)	courseName	creditHour

10. onto-Course

Figure 8. Table transformations from the "Academic" ontology

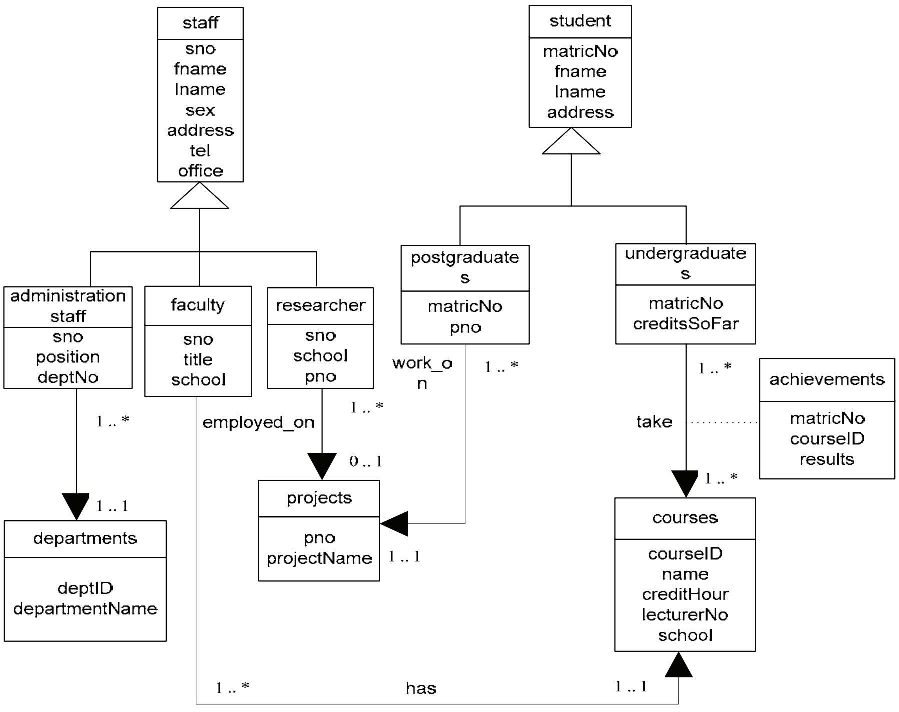
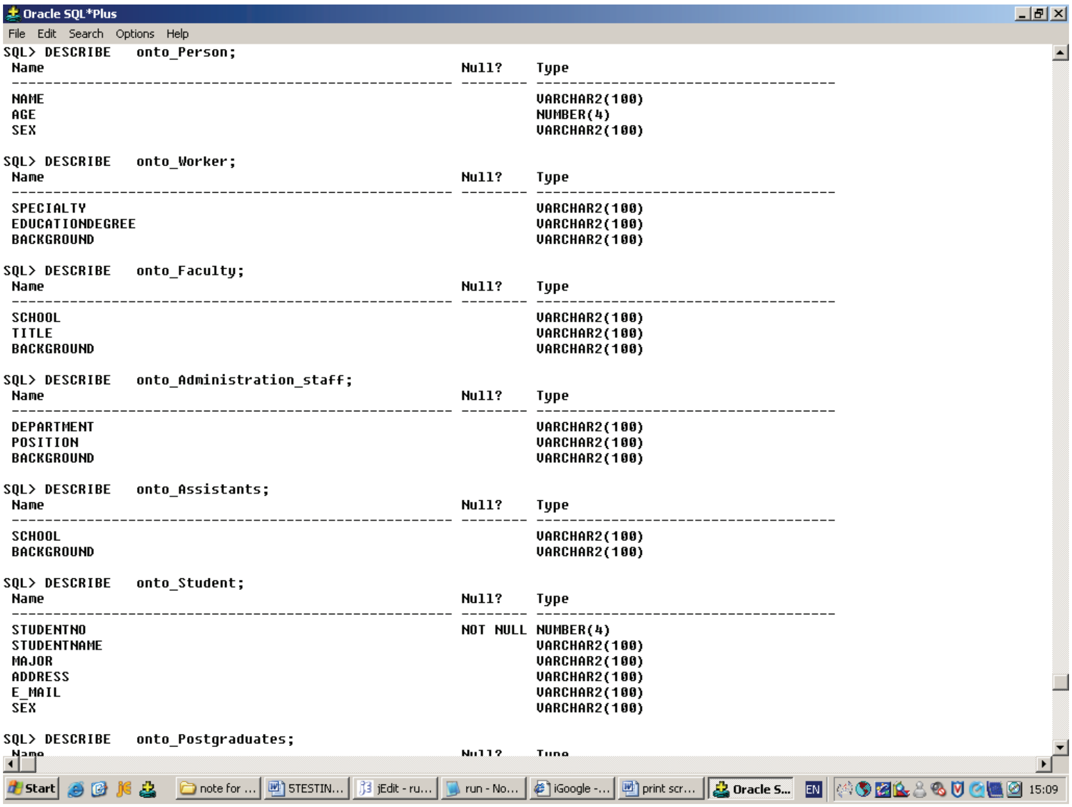
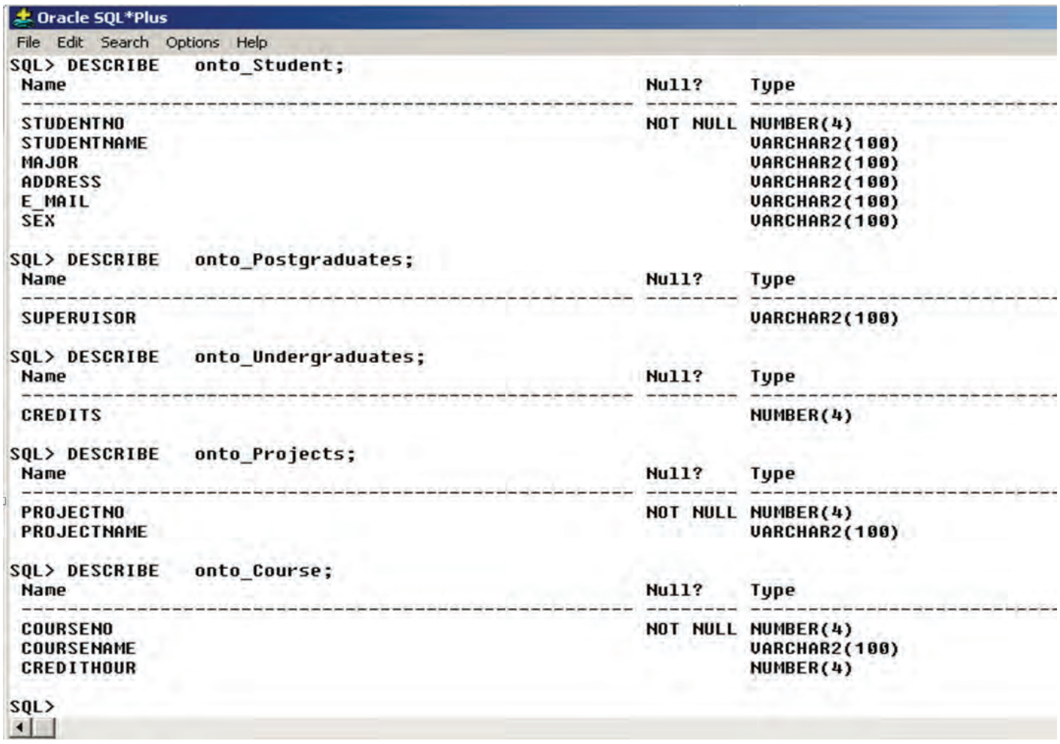


Figure 9. "University" database extended due to an ontology



```
SQL> DESCRIBE   onto_Person;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------------------------------
 NAME                                                           VARCHAR2(100)
 AGE                                                            NUMBER(4)
 SEX                                                            VARCHAR2(100)

SQL> DESCRIBE   onto_Worker;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------------------------------
 SPECIALTY                                                      VARCHAR2(100)
 EDUCATIONDEGREE                                                VARCHAR2(100)
 BACKGROUND                                                     VARCHAR2(100)

SQL> DESCRIBE   onto_Faculty;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------------------------------
 SCHOOL                                                         VARCHAR2(100)
 TITLE                                                          VARCHAR2(100)
 BACKGROUND                                                     VARCHAR2(100)

SQL> DESCRIBE   onto_Administration_staff;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------------------------------
 DEPARTMENT                                                     VARCHAR2(100)
 POSITION                                                       VARCHAR2(100)
 BACKGROUND                                                     VARCHAR2(100)

SQL> DESCRIBE   onto_Assistants;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------------------------------
 SCHOOL                                                         VARCHAR2(100)
 BACKGROUND                                                     VARCHAR2(100)

SQL> DESCRIBE   onto_Student;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------------------------------
 STUDENTNO                                             NOT NULL NUMBER(4)
 STUDENTNAME                                                    VARCHAR2(100)
 MAJOR                                                          VARCHAR2(100)
 ADDRESS                                                        VARCHAR2(100)
 E_MAIL                                                         VARCHAR2(100)
 SEX                                                            VARCHAR2(100)

SQL> DESCRIBE   onto_Postgraduates;
 Name                                                  Null?    Type
```



```
SQL> DESCRIBE   onto_Student;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ---------------------------
 STUDENTNO                                             NOT NULL NUMBER(4)
 STUDENTNAME                                                    VARCHAR2(100)
 MAJOR                                                          VARCHAR2(100)
 ADDRESS                                                        VARCHAR2(100)
 E_MAIL                                                         VARCHAR2(100)
 SEX                                                            VARCHAR2(100)

SQL> DESCRIBE   onto_Postgraduates;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ---------------------------
 SUPERVISOR                                                     VARCHAR2(100)

SQL> DESCRIBE   onto_Undergraduates;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ---------------------------
 CREDITS                                                        NUMBER(4)

SQL> DESCRIBE   onto_Projects;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ---------------------------
 PROJECTNO                                             NOT NULL NUMBER(4)
 PROJECTNAME                                                    VARCHAR2(100)

SQL> DESCRIBE   onto_Course;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ---------------------------
 COURSENO                                              NOT NULL NUMBER(4)
 COURSENAME                                                     VARCHAR2(100)
 CREDITHOUR                                                     NUMBER(4)

SQL>
```

Figure 10. The “Academic” ontology represented in SQL Plus

Table 3. IIR derived from the "Academic" OWL ontology and the "University" relational database

IIR derived from the 'Academic' OWL ontology	IIR derived from the 'University' Relational Database	IIR derived from Business Rules (corresponding relations)
***class, subclass* (17) *and equivalent class* (1)**	***class, subclass* (5) *and equivalent class* (0)**	***class, subclass* (1) *and equivalent class, attributes* (20)**
1.IIR(Worker, Person)	1.IIR(faculty, staff)	1.IIR(staff, Person)
2.IIR(Faculty, Worker)	2.IIR(administration_staff, staff)	2.IIR(Worker, staff)
3.IIR(Professors, Faculty)	3.IIR(researcher, staff)	3.IIR(Faculty, faculty)
4.IIR(Lecturer, Faculty)	4.IIR(postgraduates, student)	4.IIR(Administration_staff, administration_staff)
5.IIR(Postdoc, Faculty)	5.IIR(undergraduates, student)	5.IIR(Projects, projects)
6.IIR(Administration_staff, Worker)	According to the 'University' relational database EER diagram (**Figure 8**), these 5 IIR could be derived from it.	6.IIR(Student, students)
7.IIR(Dean, Administration_staff)		7.IIR(Postgraduates, postgraduates)
8.IIR(Chair, Administration_staff)		8.IIR(Undergraduates, undergraduates)
9.IIR(Clerical_staff, Administration_staff)		9.IIR(Course, courses)
10.IIR(System_staff, Administration_staff)		10.IIR(courseNo, courseID)
11.IIR(Director, Administration_staff)		11.IIR(projectNo, pno)
12.IIR(Assistants, Worker)		12.IIR(staff, Worker)
13.IIR(Reacher_assistants, Assistants)		13.IIR(faculty, Faculty)
14.IIR(Teaching_assistants, Assistants)		14.IIR(administration_staff, Administration_staff)
15.IIR(Student, Person)		15.IR(projects, Projects)
16.IIR(Postgraduates, Student)		16.IIR(students, Student)
17.IIR(Undergraduates, Student)		17.IIR(postgraduates, Postgraduates)
18.IIR(Teachers, Faculty)		18.IIR(undergraduates, Undergraduates)
		19.IIR(courses, Course)
		20.IIR(courseID, courseNo)
		21.IIR(pno, projectNo)
***ObjectProperty* (7) *and equivalent ObjectProperty* (2)**	***ObjectProperty* (5) *and equivalent ObjectProperty* (0)**	***ObjectProperty* (0) *and equivalent ObjectProperty* (3)**
1.---teache_of IIR(Faculty, Course)	1.---work_in IIR(administration_staff, departments	1.IIR(has, teache_of)
2.---attend_course IIR(Student, Course)	2.---has IIR(faculty, courses)	2.IIR(research_in, work_on)
3.---research_by IIR(Professors, Projects)	3.---employed_on IIR(researcher, projects)	3 IIR(study_in, work_on)
4.---instruct_by IIR(Course, Faculty)	4.---work_on IIR(postgraduates, projects)	
5.---research_in IIR(Postgraduates, Projects)	5.---take IIR(undergraduates, courses)	
6.---study_in IIR(Postgraduates, Projects)		
7.---join_course IIR(Student, Course)		
8. IIR(study_in, research_in)		
9.IIR(join_course, attend_course)		
***Constraints----NOT NULL* (6)**	***constraints----PK* (22)**	***constraints* (0)**
1.IIR(studentNo, 'studentNo, studentName,major,address,E-mail,sex')	1.IIR(sno, 'sno,fname,lname,sex,address,tel,office')	
2.IIR(courseNo, 'courseNo, courseName, creditHour')	2.IIR(sno, 'sno,title,school')	
3.IIR(projectNo, 'projectNo, projectName')	3.IIR(sno, 'sno,school,pno')	
4.IIR(Student, studentNo)	4.IIR(sno, 'sno,position,deptNo')	
5.IIR(Projects, projectNo)	5.IIR(matricNo, 'matricNo,fname,lname,sex,address')	
6.IIR(Course, courseNo)	6.IIR(matricNo, 'matricNo,pno')	
	7.IIR(matricNo, 'matricNo,creditsSoFar')	
	8.IIR(pno, 'pno,projectName')	
	9.IIR(courseID, 'courseID, courseName, creditHour, lecturerNo, school')	
	10.IIR('matricNo,courseID', 'matricNo,courseID,results')	
	11.IIR(deptID, 'deptID,departmentName')	
	12.IIR(staff, sno)	
	13.IIR(faculty, sno)	
	14.IIR(researcher, sno)	
	15.IIR(administration_staff, sno)	
	16.IIR(student, matricNo)	
	17.IIR(postgraduates, matricNo)	
	18.IIR(undergraduates, matricNo)	
	19.IIR(projects, pno)	
	20.IIR(courses, courseID)	
	21.IIR(achievements, 'matricNo,courseID')	
	22.IIR(departments, deptID)	

Table 4. IIR closures compared

Attributes	Closures from both ONTO and DB (the number of results)					Closures from DB only (the number of results)				
	Worker(3)	faculty(25)	student(22)	sno(14)	matricNo(9)	Worker(1)	faculty(16)	students(1)	sno(12)	matricNo(7)
Person	√	√	√							
Worker	√	√	√			√				
Student			√							
Faculty		√	√							
Course		√	√							
deptNo		√		√			√		√	
sex		√	√	√	√		√		√	√
school		√	√	√			√		√	
title		√		√			√		√	
position		√		√			√		√	
studentNo			√							
studentName			√							
major			√							
address		√	√	√	√		√		√	√
E-mail			√							
courseNo		√	√							
courseName		√	√							
creditHour		√	√							
projectNo				√	√					
projectName		√		√	√		√			
sno		√	√	√			√		√	
fname		√		√	√		√		√	√
lname		√		√	√		√		√	√
tel		√		√			√		√	
office		√		√			√		√	
pno		√		√	√		√		√	√
matricNo					√					√
creditSoFar					√					√
courseID		√	√							
lecturerNo		√	√							
staff	√	√	√				√			
faculty		√	√				√			
students			√					√		
courses		√	√				√			

6. Acknowledgements

This work is partly sponsored by the a grant for Distributed Information Systems Research from the Carnegie Trust for Universities of Scotland, 2007, a grant for research on Semantic Interoperability between Distributed Digital Museums from the Carnegie Trust for Universities of Scotland, 2009, and a PhD studentship of the University of the West of Scotland, UK.

REFERENCES

[1] T. R. Gruber, "A Translation Approach to Portable Ontologies," *Knowledge Acquisition*, Vol. 5, No. 2, 1993, pp. 199-220.

[2] M. West, "Database and Ontology [Online]," 2008. Wiki HomePage. http://ontolog.cim3.net/cgi-bin/wiki.pl?DatabaseAndOntology

[3] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web," *Scientific American*, Vol. 284, No. 5, 2001, pp. 34-43.

[4] Z. M. Xu, S. C. Zhang and Y. S. Dong, "Mapping between Relational Database Schema and OWL Ontology for Deep Annotation," *Proceedings of the* 2006 *IEEE/WIC/ACM International Conference on Web Intelligence*, IEEE Computer Society, 2006, pp. 548-552. http://portal.acm.org/citation.cfm?id=1248823.1249215&coll=AC

M&dl=ACM&CFID=16616566&CFTOKEN=44022427

[5] C. B. Necib and J. C. Freytag, "Query Processing Using Ontologies," *Proceedings of* 17*th International Conference on Advanced Information Systems Engineering*, Springer, Porto, Portugal, 13-17 June 2005.

[6] K. Munir, M. Odeh and R. McClatchey, "Ontology Assisted Query Reformulation Using the Semantic and Assertion Capabilities of OWL-DL Ontologies," *Proceedings of the* 2008 *International Symposium on Database Engineering & Applications*, ACM, Coimbra, Portugal, 2008, pp. 81-90.

[7] J. Feng, "The 'Information Content' Problem of a Conceptual Data Schema," *Systemist*, Vol. 20, No. 4, 1998, pp. 221-233.

[8] K. Xu, J. Feng and M. Crowe, "Defining the Notion of 'Information Content' and Reasoning about it in a Database," *Knowledge and Information Systems*, Vol. 18, No. 1, 1 January 2009, pp. 29-59

[9] F. I. Dretske, "Knowledge and the Flow of Information," MIT Press, Cambridge, 1981.

[10] K. Loney, "Oracle Database 10g: The Complete Reference," McGraw-Hill Companies, Inc., NY, 2004.

[11] B. C. Grau and B. Motik, "OWL 1.1 Web Ontology Language: Model-Theoretic Semantics. W3C Working Draft [Online]," 8 January 2008. W3C. http://www.w3.org/TR/owl11-semantics/

[12] Z. M. Xu and Y. J. Huang, "Conversion from OWL Ontology to Relational Database Schema," College of Computer and Information Engineering, Hohai University, Nanjing, 2006.

J. Software Engineering & Applications, **2010, 3, 644-652**
doi:10.4236/jsea.2010.37074 Published Online July 2010 (http://www.SciRP.org/journal/jsea)

The Need to Evaluate Strategy and Tactics before the Software Development Process Begins

Samir Kherraf[1], Laila Cheikhi[2], Alain Abran[1], Witold Suryn[1], Eric Lefebvre[1]

[1]École de Technologie Supérieure, Université du Québec, Montréal, Canada; [2]École Nationale Supérieure d'Informatique et d' Analyse des Systèmes, Université Mohammed V-Souissi, Rabat, Maroc.
Email: Samir.kherraf.1@ens.etsmtl.ca, Cheikhi@ensias.ma, {Alain.abran, Witold.suryn, Eric.lefebvre}@etsmtl.ca

Received May 26th, 2010; revised June 18th, 2010; accepted June 26th, 2010.

ABSTRACT

Experience has shown that poor strategy or bad tactics adopted when planning a software project influence the final quality of that product, even when the whole development process is undertaken with a quality approach. This paper addresses the quality attributes of the strategy and tactics of the software project plan that should be in place in order to deliver a good software product. It presents an initial work in which a set of required quality attributes is identified to evaluate the quality of the strategy and tactics of the software project plan, based on the Business Motivation Model (BMM) and the quality attributes available in the ISO 9126 *standard on software product quality.*

Keywords: *Business Motivation Model, Quality of Strategy, Quality of Tactic, ISO* 9126, *Software Product Quality*

1. Introduction

In any engineering field, the establishment of the project plan is an important step. The plan includes, among other things, the goal and the objectives, and the means for attaining them. In software engineering in particular, the establishment of the project plan and the realization of that plan are required activities, and they contribute to the production of a high-quality software product.

The goal and objectives represent the basis on which the development of software relies. It is also recognized that modifying software during or post-development in order to include new or changed requirements is easier than changing the mission for which the project was initiated, the latter requiring a quality approach.

The software engineering community has proposed and used several methods, techniques, and tools to support software engineers in producing quality products, such as Extreme Programming, RUP, and the Agile method. Whether software engineers are developing new software products or enhancing existing ones, they have to rely on resources, techniques, and methods to meet the required project deadlines and do so within budget. Sometimes, even for highly experienced software development teams with the best technology using a development method efficiently, the outcomes in terms of software quality can be disappointing: if both the project resources and the project process are under control, the problem may reside in earlier phases, prior to the development process itself, with the software project plan, for example, which may not have been under control.

Generally, the implementation of a particular software project is part of the global strategy of the organization, that is, the business plan that sets up the generic context of a specific project plan expressed in terms of strategy (goal) and tactics (objectives) [1]. Thus, a software project plan is a particular case of a business plan, which meets the requirements of a Business Motivation Model (BMM), as recommended in [1].

A set of quality attributes and measurements is proposed in ISO 9126 to evaluate the quality of the software product during its development phases. Can this set also be used to evaluate the quality of the strategy and tactics of the software project? Here, we propose a set of quality attributes to evaluate the quality of the strategy and tactics of the software project by adapting the quality attributes proposed in ISO 9126 [2] to the context of the strategy and tactics activities of the BMM [1] for software projects. It is important to note that these BMM activities are to take place before the development of the software product itself begins.

We stress that the main purpose of the work reported here is not to propose quality attributes for the BMM key concepts related to strategy and tactics in general, but to use these concepts in the context of software project plans, in particular in the initialization phase before beginning the development process.

This paper is organized as follows: Section 2 presents an overview of the BMM and its two key concepts: strategy and tactics. Section 3 presents an overview of ISO 9126 and related software product quality models and measurements. Section 4 presents some work related to the BMM and to ISO 9126. Section 5 illustrates the usefulness of the BMM in ISO 9126. Section 6 identifies quality attributes for the two key concepts of the BMM related to strategy and tactics, and Section 7 presents a discussion on the findings.

2. Business Motivation Model

One of the recent developments in the Object Management Group (OMG) standards for modeling business plans is the specification of the Business Motivation Model (BMM), which "*provides a scheme or structure for developing, communicating, and managing business plans in an organized manner*" [1]. The main activities of the BMM are aimed at "*identifying factors that motivate the establishing of business plans, defining the elements of business plans, and indicating how all these factors and elements inter-relate*" [1].

The BMM focuses on four key concepts: Ends, Means, Influencer, and Assessment, which constitute the two major areas of the BMM (**Figure 1**):

The first area is related to the Ends and Means of business plans: "*Ends...are things the enterprise wishes to achieve, for example, Goals and Objectives, and Means...are things the enterprise will employ to achieve those Ends, for example, Strategies, Tactics, Business Policies, and Business Rules.*"

The second area is related to the Influencers: "*Influencers...shape the elements of the business plans, and the Assessments made about the impacts of such Influencers on Ends and Means (i.e. Strengths, Weaknesses, Opportunities, and Threats).*"

Although the elements of the business plan are initially developed to address questions related to the business field from a business viewpoint, it is interesting to look at their applicability to the software project plan to address issues related to software quality, also from the business

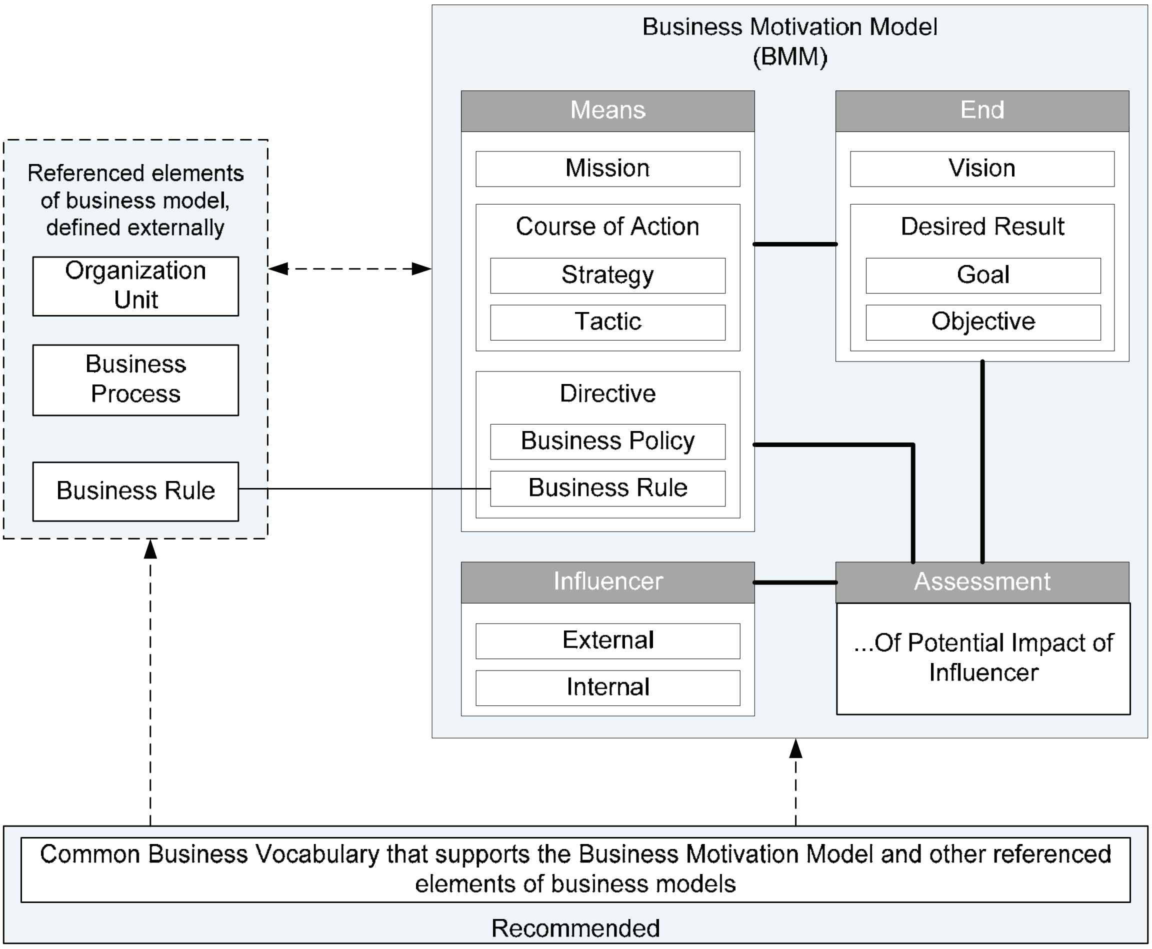

Business Motivation Model
(BMM)
Means
Mission
Course of Action
Strategy
Tactic
Directive
Business Policy
Business Rule
End
Vision
Desired Result
Goal
Objective
Influencer
External
Internal
Assessment
...Of Potential Impact of Influencer
Referenced elements of business model, defined externally
Organization Unit
Business Process
Business Rule
Common Business Vocabulary that supports the Business Motivation Model and other referenced elements of business models
Recommended

Figure 1. Business motivation model overview [1]

viewpoint. The basic idea of the BMM being "*to develop a business model for the elements of the business plan before system design or technical development is begun*" [1], these elements are also important to take into account in the context of the software project plan before the development process is begun. Such elements will be used to determine the likely ways of obtaining a quality software product, and to indicate where economies in terms of budget and resources can be realized.

The following sections focus on the elements related to strategy and tactics of the Means of the BMM and their usefulness in producing high-quality software.

2.1 Strategy and Tactics

According to the BMM, "a Strategy represents the essential Course of Action to achieve Ends—Goals in particular. A Strategy usually channels efforts towards those Goals. A Strategy is more than simply a resource, skill, or competency that the enterprise can call upon; rather, a Strategy is accepted by the enterprise as the right approach to achieve its Goals, given the environmental constraints and risks" [1]. In practice, developing a strategy consists in defining actions that must be coherent and executed correctly to achieve its goal. It is applicable to every type of action: political, economic, etc.

Examples would be Microsoft's strategy against Open Source software and its business strategy with companies producing computers (e.g. selling new computers with Windows pre-installed) to oblige customers to use and experiment with its new operating system.

Next, a tactic is defined in the BMM as, "a Course of Action that represents part of the detailing of Strategies. A Tactic implements Strategies. For example, the tactic 'Call first-time customers personally' implements the Strategy 'Increase repeat business'. Tactics generally channel efforts towards Objectives. For example, the Tactic 'Ship products for free' channels efforts towards the Objective 'Within six months, 10% increase in product sales'" [1]. A tactic is therefore the set of actions that must be realized in order to contribute to achieving the strategy goal. As for the strategy, the tactic is applicable to every type of action: economic, commercial, sport, diplomatic, etc.

In the example of Microsoft's strategy to dominate the operating system market with Windows, the set of tactics could be as follows:

1) Enter into agreements with computer producers to sell computers with Windows pre-installed.

2) Develop useful software applications for specialists and the general public that function only in the Windows environment.

3) Ensure that only Microsoft Corporation will be able to improve its applications and its Windows operating system, and do this internally.

4) Make improvements to Windows according to the automated feedback of its users—via the Internet.

As mentioned before, both strategy and tactics are applicable to any type of project and in any type of discipline. Therefore, in the context of the software project plan, the strategy and tactics established for a particular project should be accepted by the project manager as the relevant approach to achieving the project goal and objectives, depending on the context of the project. For example, the strategy and tactics adopted for a health sector software project and a defense software project are obviously different in some ways from those adopted for a game software project.

2.2 Difference between Strategy and Tactics

The terms "tactic" and "strategy" are sometimes used incorrectly and often interchangeably. In a military context, tactics are conceptual actions associated with troop engagement which are implemented to achieve an objective, while strategy is concerned with how those various actions are linked. A strategy is a plan of action designed to achieve a particular goal, whereas a tactic is a specific mission designed to achieve a specific objective. Michel de Certeau [3] defines a tactic as a calculated action determined by the absence of a proper "locus", and it is deployed and organized by the laws of a foreign power. Tactics are isolated actions or events that take advantage of opportunities offered by gaps in a particular strategic system. He defines a strategy as an entity that is recognized as an authority, and is relatively inflexible because it is embedded in its proper locus, either spatially or institutionally.

In discussing the difference between strategy and tactics, Hall in [4] refers to:

The teleological point of view: "Strategy supports the tactical objective, while the tactics supports the goals."

The pragmatic point of view: "A tactic is something you can change under your authority, but to change strategy you must ask your boss."

Some of the statements provided "as is" in the BMM document [1] showing the differences between the strategy (Goal) and tactics (Objectives) are presented in **Table 1**.

The most common way to explain the differences between strategy and tactics is a war analogy: a tactic is designed to win a battle and the strategy is designed to win the war. Another example would be a game (chess, for example): a tactic requires only the calculation of variants (I play it, he must play it, and then I play it, etc.), while the strategy relies on general heuristics and the intuition of the player.

From **Table 1**, the bottom line states that "objectives should always be measurable", which implies that objectives will have measurements. In the BMM, the informative Appendix titled "Metrics for the BMM" states that "*implicit in many areas of the Business Motivation*

Table 1. Differences between strategy and tactics

Differences between Strategy (Goal) and Tactics (Objectives)
1. Strategies usually ***channel efforts towards*** Goals. Tactics generally ***channel efforts towards*** Objectives.
2. Strategies tend to **be longer term and broader in scope.** Tactic tends to **be shorter term and narrower** in scope.
3. Strategies pair only with **Goals**, and Tactics only with **Objectives**.
4. There is a continuum from major Strategies that **impact the whole of the business to minor** Tactics **with limited, local effects**.
5. Strategies are put into place to support the **long-term Goals**—*i.e*. a planning horizon that is typically several years or more—while Tactics are the Courses of Action implemented to deal with the **shorter planning horizon** of a year or less (the current operational plans**).**
6. Goal tends to be **longer term**, **qualitative** (rather than quantitative), **general** (rather than specific), and **ongoing**. Objective tends to be **short term**, **quantitative** (rather than qualitative), **specific** (rather than general), and **not continuing** beyond its time frame (which may be cyclical).
7. Objectives should always be **time-targeted** and **measurable.** Goals, in contrast, **are not specific** in these ways.

Model is the subject of metrics. In almost all organizations, there are 'things of interest' that are heavily measured and tracked. These metrics govern, control, and influence a wide range of important aspects of the organization" [1].

However, the BMM does not provide a set of acceptable measures for measuring objectives, but only gives examples of some cases to show that it is possible to measure objectives, such as "quantify the Goal" and "be profitable". The enterprise might, for example, set one objective to have a monthly net revenue of at least $ 5 million (by a specified date) and another to have an annual net revenue of at least $ 100 million (by a specified date)" [1].

The non availability of a set of measurements accepted by the BMM organization is symptomatic of the reality that "*the enterprise will decide on many different things to be measured. Each of these measurements will have differing degrees of importance relative to the attainment of some Objective or set of Objectives*" [1].

3. ISO 9126

The ISO 9126 series was published between 2001 and 2004, under the general title: Information Technology—Software Product Quality. This set of ISO documents includes four parts [2,5-7]. The first part of ISO 9126 [2] is an international standard providing three views of quality:

- Internal view, which can be evaluated without the execution of the software during the design and coding phases;
- External view, which can be evaluated with the execution of the software during the testing and operation phases;
- In-use view, which can be evaluated in terms of using the software in a defined context and environment, and not in terms of its intrinsic properties.

ISO 9126 also proposes a two-part quality model for these three quality views: an internal and an external quality model (shared) and an in-use quality model (individual).

The other three parts of the ISO 9126 [5-7] are technical reports, each of which proposes a set of non exhaustive lists of measures for each quality model.

3.1 Quality Model

The first part of the ISO 9126 quality model is related to internal quality and external quality. Since internal quality and external quality share the same structure of two hierarchical levels, they are represented in one model—see **Figure 2**. The second part concerns the quality-in-use model with only one hierarchical level—see **Figure 3**.

According to ISO 9126, these parts of the quality model provide a set of characteristics and subcharacteristics that could be combined in order to specify the software quality requirements and to evaluate the software product quality throughout the whole software life cycle phases. Moreover, this model allows for the specification and evaluation of software product quality from different perspectives by different stakeholders of the software project, such as the user, the developer, the maintainer, the acquirer, the evaluator, and the quality manager.

The three technical reports of ISO 9126 provide a catalog of measures for each quality characteristic (subcharacteristic) as a tool for evaluating software product quality.

3.2 Quality Measures

Technical reports ISO TR 9126-2 and -3 [5,6] provide a list of measures for each subcharacteristic of the internal and the external quality model (**Figure 2**). ISO TR 9126-4 [7] provides a set of measures for each characteristic of the in-use quality model (**Figure 3**).

The internal measures are applied to the intermediate product and deal with the static aspect of the software product, while the external measures are applied to the final product and deal with the dynamic aspect of the software. The quality-in-use measures reflect quality from the user's point of view of the system containing the software: they aim to measure to what extent the user's objectives are achieved.

4. Related Work

Software product quality is widely discussed in standards,

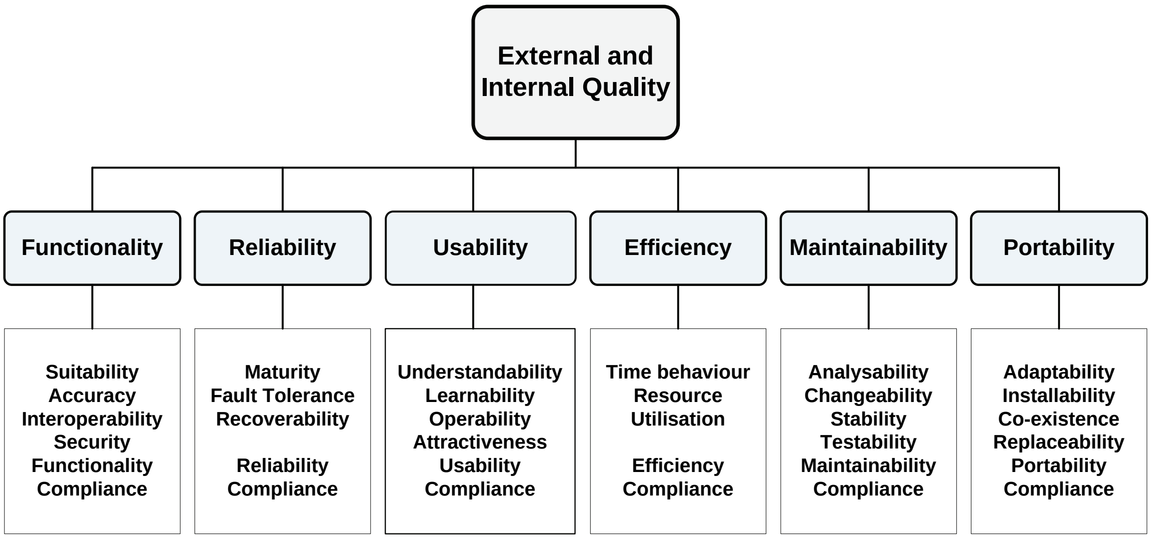


Figure 2. Internal & external quality model of ISO 9126 [2]

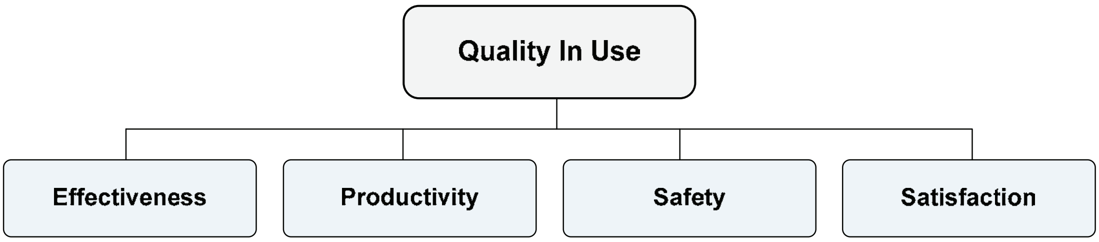


Figure 3. Quality-In-Use model of ISO 9126 [2]

but the quality of the strategy and tactics of the software project plan that lead to this software quality are not addressed. While the researcher and practitioner communities propose quality attributes and measurements to evaluate the quality of the software product during its development (specification to delivery) [2,8-10], less work has been carried out on quality in the earlier phases of the software, like the initialization phase. This phase places the whole process in a business-level environment in which the mission should be realized in terms of required “strategies for approaching goals, and tactics for achieving objectives” [1].

Various works discuss strategy, but in different contexts. Ronan Fitzpatrick [11] introduces a new paradigm for software quality, which he calls strategic quality drivers for acquirers and suppliers of the software product. He presents six strategic quality drivers that impact the acquirer (procurer) of software, which are: Technical excellence, User acceptance, Corporate alignment, Statutory conformance, Investment efficiency, and Competitive support. He also proposes five other strategic quality drivers that impact the supplier (producer), which are: Quality management, Development excellence, Domain specialty, Corporate accreditation, and Competitive excellence. This approach, the Software Quality—Strategic Driver Model, is an excellent foundation for the academic syllabus for the study of software quality, and represents an opportunity for quality thinking to be applied at a strategic level.

Alex Wright [12] reviews the relationship between quality and strategy. In his view, quality “*has failed to influence organizations’ strategy and strategic processes due to its continued operational bias; and the traditional calls for quality to be part of an organization’s strategy are misguided and originate from our own limited perception of quality as essentially operational*.” He recommends that quality be integrated by academic researchers and practitioners into the strategic process of the organization and be part of an organization’s strategy, but not a contributor to it.

The Malcolm Baldrige National Quality Award (MBNQA) [13] is widely recognized in the USA. It consists of seven interrelated categories that comprise the organizational system for performance and excellence. This model is used to assess the quality status of organizations with a focus on the relationships between leadership, information and analysis, human resource planning, process quality, and the customer. It does not recommend any method on how an organization should develop and deploy an excellent strategy, but does encourage organizations and their quality practitioners to consider the relationship between strategy and quality.

The European Foundation for Quality Management (EFQM) has proposed the Organizational Excellence Model [14], a non prescriptive framework based on nine criteria. Five of these criteria are related to "Enablers" and four to "Results". The purpose of this framework is to explain the connections between what an organization does (Enablers) and the outputs it is able to achieve (Results). The model is also used to define what resources and capabilities are necessary in order to deliver on the organization's strategic objectives. This model is seen as endorsing a cyclical approach to improvement.

Moreover, quality from the business perspective is discussed in [15] in the context of the TL9000 standards, and not the BMM standard. The authors suggest an integrated life cycle quality model, referred to as the "complement model for software product quality". The approach proposed in [15] combines the high-level quality view of the TL9000 Handbook and the detailed view from ISO 9126.

5. BMM and ISO 9126 in the Software Life Cycle

The three ISO's quality views concern the "Product and the User views" of the software product during its development and when it is delivered to the final user. What is also needed is a view of the quality of the software from a business perspective, which should be defined very early on in the software project plan, before the project approval stage. This business view of quality should be part of the BMM strategy and related tactics, and be supported by it.

Therefore, the business view of quality should be integrated into the BMM plans, that is, during the "initialization" of each new software project plan (**Figure 4**).

As mentioned before, it is claimed that ISO 9126 is applicable to all types of software projects and makes it possible to evaluate the quality of the software product during all the phases of the software life cycle.

The qualities of the "Product development view" and "Final product view" are discussed in the ISO 9126 standard, and a set of quality characteristics is proposed. The purpose there is to address the first view, "Product initialization", which is very important for any project in any engineering field. The goal and objectives have to be identified in the project plan thoroughly and without ambiguity, and the means of achieving them should also be identified. These means are strategy and the related tactics.

Moreover, according to ISO 9126, the quality characteristics of a software product are described as "external" and "internal" quality characteristics that the software product must satisfy to obtain a final product. This quality is expressed in terms of "in-use" quality characteristics. What is needed is the set of quality characteristics of strategy and tactics that contribute to the quality of the software product. In the next section, we present the set of quality characteristics of strategy and tactics of the software project plan that should be considered before starting to develop the software product.

6. Quality Attributes for Strategy and Tactics

The ISO 9126 documents are used to identify the quality attributes of the strategy and the tactics of the software project plan. Those attributes are presented in the following sections.

6.1 Quality Attributes for Strategy

As far as the BMM business plan is concerned, a strategy in the context of the software project plan also represents

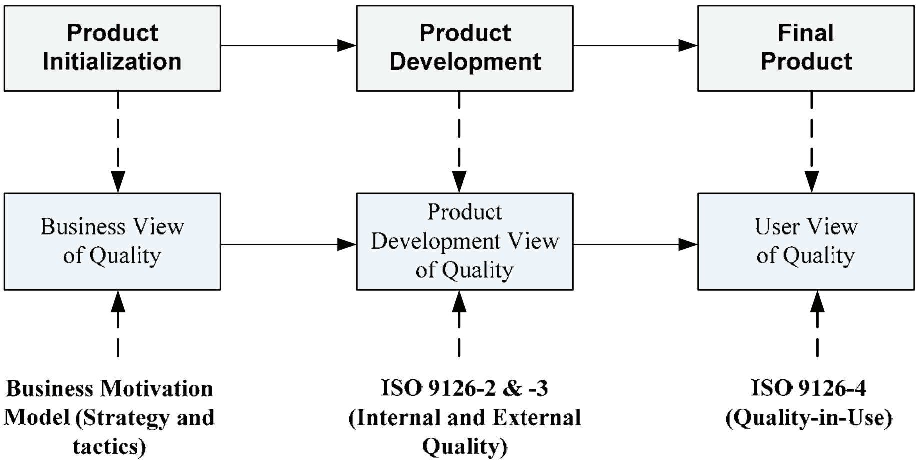

Product Initialization
Product Development
Final Product
Business View of Quality
Product Development View of Quality
User View of Quality
Business Motivation Model (Strategy and tactics)
ISO 9126-2 & -3 (Internal and External Quality)
ISO 9126-4 (Quality-in-Use)

Figure 4. Views of quality in the software life cycle

a proper approach that should be adopted in the long term to achieve its goal, taking into account the constraints and risks posed by the environment. Referring to ISO 9126 [2], the results of the strategy—in terms of quality—appear not only in the good approach, but also in the quality approach used to produce the final software product in a defined context and environment—e.g. ISO 9126 Quality-in-Use.

Therefore, the quality of the strategy should be achieved by the four quality characteristics of Quality-in-Use —see **Figure 3**. The definition of these quality attributes is based on, and adapted from, those of the ISO Quality-in-Use quality model to the context of the software project plan strategy established for developing the software product:

- **Effectiveness:** This characteristic should make it possible to measure the accuracy and completeness of the realized strategy goal established for developing the software product. Effectiveness doesn't concern the ways in which the goal is to be realized.
- **Productivity**: This characteristic should make it possible to evaluate the use of various resources in order to achieve the goal of the strategy adopted for developing the software product. Productivity enables measurement of the success of the strategy goal, and therefore the proposal of enhancements.
- **Safety**: This characteristic makes it possible to evaluate the strategy risk levels (negative impacts) that could affect the users, the software, the business, and the environment; for example, identification of emerging incidents capable of causing economic damage, such as competition, etc.
- **Satisfaction**: This characteristic should make it possible to satisfy the goal of the strategy and the actions taken to achieve it. Satisfaction concerns the degree of achievement of the established needs of the strategy.

In the context of the software project plan, the quality of a strategy is defined here as the quality approach, characterized by effectiveness, productivity, security, and satisfaction, that an enterprise should adopt in the long term to achieve its goals in a defined context and environment.

6.2 Quality Attributes for Tactic

As far as the business plan of the BMM is concerned, the tactics in the context of the software project plan also represent the set of guidelines to follow in order to achieve the goal and support the strategy. Therefore, the tactics—on the quality level—appear in the set of quality guidelines to follow in order to achieve the strategy goal.

Moreover, according to the BMM, the strategy contributes to the identification of the tactics [1] and, according to the ISO 9126 [2] standard, the quality-in-use requirements contribute to the identification of the external quality requirements of the software. Therefore, the quality of the tactic emerges on the set of external quality characteristics (**Figure 2**) that the tactic should satisfy in order to meet the strategy goal.

From the set of ISO 9126 external quality characteristics, a set of quality attributes suitable for a tactic are identified and redefined for the context of the software project plan, as follows:

- **Functionality**: This characteristic focuses on what to do in order to satisfy the tactic's objective. A tactic must not only be functional and useful, but it must also function in a defined context and meet established objectives (Suitability). Moreover, quality goes beyond whether a tactic functions or not, to how well its application produces good performance or good indicators to be followed (Accuracy).
- **Reliability**: This characteristic concerns the degree of confidence of the tactic. A tactic must maintain a performance level when faced with faulty operation (Fault tolerance). It should not lead to the failure of its purpose, but to the achievement of its objective (Maturity).
- **Usability**: This characteristic is related to the level of use of the tactic by the users. A tactic must be suitable and easily comprehensible (Understandability), accompanied by a well-documented process in order to facilitate its application (Learnability). A tactic must also be operational (Operability) in order that it can be used in an adequate way for specific tasks.
- **Efficiency**: This characteristic focuses on the suitable performances that the tactic should provide according to the resources used. An effective tactic allows the use of the essential resources: material, personal, budget, and planning required for its achievement (Resource utilization).
- **Maintainability:** This characteristic concerns the capacity of a tactic to be modified, enhanced, and adapted to the changes in the application domain. Thus, a tactic should be diagnosed: 1) to identify the causes of its failure (Analyzability), 2) to identify the solution and be able to implement the required modifications (Changeability), and 3) to test these modifications (Testability) in order to resolve problems arising after the modifications and to ensure a stable tactic (Stability).
- **Portability:** This characteristic represents the capability of the tactic to be used in different environments. A portable tactic is one that is adaptable in different strategies without the use of resources or actions other than those already prescribed (Adaptability). The use of two or more tactics together (Co-existence) is a very important factor in the realization of a strategy and includes several disciplines. Moreover, the capacity of the tactic to be used instead of another in the strategy for a different objective, but under the same conditions, is also important (Replaceability).
- **Compliance:** The tactic should conform to the

guidelines and standards related to the various quality characteristics identified. It should be realized according to the rules of the application domain: the software.

In the work reported here, the quality of a tactic is defined in the context of the software project plan as the set of quality guidelines related to the Functionality, Reliability, Usability, Effectiveness, Maintainability, and Portability that the tactic should satisfy in order to meet the strategy goal.

7. Summary

This paper has addressed the issue of identifying and including quality attributes early in the software project plan, *i.e.* before starting to develop the software product. This issue is related to the key concepts of the strategy (goal) and tactics (objectives) of the BMM plan. In a software engineering project, these two concepts constitute a basic issue to tackle, whether for developing new software or for enhancing or redeveloping existing software.

A software quality product is often considered in terms of the contractual needs to be achieved between the client and the developer. The focus is generally directed toward the software life cycle process, as described in ISO 9126. So, it is appropriate to address quality before the development process starts, that is, during the initialization phase when the goal (strategy) and the objectives (tactics) are established, to contribute effectively to the quality of the final software product.

Good strategy and tactics are at the heart of successful software project plans in any organization. However, while there are a few different approaches to integrating quality into strategic and tactical thinking/planning, none have looked at including or evaluating the quality of the strategy and the quality of the tactics of the software project plan.

The motivation of the work reported here is to improve the quality of the software by improving the quality of the strategy and the quality of the tactics. Therefore, in this paper, a set of quality characteristics for these two key concepts of the BMM was identified in the context of a software project plan. It was identified based on those provided in ISO 9126 and were redefined by adapting the ISO 9126 definitions to strategy and tactics.

Table 2 groups together the various quality characteristics identified for the strategy and tactics.

Therefore, in the context of the software engineering project plan:

- The quality of strategy is defined here as the quality approach, characterized by effectiveness, productivity, security, and satisfaction, that an enterprise should adopt in the long term to achieve its goals in a defined context and environment.
- The quality of a tactic is defined here as the set of quality guidelines, related to Functionality, Reliability, Usability, Effectiveness, Maintainability, and Portability, that the tactic should satisfy in order to meet the strategy goal.

Table 2. Quality characteristics for strategy and tactics

Strategy Quality characteristics
• Effectiveness • Productivity • Safety • Satisfaction
Tactics Quality characteristics
• Functionality: suitability, accuracy • Reliability: maturity, fault tolerance • Usability: understandability, learnability, operability. • Efficiency: resource utilization • Maintainability: analyzability, changeability, testability, stability • Portability: adaptability, co-existence, replaceability • Compliance

Another issue addressed in this paper is the usefulness of including a new perspective in a software project plan: the business view, to set targets prior to beginning the development of the software project. Once the project has been completed, we can evaluate whether or not these established targets have been met, and, if so, to what extent. Moreover, such a view is important, since it places the whole process in a business-level environment in which the mission should be realized in terms of the required strategy (goal) and tactics (objectives).

On the one hand, when developing new software in a non mature market or for an emerging market (the first time that type of software has been developed), there are no best practices, guidelines, or techniques that have already been used and tested which can be adapted to the needs of the project. Therefore, the non consideration of the business view of quality will affect the quality of the software product during the development phase, not only from the business perspective, but also from that of the developers and end users. On the other hand, for a repeatable type of project, even if best practices are available, the business view of quality should also be emphasized when choosing suitable practices, combining them, or improving them to meet the needs of the new project; that is, determining "what" the business wants to accomplish and "how" it intends to accomplish it [1].

In contrast, evaluation of the quality of the strategy and tactics, that is, the goal and objectives respectively, requires the availability of measurements. While the BMM recognizes the usefulness of measures for evaluating the objectives, it does not provide a set of accepted measures to use. This can be justified from three points of view. The first point is that the notion of metrics, although recognized as an important discipline by the BMM, is presented only in an informative appendix, which needs to be reworked in order to incorporate it as a part of the BMM elements.

The second point is the absence of quality attributes to enable evaluation of the usefulness or otherwise of the stated goal and objectives. In fact, a set of quality attributes related to strategy and tactics has been identified in this paper. The third point is related to the difficulty of designing good measurements: "*The metrics for an Objective are established by the measures of performance of the Goal that the Objective quantifies. To be able to do this, an appropriate unit of measure for the metric must be determined for each Objective. The Objective then expresses the target value that the metric should attain in the timeframe specified. In that way, while a Goal sets the direction, its corresponding Objectives set the milestones to be attained in pursuing the Goal*" [1].

As future work to validate the mapping of the ISO 9126 quality model and quality of strategy and tactics, research is being pursued to propose measurements, indicators, and criteria to measure, in practical cases, the quality attributes of the strategy and the tactics of a software project plan.

REFERENCES

[1] Object Management Group, "Business Motivation Model Specification," September 2007.

[2] ISO/IEC, "IS 9126-1, Software Engineering—Product Quality—Part 1: Quality Model," International Organization for Standardization, Geneva, 2001.

[3] M. de Certeau, "The Practice of Everyday Life,'' University of California Press, 2002.

[4] J. Hall, "Tactics, Strategies, and Quality Words," *Business Rule Journal*, 2005.

[5] ISO/IEC, "TR 9126-2, Software Engineering—Product Quality—Part 2: External Metrics," International Organization for Standardization, Geneva, 2003.

[6] ISO/IEC, "TR 9126-3, Software Engineering—Product Quality—Part 3: Internal Metrics," International Organization for Standardization, Geneva, 2003.

[7] ISO/IEC, "TR 9126-4, Software Engineering—Product Quality—Part 4: Quality in Use Metrics," International Organization for Standardization, Geneva, 2004.

[8] B. W. Boehm, J. R. Brown and M. Lipow, "Quantitative Evaluation of Software Quality," *Proceedings of the* 2*nd International Conference on Software Engineering*, IEEE Computer Society, Los Alamitos (CA), 1976, pp. 592-605.

[9] R. G. Dromey, "Concerning the Chimera [Software Quality]," *IEEE Software*, Vol. 13, No. 1, 1996, pp. 33-43.

[10] J. A. McCall, P. K. Richards and G. F. Walter, "Factors in Software Quality," US Rome Air Development Center Reports, US Department of Commerce, Vol. 1-3, 1977.

[11] R. Fitzpatrick, "Strategic Drivers of Software Quality: Beyond External and Internal Software Quality," 2*nd Asia-Pacific Conference on Quality Software*, IEEE Computer Society, 2001.

[12] A. Wright, "Quality's Strategic Failure: A Review of the Key Literature," Occasional Paper Series 2003, University of Wolverhampton, 2003.

[13] Baldrige National Quality Program, Criteria for Performance Excellence, B.N.Q. Program, Editor, 2009-2010.

[14] European Foundation for Quality Management, The EFQM Excellence Model 2010, EFQM, 2010.

[15] W. Suryn, A. Abran and C. Laporte, "An Integrated Life Cycle Quality Model for General Public Market Software Products," *Software Quality Management XII Conference*, British Computer Society, 2004.

J. Software Engineering & Applications, **2010, 3, 653-660**
doi:10.4236/jsea.2010.37075 Published Online July 2010 (http://www.SciRP.org/journal/jsea)

Model Transformation Using a Simplified Metamodel

Hongming Liu, Xiaoping Jia

College of Computing and Digital Media, DePaul University, Chicago, USA.
Email: {jordan, xjia}@cdm.depaul.edu

Received March 2nd, 2010; revised April 1st, 2010; accepted April 3rd, 2010.

ABSTRACT

Model Driven Engineering (*MDE*) *is a model-centric software development approach aims at improving the quality and productivity of software development processes. While some progresses in MDE have been made, there are still many challenges in realizing the full benefits of model driven engineering. These challenges include incompleteness in existing modeling notations, inadequate in tools support, and the lack of effective model transformation mechanism. This paper provides a solution to build a template-based model transformation framework using a simplified metamode called Hierarchical Relational Metamodel* (*HRM*). *This framework supports MDE while providing the benefits of readability and rigorousness of meta-model definitions and transformation definitions.*

Keywords: *Model Driven Engineering, Modeling, Metamodeling, Model Transformation*

1. Introduction

Model-Driven Engineering (MDE) tackles the problem of system development by promoting the use of models as the primary artifact to be constructed and maintained [1,2]. MDE shifts software development from a code-centric activity to a model-centric activity. Accomplishing this shift entails developing support for modeling concepts at different levels of abstraction and then transforming abstract models to more concrete descriptions of software. In other words, MDE reduces complexity in software development through modularing and abstraction [3].

Because of MDE's potential to dramatically change the way we develop applications, companies are already working to deliver supporting technologies. Although some variants of MDE, especially Model-Driven Architecture (MDA), are already quite advanced and serve as the conceptual foundation for commercial software products, there are many challenges to achieving true Model-Driven Engineering.

The major challenges that researchers face when attempting to realize the MDE vision were discussed in [4]. According to many research projects that point out the inadequacy in MDE development [5], there are two main challenges that MDE infrastructure faces:

- Providing precise, analyzable, transformable and executable models [4].
- Providing well-defined transformations that support rigorous model evolution, refinement, and code generation [5].

The second challenge, model transformation that supports rigorous model evolution, refinement, and code generation is also an active research area [6]. There are many research projects that provide fundamentals for model transformation. Model transformation is the process of converting one model into another model. Performing a model transformation requires a clear understanding of the syntax and semantics of both the source and target models. To take modeling to a higher level of abstraction, it is necessary to have a standard mechanism to define metamodels of modeling languages. OMG addresses this issue in its MDE initiative Model Driven Architecture (MDA) by creating Model Object Facility (MOF). In response to the need for a standard approach to defining the functions that map between metamodels, the OMG issued the MOF 2.0 Query/View/Transformation (QVT) Request for Proposals. Several replies were given by a number of companies and research institutions that evolved over three years to produce a common proposal that was submitted and approved [7].

This paper is organized as follows: Section 2 introduces the motivation and overview of our transformation approach. It also covers the characteristics of this model transformation approach. Section 3 presents a case study that demonstrates our approach. Section 4 discusses related work and Section 5 evaluates the transformation tools. Finally Section 6 concludes the paper.

2. Our Model Transformation Approach: HRMT

Both of the challenges mentioned in Introduction section point to a mutual research topic: metamodeling. Metamodeling is the key technology that ensures precise, analyzable models, and it is the very element of metamodeling that is the basis for transformation definition. Considering these challenges and their connection to metamodeling, we provide a solution that uses a simplified metamodel as the foundation for building a template-based model transformation framework. This simplified metamodel is called the Hierarchical Relational Metamodel (HRM). The Hierarchical Relational Metamodel is built upon Z-based Object-Oriented Modeling notation (ZOOM). HRM maintains both a tree structure and the relationships among model elements. The model elements and the tree structure are constructs of the ZOOM modeling language comparable to constructs of a programming language. To capture more complicated modeling language constructs like association, we adopt a mathematical collection to depict the relationships among different constructs.

Figure 1 shows the basic structure of our Hierarchical Relational Metamodel Transformation (HRMT) framework, A transformation engine takes the HRM defined source model as input, and use a template comprise of a set of transformation rules to produce output model in a format specified by the templates. In other words, the output from the transformation engine is a transformation of the input model. We regard a model as a set of model elements that are in correspondence with a metamodel element via the instantiation relationship. Meta-model based transformations use only the elements of the metamodels, thus the transformation description is expressed in terms of the two metamodels.

2.1 Source Model Representation

We use ZOOM notation to represent Platform Independent Model (PIM). ZOOM notation has a textual syntax defined by BNF, which gives us a simplified way to define and use ZOOM meta-model. Listing 1 shows an example of a model in ZOOM notation. Since ZOOM provides the textual syntax in a form that most programming languages have, we are able to build an internal representation of ZOOM models in a structure similar to Abstract Syntax Tree (AST), only the node in the tree will be constructs of the modeling language instead of constructs of programming language. However, to capture more complicated modeling language constructs like association. We adopt mathematical collection to depict the relationships of different constructs. Considering it's tree structure and such relationships, we name our metamodel Hierarchical Relational Metamodel (HRM).

The use of HRM provides a way for transformation to understand and make use of the abstract syntax and semantics of both the source and target models. Base on HRM, we design our template based model transformation to get the information necessary to generate target model or code from HRM-compliant models inside a model repository. A set of interchangeable templates can be provided for model transformation between different target technical platforms.

2.2 A Metamodeling Language

Metamodeling is a critical part of our transformation approach. It provides a mechanism to unambiguously define modeling languages ZOOM in our case. It is the prerequisite for a model transformation tool to access and make use of the models. We will now look into the design of our Hierarchical Relational Meta-model (HRM).

2.2.1 Hierachical Relational Metamodel

The fact that ZOOM notation has a textual syntax defined by BNF gives us a simplified way to define and use ZOOM model's metamodel. From implementation point of view, metamodel defines the internal representation of models. In programming language, this internal representation often takes the form of Abstract Syntax Tree (AST) that can be processed by interpreter or compiler. Since ZOOM provides the textual syntax in a form that most programming languages have, we are able to build an internal representation of ZOOM models in a structure

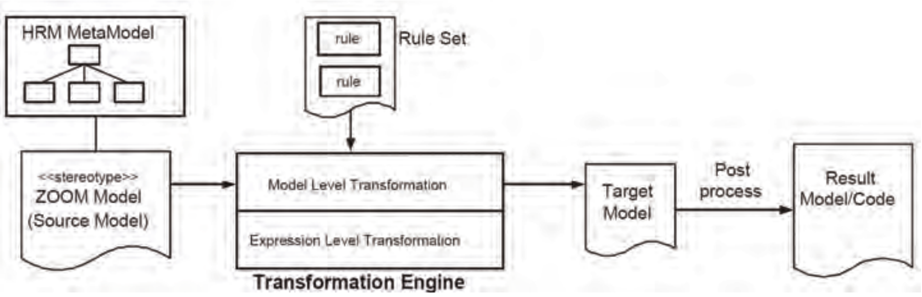


Figure 1. HRMT model transformation process overview

similar to Abstract Syntax Tree. The only difference is the nodes in the tree are constructs of the modeling language instead of constructs of programming language. To capture more complicated modeling language con--structs like association, we also adapt mathematics collection to depict the relationships of different constructs. It is considering its tree structure and such relationships that we name this metamodel Hierarchical Relational Metamodel (HRM).

2.2.2 HRM Definition

We provide the following definition of HRM:

Definition 1. Hierarchical Relational Meta-model is a 3-tuple: HRM = (N, C, R), where

N is a set of nodes: N = $\{n_1, n_2 \ldots n_j\}$

C is a relation on, which forms a tree structure that has one root and no unconnected nodes. Each node may have zero or more children. In other words, a node is either a *leaf* (*i.e*. with no children) or can be decomposed as one or more children and each child forms a subtree.

R= $\{r_1, r_2 \ldots 4_j\}$ is a set of relations between nodes, where r*i* is a relation on N × N.

Figure 2 shows a simple class diagram that has four classes: Student, Graduate, Undergraduate and Course. The corresponding HRM diagram is also shown in **Figure 2** in the middle. This metamodel can be represented as (N, C, R) according to Definition 1. More specifically, we can elaborate the contents of its three components as in **Table 1**.

The components r_1, r_2, r_3 and r_4 are relations between classes n_1, n_2, n_3, n_4 and relationship enroll, x, y.

2.3 Transformation Template

The rule set shown in **Figure 1** is a collection of transformation rules. Here we provide the definition of transformation rule as followings:

Definition 2. A transformation rule r = P -> (T_{pre}, T_{post}) where

P defines the pattern to select the element of source model and the template pair (T_{pre}, T_{post}) defines the mapping to target model.

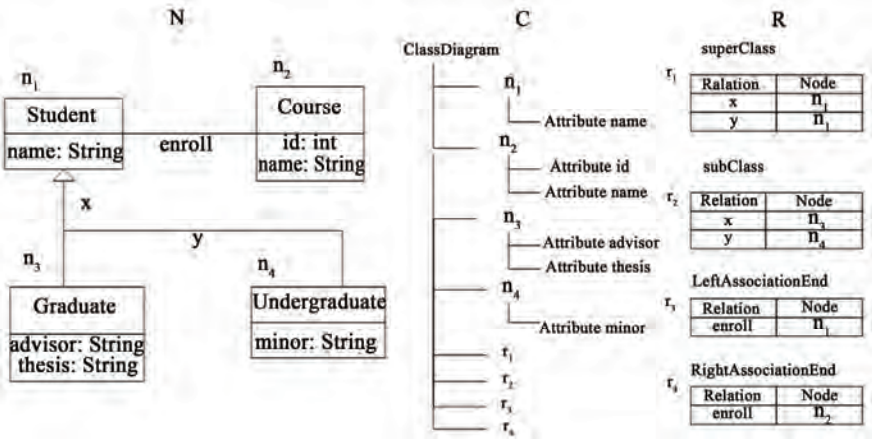
T_{pre} defines the mapping to target model before traversing children of selected element

T_{post} defines the mapping to target model

after traversing children of selected element.

The rationale of this design is closely related to the transformation algorithm that we will talk about in the next subsection.

In our framework, the development of transformation is in a large part the process of constructing transformation rules. The rule set in the **Figure 1** is an extensible component. Different set of templates can be used in different transformation tasks for various target platforms. That's why we also call the template "cartridge" to re-



(a) (b) (c)

Figure 2. HRM example of a class diagram

Table 1. HRM metamodel components

HRM Node	Content
N	{ ClassDiagrm, n_1, n_3, enroll, x, y, n_1.name, n_3.advisor,…}
C	{ (n_1, n_1.name), (n_3, n_3.advisor), (n_3, n_3.thesis),…}
R	{ r_1, r_2, r_3, r_4}
r_1	{(x, n_1), (y, n_1)}
r_2	{(x, n_3),(y, n_4)}
r_3	{(enroll, n_1)}
r_4	{(enroll, n_2)}

flect the exchangeability of templates. Template is the core component of the transformation framework.

2.4 Transformation Algorithm

Metamodel based transformation uses the elements of metamodel. Our adopting of Hierarchical Relational Metamodel (HRM) allows us to build an internal representation of ZOOM models in a structure similar to Abstract Syntax Tree (AST). Once metamodel is generated as an AST like structure, it is accessible by the transformation process through traversing the tree.

We use an algorithm of "pre-order" to traverse of the tree which means each node is visited before its children are visited and the root is visited first.

As we can see in Definition 2, a transformation rule has two mapping part, T_{pre} and T_{post}. They are represented as *rule.pre* and *rule.post*. *rule.pre* is the mapping before traversing children of selected element, while *rule.post* is the mapping after traversing children node will get visited.

The use of template and HRM-based transformation algorithm help produce the specification of target model or code. However, the order of the specification is not necessary in a desirable order. This is the reason why we introduce a post process that is responsible to reorganize the specification.

The post process will rearrange the specification in a desirable style that fits to the target technical platform. This approach has a similar style as proposed in Knuth's Literate Programming [8]. Literate programming is a methodology that combines a programming language with a documentation language, thereby making programs more robust, more portable, more easily maintained, and arguably more fun to write than programs that are written only in a high-level language. The main idea is to treat a program as a piece of literature, addressed to human beings rather than to a computer. The program is also viewed as a hypertext document, rather like the World Wide Web. Here we treated the generated model specification or code as pieces of segment that can be flexible rearranged so that it confirms to the requirements of target technical platform.

3. Case Study: A Hospital Information Management System

A case study that demonstrates our transformation framework has been done. It showcases the ability of transforming a ZOOM model specification to applications running in multi-platform. In order to show the power of our approach, the system described is not trivial. It is not a toy system, but it is a real-life example. This case study demonstrates how a fairly simple PIM is transformed automatically into rather complex PSMs and code, and fulfils real-life needs. The complexity of the complete example is considerable. However, the example is not completely detailed out in all parts of the system in order to limit the size of this paper.

The Hospital Information Management System (HIMS) is a web application designed to improve access to patient information through a central electronic information system, an Electronic Healthcare Record (EHR) [9]. A HIMS's goal is to streamline patient information flow and its accessibility for doctors and other health care providers. The implementation of HIMS will improve patient care quality and patient safety over time.

Using MDE to develop a Healthcare System is an active research topic. Raistrick in [10] outlines how MDA and UML were used in the context of an extension of the processing of clinical data to provide a patient-based electronic record. In [11], a method was tested on a patient record of a hospital which provided rules for generating SGML/XML DTD element and parameter entity declarations from object-oriented UML class diagrams.

The first task of developing HIMS using HRMT is defining the system independently from any specific technology. In another word, it is the creating of PIM. But hospitals do not want a model; they want a running system. Therefore, we need to transform the PIM into a PSM that is compatible with the hospital's technology infrastructure. In our case study, we choose Microsoft .NET and J2EE as the target web application platforms, considering the popularity of both platforms. We also choose Microsoft Access and SQL Server as target database platforms.

Using our HRMT framework, we are able to transform the PIM into full-fledged web applications in both .NET and J2EE platforms. We provide much more details about the case study in our research web site [12]. The web site also provides download of HRMT tool and ZOOM Software suite. Documentation of how to use the HRMT tool is included there as well.

4. Related Work

Many contributions related to model transformation have been discussed in literature [13]. A number of solutions to describe and implement model transformation are currently available. Different top-level taxonomies can be found in [14]. In order to compare our tool with other transformation tools more specifically, we choose four transformation tools for the following evaluations. Each of these tools represents a different transformation approach.

Direct-manipulation approach consists in providing some visitor mechanism to traverse the internal representation of a model and write code to a text stream. An example of this approach is Jamda [15], which is an object-oriented framework providing a set of classes to represent UML models, an API for manipulating models, and a visitor mechanism (so called CodeWriters) to generate code. Jamda does not support the MOF standard to define new meta-models; however, new model element types can be introduced by subclassing the existing Java classes that represent the predefined model element types.

Extensible Stylesheet Language Transformations (XSLT) is an XML-based language used for the transformation of XML documents into other XML document. XSLT may be used effectively for some class of transformations of MOF models, as they may be represented as XML documents via the XMI specification.

AndroMDA is a code generation tool that takes a UML model as input and generates source code as output. It adopts a template-based transformation methodology similar to ours in a degree but differs significantly in handling of metamodel. Compared to direct-manipulation transformation, the structure of a template resembles more closely the code to be generated. Templates lend themselves to iterative development as they can be easily derived from examples. Since the template approaches discussed in this section operate on text, the patterns they contain are untyped and can represent syntactically or semantically incorrect code fragments. On the other hand, textual templates are independent of the target language and simplify the generation of any textual artifacts, including documentation.

ATL is a model transformation language (MTL) developed by OBEO and INRIA to answer the QVT Request For Proposal. It can be used to do syntactic or semantic translation. ATL is built on top of a model transformation Virtual Machine. A model-transformation-oriented virtual machine has been defined and implemented to provide execution support for ATL while maintaining a certain level of flexibility. As a matter of fact, ATL becomes executable simply because a specific transformation from its metamodel to the virtual machine byte code exists. Extending ATL is therefore mainly a matter of specifying the new language features execution semantics in terms of simple instructions: basic actions on models (elements creations and properties assignments).

5. Evaluation

5.1 Evaluation Metrics of Transformation Tools

The purpose of this section is to compare our model transformation approach with other tools to evaluate its strength and weakness. As readability of metamodel and transformation definition is one of the advantages of our approach, we need to look deeper into the metrics that measure this quality. A large number of software product metrics have been proposed for the quality of software such as maintainability. Many of these metrics have not been properly validated due to poor methods of validation and non acceptance of metrics on scientific grounds [16]. In the literature, two types of validations, namely internal (theoretical) and external (empirical) are recommended [17]. Internal validation is a theoretical exercise that ensures that the metric is a proper numerical characterization of the property it claims to measure. Demonstrating that a metric measures what it purports to measure is a form of theoretical validation. External validation involves empirically demonstrating that a metric can be an important component or predictor of some software attributes of interest.

Kumar and Soni [18] have proposed a hierarchical model to evaluate qualities of object-oriented software. This proposed model has been used for evaluation of maintainability assessment of object-oriented design quality, especially in design phase. In this model, quality factors such as maintainability are measured by a set of metrics such as Number of Classes (NOC), Number of Ancestors (NOA) and Number of Methods (NOM). In [19], they present empirical experiments to validate this hierarchical model of object-oriented design quality metrics. We will introduce a set of metrics that we identified for readability, shown in **Table 2**. We will explain what each metric means and the rationale of choosing it. Although we do not conduct individual validation of each metric, our choices of metrics are following the same practice demonstrated in Kumar and Soni's study [18], and can be validated using a similar framework. Although we are not using this exact metric, we are following the approach of identifying factors that contribute to the educational grade level or readability. In the Flesch-Kincaid metric, two factors are identified: Avg-Number-WordsPerSentence and AvgNumberSyllables-PerWord. Considering the characteristics of our text format, we identified a more comprehensive set of factors. **Table 2** shows the factors that we identified.

5.2 Evaluation Result

We conducted an experimental trial on each of them. In the trial case, we used the example mentioned in Section 2. Using PIM shown in Listing 1, we generated Java code with each of the tools and evaluated the transformation using the metrics mentioned above.

Since all four tools use XMI as the format of source model, the metrics evaluation is between the ZOOM input format and XMI format. We will show in **Table 3** the result. Additionally, because the choice of input model reflects essentially the choice of metamodel, it indeed reflects the complicity of HRM and MOF comparison.

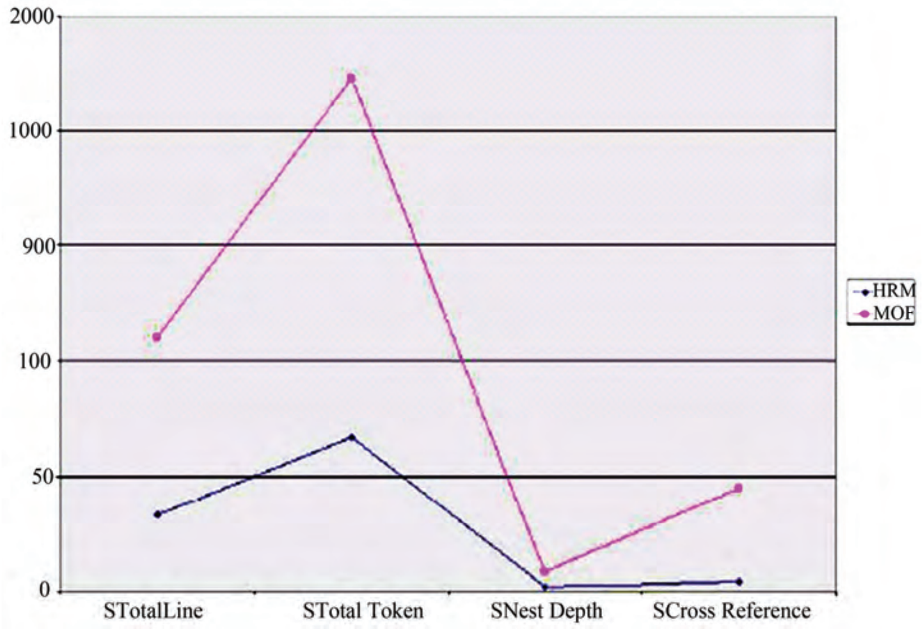
The result in **Table 3** shows that in all 4 metrics, ZOOM has a significantly lower number comparing to XMI. The TotalLine and Total-Token show that XMI model is much longer and verbose. The deeper nesting and significant amount of cross-references also made the XMI model harder to read. The result proves that using HRM can significantly simplify the metamodel.

Table 2. Explanation of metric factor

Metric Factor	Explanation
Source Model Total Lines (STotalLine)	Lines in the text of source model
Source Model Total Tokens (STotalToken)	tokens in the text of source model
Template Total Lines (TTotalLine)	lines in template
Template Model Total Tokens (TTotalToken)	Tokens in template
Source Model Nesting Depth (SNestDepth)	Deepest nesting level of source model
Template Model Nesting Depth (TNestDepth)	Deepest nesting level of template
Cross Reference in Source Model (SCrossReference)	Cross reference in source model
Cross Reference in Template (TCrossReference)	Cross reference in template
Reference to Metamodel in Template (MetaReference)	Reference to metamodel in template

Table 3. Evaluation result of source model

	HRMT	Jamda/Stylus/AndroMDA/ATL
Test Case		Generate Java code for Roster
Metamodel	HRM	MOF
Input Format	ZOOM	XMI
STotalLine	33	266
STotalToken	67	1612
SNestDepth	2	8
SCrossReference	4	45

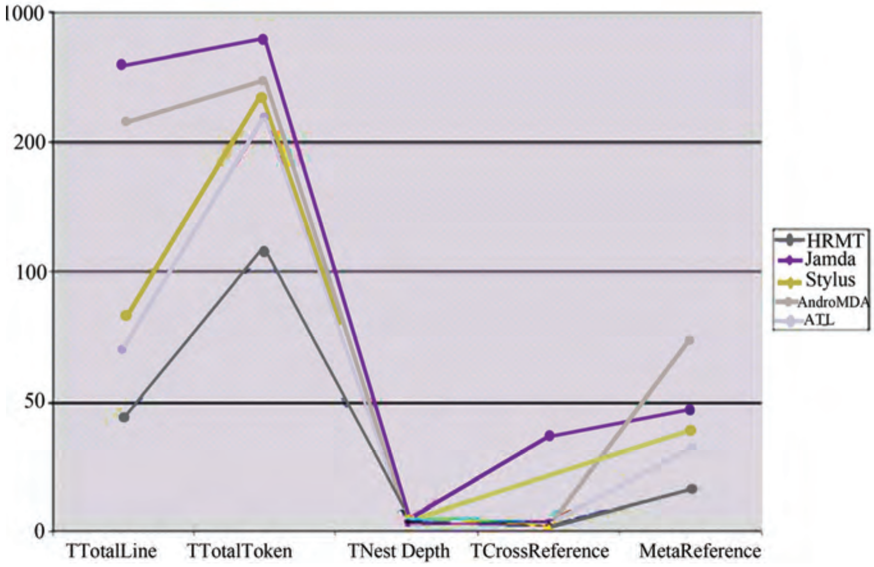


The rest of the metrics are about the template. Since the template is the main document that needs to be developed in the transformation process, these metrics reflect the transformation complicity. They are different from each other depending on the tools we are measuring. **Table 4** shows the result of comparing all the tools.

Overall in the case of TotalLine and TotalToken, HRMT uses the shortest template comparing to others. It's about half of ATL and only a fraction of Jamda, Stylus, and AndroMDA. Except for Jamda, there is no sig-

Table 4. Evaluation result of transformation template

	HRMT	Jamda	Stylus	AndroMDA	ATL
Test Case			Generate Java code for Roster		
TTotalLine	44	652	82	207	70
TTotalToken	108	1927	327	394	223
TNestDepth	4	5	6	5	3
TCrossReference	2	37	2	4	4
MetaReference	17	47	39	73	32



nificant difference between nesting depth and cross-reference amount of all the approaches. This implies that all the tools except Jamda are used in a similar way to organize the template. The high number of CrossReference in Jamda is because it is using Java API to perform the transformation directly, and Java API organized their functions in different methods, files, and even in different packages. MetaReference is the most critical metrics, because accessing metamodel information is the crucial step in model transformation. The more times that transformation has to access the metamodel, the more complicated the transformation process is. From the evaluation result, we can see that HRMT comes out using the least number of references to metamodel in both MetaReference and UniqueMetaReference. This is direct proof of having a simplified meta-model.

6. Contribution and Future Work

In this paper we present a framework that provide a simple, effective, and practical way to accomplish model transformations. This framework uses a simplified metamodel as the foundation for building a template-based model transformation framework. This simplified metamodel is called Hierarchical Relational Metamodel (HRM). The Hierarchical Relational Metamodel is built upon Z-based Object-Oriented Modeling notation (ZOOM). A template-based model transformation framework using Hierarchical Relational Meta-model (HRM) is introduced.

The current development of this project has made substantial progress and further research effort will be mainly focusing on two things 1) Fine-tuning and optimizing the tool and 2) Integration with other tools.

REFERENCES

[1] S. Kent, “Model Driven Engineering,” *Proceedings of the 3rd International Conference on Integrated Formal Methods*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 2335, 2002.

[2] K. Balasubramanian, A. Gokhale, G. Karsai, J. Sztipanovits and S. Neema, “Developing Applications Using Model-Driven Design Environments,” *Computer*, Vol. 39, No. 2, 2006, pp. 33-40.

[3] J. Gray, Y. H. Lin and J. Zhang, “Automating Change Evolution in Model-Driven Engineering,” *Computer*, Vol. 39, No. 2, 2006, pp. 51-58.

[4] R. France and B. Rumpe, “Model-Driven Development of Complex Software: A Research Roadmap,” *Future of*

Software Engineering, IEEE Computer Society, Washington, D.C., 2007, pp. 37-54.

[5] A. Uhl, "Model-Driven Development in the Enterprise," *IEEE Software*, Vol. 25, No. 1, 2008, pp. 46-49.

[6] S. Sendall and W. Kozaczynski, "Model Transformation: The Heart and Soul of Model-Driven Software Development," *IEEE Software*, Vol. 20, No. 5, 2003, pp. 42-45.

[7] Model Object Facility Query/View/Transformation final Adopted Specification, Object Management Group Document ad/05-11-01.

[8] D. E. Knuth, "Literate Programming," *CSLI Lecture Notes*, No. 27, 2003.

[9] "Electronic Healthcare Record Definition, Attributes and Essential Requirements," *Healthcare Information and Management Systems Society*, 2003.

[10] C. Raistrick, "Applying MDA and UML in the Development of a Healthcare System," *UML Satellite Activities*, 2004, pp. 203-218.

[11] E. Kuikka and A. Eerola, "A Correspondence between UML Ddiagrams and SGML/XML dtds," *Digital Documents and Electronic Publishing/Principles of Digital Document Processing*, 2000, pp. 161-175.

[12] Z-Based Object-Oriented Modeling Project. http://se.cs.depaul.edu/ise/zoom/

[13] D. Varró and Z. Balogh, "Automating Model Transformation by Example Using Inductive Logic Programming," *Proceedings of the* 2007 *ACM Symposium on Applied Computing*, New York, 2007, pp. 978-984.

[14] S. Helsen and K. Czarnecki, "Classification of Model Transformation Approaches," *Object-Oriented Programming, Systems, Languages & Applications* 03, *Workshop on Generative Techniques in the Context of Model-Driven Architecture*, 2003.

[15] Jamda: The Java Model Driven Architecture. http://sourceforge.net/projects/jamda/

[16] C. Kaner and W. P. Bond, "Software Engineering Metrics: What do They Measure and How do We Know?" *International Software Metrics Symposium* 2004, IEEE Computer Society Press, 2004.

[17] E. Fenton, "Software Metrics: Theory, Tools and Validation," *Software Engineering Journal*, Vol. 5, No. 1, 1990, pp. 65-78.

[18] M. Kumar and D. Soni, "Observations on Object-Oriented Design Assessment and Evolving New Model," *Proceedings of the National Conference on Software Engineering*, 2007, pp. 161-164.

[19] D. Soni, R. Shrivastava and M. Kumar, "A Framework for Validation of Object Oriented Design Metrics," IJC-SIS, 2009.

J. Software Engineering & Applications, **2010, 3, 661-673**
doi:10.4236/jsea.2010.37076 Published Online July 2010 (http://www.SciRP.org/journal/jsea)

Development of a Simulation-Based Intelligent Decision Support System for the Adaptive Real-Time Control of Flexible Manufacturing Systems

Babak Shirazi[1], Iraj Mahdavi[1], Maghsud Solimanpur[2]

[1]Mazandaran University of Science and Technology, Babol, Iran; [2]Urmia University, Urmia, Iran.
Email: irajarash@rediffmail.com

Received April 25th, 2010; revised May 19th, 2010; accepted May 27th, 2010.

ABSTRACT

This paper describes a simulation-based intelligent decision support system (IDSS) for real time control of a flexible manufacturing system (FMS) with machine and tool flexibility. The manufacturing processes involved in FMS are complicated since each operation may be done by several machining centers. The system design approach is built around the theory of dynamic supervisory control based on a rule-based expert system. The paper considers flexibility in operation assignment and scheduling of multi-purpose machining centers which have different tools with their own efficiency. The architecture of the proposed controller consists of a simulator module coordinated with an IDSS via a real time event handler for implementing inter-process synchronization. The controller's performance is validated by benchmark test problem.

Keywords: *Intelligent Decision Support System, Real Time Control, Flexible Manufacturing System, Multi-Purpose Machining Centers*

1. Introduction

A flexible manufacturing system (FMS) architecture can be characterized as a set of multi-purpose machine tools connected by automatic material handling and tool transportation devices. The material handling system has a mechanism to transport parts between machining centers. Automatic tool transportation devices can also transfer tools among tool magazines and the central tool storage area [1,2]. Any material handling system has a mechanism to transport parts and tools automatically. These systems can transfer tools among tool magazines and the central tool storage area while the system is in operation [3,4]. FMSs are essentially more flexible than the conventional manufacturing systems, mainly because of utilizing versatile manufacturing lines, redundant and reconfigurable machines, alternate routings, and flexibility in operation sequencing [5,6].

Due to different operations on a product and machine requirements to process each step of production, it is so hard to control different events that might happened at different cells to achieve best practice of performance criteria [7]. Regarding these considerations, control of these environments plays an essential role at manufacturing systems. Control framework has been studied on FMSs in the literature and there are different methods for selecting the most appropriate control policies at each decision point [8-15]. These strategies deal with the allocation of jobs to multi-purpose machining centers which have to be made in a flexible way. Most of these studies focus on reactive strategies that enable the FMS to better deal with randomness and variability. It means that most of these FMS controllers usually use fixed and offline policies to operate the system. However, these methods do not consider many realistic constraints and dynamic changes such as tool magazine capacity, operative efficiency changes and availability of tools in the part selection and operation assignment problems. These offline methods are mainly categorized into two forms: priori reactive control and the posteriori reactive control methods. The control is planned according to the structural information, forecasts, orders, management rules and objectives [16]. The online posteriori control adapted directly to the system for preventive deviations by controlling occurrence of events.

Improving the performance of an FMS supervised by an effective controller is still a complex task that not only is time consuming but also needs much human expertise in decision making [17]. In order to implement an adaptive controller, DSS have become an effective method for their adaptability in controlling complex and dynamic operations [18]. There have been limited investigations on IDSS for controlling such systems as a unified approach. There is a need to construct a framework in which a knowledge-based decision analysis will assist the decision process to improve the FMS control parameters.

An effective approach for reinforcement of IDSS performance is to develop an embedded simulation model that meets the desired objectives of the system [19-22]. Discrete-event simulation is a very powerful tool that can be used to evaluate alternative control policies in the manufacturing system [23-26]. Although the procedure of analyzing simulation results could rely on various guidelines and rules, decision-making still requires significant human expertise and computer resources. To efficiently use simulation in the decision process, integration of IDSS with simulation has been emphasized [27-30]. However, there have been limited investigations on integrating IDSS with the modular simulation languages as a unified approach for controlling manufacturing systems. So FMS control appears to be an excellent area for applying adaptive IDSS simulation-based controller.

This research focuses on developing a simulation-based intelligent expert system with dynamic rules contemplating tool and machine flexibility control. For implementing inter-process synchronization in real-time control of FMS, the proposed IDSS receives online results from simulation module and different scenarios of control parameters with simulation replication action. The outline of the paper is as follows. Section 2 describes adaptive flexibility control on FMS shop floor. Section 3 deals with FMS adaptive controller architecture to build IDSS. Sections 4 present experimental study to validate the effectiveness of the proposed system. Finally, conclusions are made in Section 5.

2. Adaptive Flexibility Control on FMS Shop Floor

2.1 Adaptive Control Mechanism

Adaptive supervisory implies selection of an appropriate control policy based on the current state of the workcell. Regarding dynamic control of manufacturing systems, jobs are dispatched to machining centers using dispatching rules at the specific moment based on the available information. Afterwards, appropriate tool is mounted in machining center according to the tooling strategy [31]. Because of the flexible characteristics of FMSs, control decisions should be applied as soon as possible based on the real time state of the system. An FMS adaptive controller has to deal with the dynamic environment in which the system operates to seize online machines and tools redundancy capabilities, alternative routing and hazard control remedy.

2.2 FMS Shop Floor Flexibility Control Functions

The most commonly accepted definition of flexibility is the ability to take up different positions or alternatively the ability to adopt a range of states [32]. Many different authors have defined many different types of flexibilities (machine, process, product, operation, routing, volume, production and expansion flexibility) in the literature [33-37]. Here we consider the flexibility control function as machine flexibility and tool flexibility. Browne *et al.* [38] defined machine flexibility as the ease of change to process a given set of part types. Buzacott [39] clarifies machine flexibility as the ability of the system to cope with changes. There are three technical constraints related to a machining center: number and capacity of machine-tools, local input/output buffer (LIB/LOB) size and operative efficiency. Das and Nagendra [40] define machine flexibility of a machining center as the ability of performing more than one type of processing operation efficiently. Therefore, machine flexibility is measured by the number of operations that a workstation processes and the time needed to switch from one operation to another. The more operations a workstation processes and the less time switching takes, the higher the machine flexibility becomes [37]. **Figure 1** shows the proposed adaptive flexibility control functions of the FMS shop floor.

As illustrated in **Figure 1,** tool flexibility can be defined as getting the right tool, to the right place at the right time [34,41]. The need for tooling strategies originates from the high variety and number of cutting tools that are typically found in automated manufacturing systems. The adoption of appropriate tool management policies that consider alternative tools allows the desired part mix and quantities to be manufactured efficiently while achieving improved performance [42]. At machine tool level, there are two technical constraints related to tool allocation: tool magazine capacity and tool life. Due to tool magazine capacity, there is a restriction on the number of operations that can be processed in a single tool setup. On the other hand, if tools can be loaded and unloaded while the machine is running, the capacity of the tool magazine can be assumed to be unlimited [32,43].

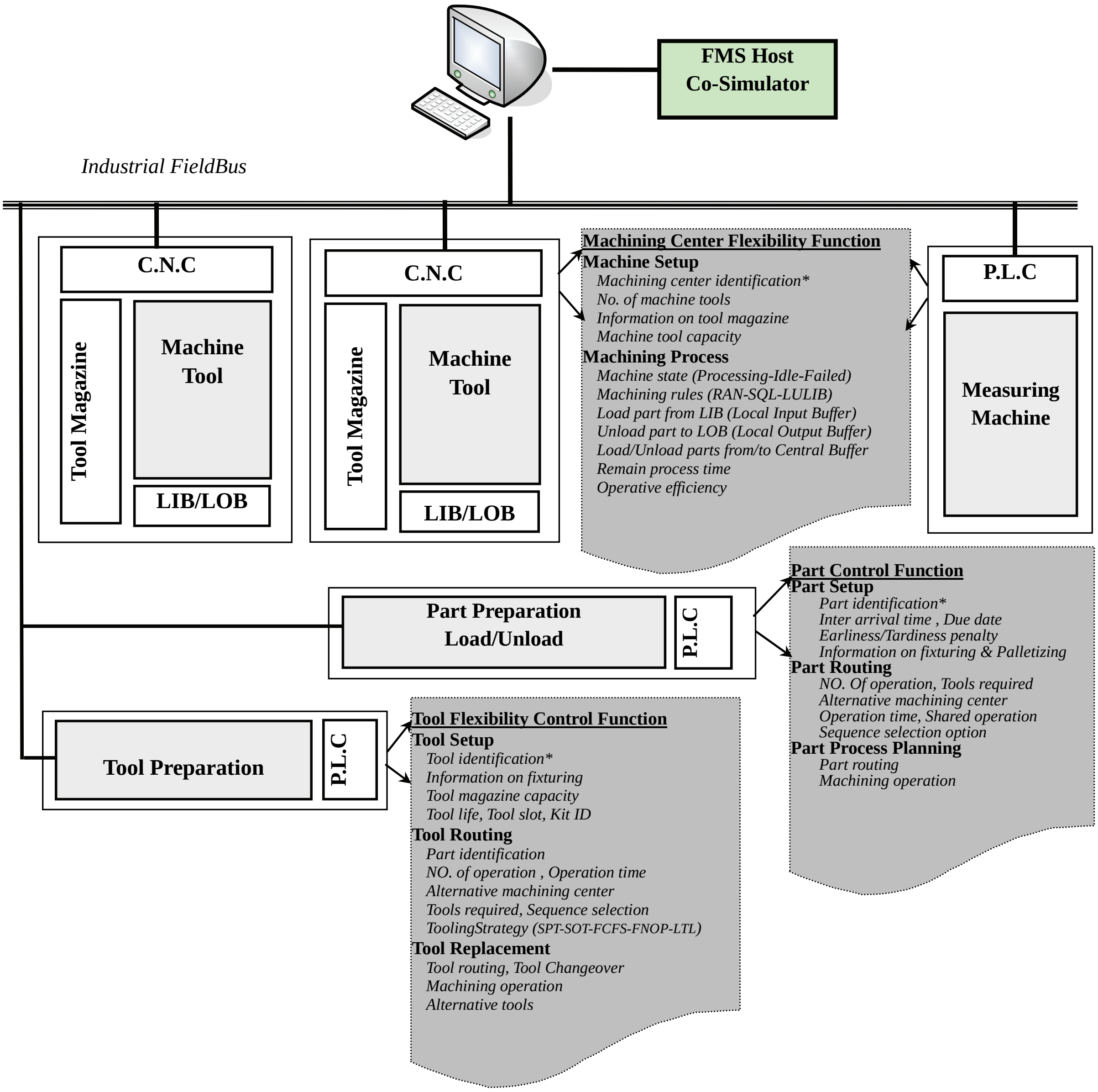

FMS Host
Co-Simulator
Industrial FieldBus
C.N.C
Tool Magazine
Machine Tool
LIB/LOB
C.N.C
Tool Magazine
Machine Tool
LIB/LOB
Machining Center Flexibility Function
Machine Setup
Machining center identification*
No. of machine tools
Information on tool magazine
Machine tool capacity
Machining Process
Machine state (Processing-Idle-Failed)
Machining rules (RAN-SQL-LULIB)
Load part from LIB (Local Input Buffer)
Unload part to LOB (Local Output Buffer)
Load/Unload parts from/to Central Buffer
Remain process time
Operative efficiency
P.L.C
Measuring Machine
Part Preparation
Load/Unload
P.L.C
Part Control Function
Part Setup
Part identification*
Inter arrival time , Due date
Earliness/Tardiness penalty
Information on fixturing & Palletizing
Part Routing
NO. Of operation, Tools required
Alternative machining center
Operation time, Shared operation
Sequence selection option
Part Process Planning
Part routing
Machining operation
Tool Preparation
P.L.C
Tool Flexibility Control Function
Tool Setup
Tool identification*
Information on fixturing
Tool magazine capacity
Tool life, Tool slot, Kit ID
Tool Routing
Part identification
NO. of operation , Operation time
Alternative machining center
Tools required, Sequence selection
ToolingStrategy (SPT-SOT-FCFS-FNOP-LTL)
Tool Replacement
Tool routing, Tool Changeover
Machining operation
Alternative tools

Figure 1. FMS shop floor flexibility control functions

The tool magazine capacity is an influential factor in determining the flexibility of the system. A proper tool management is needed to control processing of parts and enhance the flexibility to variety of parts. It is important to design a tool management control function so that the proper tools are available at the right machine at the desired time for processing of scheduled parts.

The work-order processing and part control system essentially drives other control functions. This module concerns the determination of a subset of part types from a set of part types for processing.

A number of criteria can be used for selecting a set of part types for processing (*i.e.* due date, inter arrival time, requirement of tools, operation time, shared operation, operations sequence).

3. FMS Adaptive Controller Architecture

3.1 FMS Configuration Parameters

The following notations and criteria are utilized in developing the rule-based model of the FMS controller addressed in this research. **Table 1** represents the notations.

These parameters are defined in such a way that contains information about the previous control functions on a platform of multi-purpose machines. In other words, definition of these parameters considers machine and tool

Table 1. FMS configuration parameters and performance criteria

Notation	Definition	Notation	Definition
FMS configuration parameters			
P_i	i-th production order, $1 \le i \le p$	OE_{ijkl}	Operative efficiency of O_{ij} on M_{kl}
O_{ij}	j-th operation of order P_i, $1 \le j \le n_i$	DD_{Pi}	Due date of P_i
M/C_k	k-th machining centers, $1 \le k \le m$	T_{ij}	Time for processing operation O_{ij}
M_{kl}	l-th machine of M/C_k, $1 \le l \le L_k$	α_i	Penalty weight for P_i when ACT_i is less than DD_{Pi}
IAT_i	inter arrival time between P_i and P_{i-1}	β_i	Penalty weight for P_i when ACT_i is greater than DD_{Pi}
TMC_{kl}	Tool magazine capacity of M_{kl}	n_i	The number of P_i operations.
TL_{hkl}	Tool life of h-th tool of M_{kl} (time based)	RE_i	The number of operation remain to complete P_i
TM_{hkl}	h-th tool of M_{kl} tool magazine	$MinU$	Minimum utilization
MS_{ij}	Set of machines which can handle O_{ij}	$MaxU$	Maximum utilization
LIB_{kl}	Local input buffer size of M_{kl}	ST_{ij}	Standard time of O_{ij} with 100% operative efficiency
LOB_{kl}	Local output buffer size of M_{kl}	TP	Throughput
PTH	Duration of planning time horizon	RTO_{ij}	Number of required Tool for O_{ij}
t	Current time	ETT_{ij}	Elapsed time between O_{ij} and its latter operation
Simulator outputs performance criteria			
Notation	**Definition**	**Notation**	**Definition**
TBD_{kl}	Time between departures on M_{kl}	TIT_{kl}	Total idle time of M_{kl}
ACT_i	Actual cycle time of order P_i	TWT_i	Total waiting time of P_i
Z_i	Total penalty of P_i	MU_{kl}	Machine M_{kl} utilization
$OS_i(t)$	Set of operations of P_i processed until t	$OS_{kl}(t)$	Set of operations processed on M_{kl} until t
QM_{kl}	Queue size of M_{kl}	CT_{kl}	Completion time in M_{kl}
TU_{hkl}	Tool usage of h-th tool of M_{kl} (time based)	BU_{kl}	Buffer usage of M_{kl}

Table 2. Binary control flags

Variable	Definition	Variable	Definition
OA_{ijkl}	Equal to 1 if O_{ij} is assigned to M_{kl}, otherwise it is equal to 0	MIU_{kl}	Equal to 1 if machine M_{kl} is in use; otherwise it is equal to 0
TML_{hijkl}	Equal to 1 if TM_{hkl} load to perform O_{ij} on M_{kl} and equal to 0 if unloads RTM_h	BSA_{kl}	Equal to 1 if buffer space of M_{kl} is available; otherwise it is equal to 0
PC_i	Equal to 1 if P_i complete otherwise it is equal to 0	MB_{kl}	Equal to 1 if machine M_{kl} is bottleneck; otherwise it is equal to 0
λ_i	Equal to 1 if ACT_i is less than DD_{Pi}, otherwise it is equal to 0	OD_{ijkl}	Equal to 1 if O_{ij} is done on M_{kl}, and depart it; otherwise it is equal to 0
APO_i	Equal to 1 if P_i should be scheduled next otherwise it is equal to 0	PA_i	Equal to 1 if P_i arrive otherwise it is equal to 0
OW_{ij}	*Equal to 1 if O_{ij} is waiting for process; otherwise it is equal to 0*		

flexibility characteristics of an FMS. **Table 2** shows the binary control flags (BCF's).

3.2 Simulation-Based Intelligent Decision Support System

Figure 2 shows the combination between simulation and intelligent decision support system as for FMS adaptive control. The figure shows the cooperation between IDSS and the simulator module. The current configuration parameters of the FMS are read by user interface and are used as the input data to build conceptual model. The simulation model will evaluate the current shop performance, such as actual cycle time, tool and buffer utilization. This process continues until a satisfying and controllable shop floor configuration is reached.

The system presents details of the architecture, components and functions of a FMS decision-making controller. The proposed controller consists of a simulator model coordinate rule based IDSS with a real time mechanism. The simulation output data are fed to the knowledge-based system as input data. The rule-based IDSS analyzes output of simulation model to control the real-time status of FMS. Once the IDSS makes recommendations, the simulation model is adjusted accordingly and the process is repeated. The simulation and IDSS components cooperate with each other until the control goals are achieved. Since the primary objective is to improve the throughput of the shop floor, a simulation analysis assisted by decision process is carried out. The status of the cell, machines, part orders, the availability

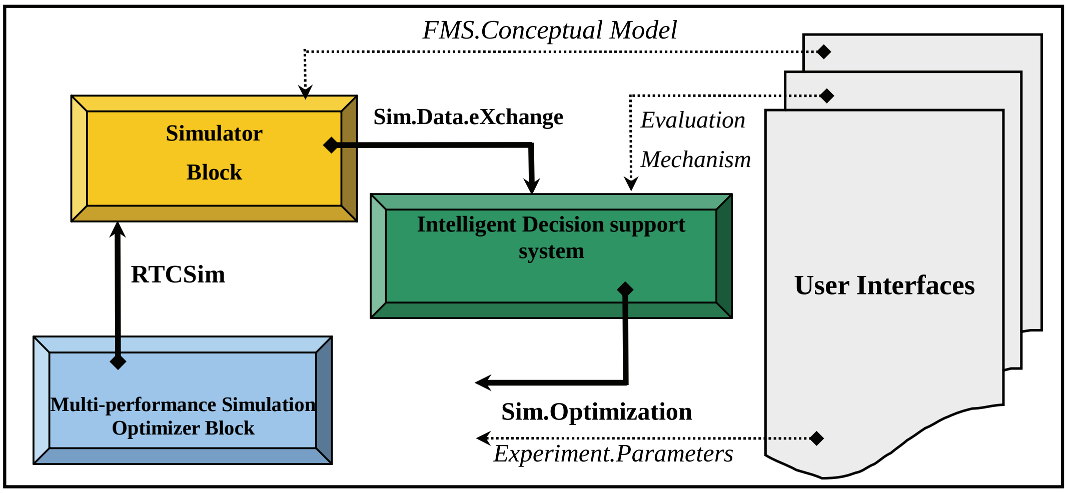


Figure 2. The structure of simulation-based IDSS for FMS adaptive control

of operators and system control flags are recorded in separate databases. Sequence of jobs is used to control the flow of parts through the system. The first step to estimate the performance criteria is assigning the operations to machines and scheduling the operations on each machine.

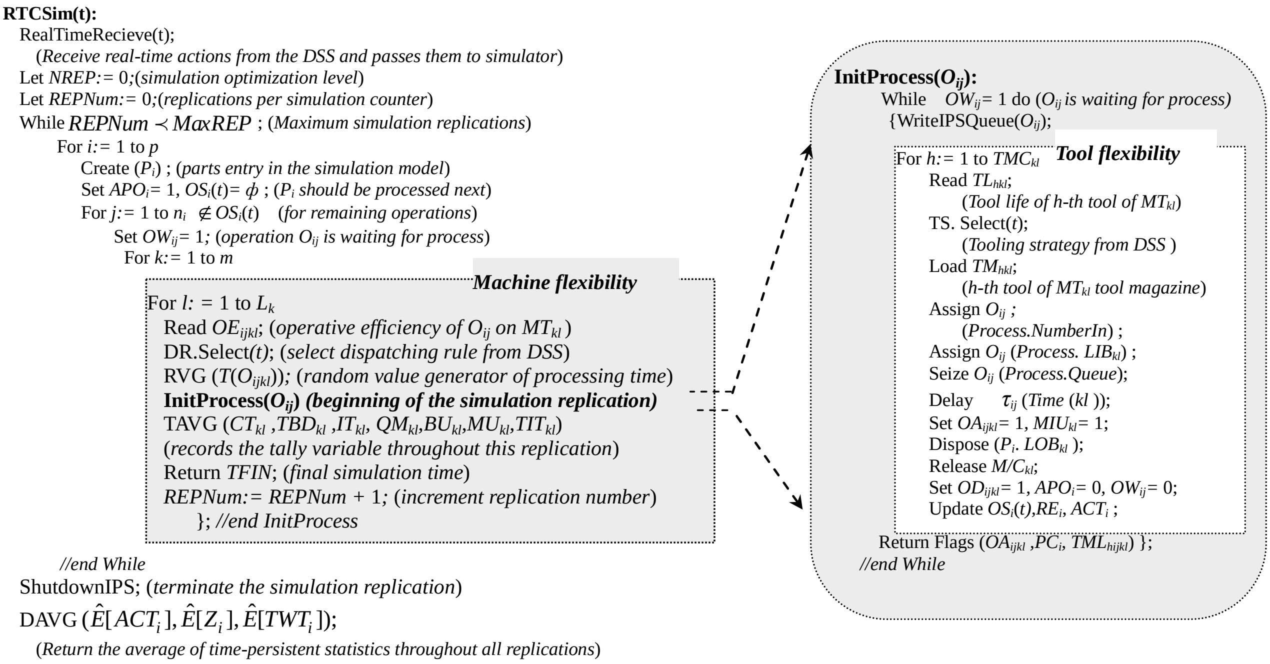
The above posteriori adaptive control mechanism employs a simulator block to predict different performance criteria of the FMS conceptual model. The simulator contains the discrete event simulation model and is able to measure several FMS performance criteria depending on the different inputs. The simulation results are then forwarded between external interfaces belonging to different external models. On the other hand, these interfaces handle the necessary communications with the simulation and coordinate IDSS control signal transformations into the simulator.

Sequence of jobs is used to control the flow of parts through the system. The first step to estimate the performance criteria of FMS is assigning the operations O_{ij} to machining centers and extracting the set $OSM_{kl}(t)$ includes operations processed on M/C_k until t. The real time adaptive control framework is based on affiliating all current events and expected future event to a time tag for process synchronization. The following pseudo-code shows the initialization phase of the simulation in order to configure the FMS conceptual model.

The initialization phase should be run in execution mode using the function *RealTimeInitialize*(t) to synchronize simulation logic with an external process of FMS controller system. The module *RTCSim*(t) represents FMS events simulation to handle machine and tool flexibility.

```
FMSConfigParam( )
    Read Number of parts, machining centers and planning horizon ($p,m,PTH$);
    For $i$: = 1 to $p$ Read Number of parts operations and due date ($n_i$, $DD_{Pi}$, $IAT_i$, $\alpha_i$, $\beta_i$);
    For $k$: = 1 to $m$ Read Number of machines at each machining centers ($L_k$);
    For $k$: = 1 to $m$ initialize Machining Centers Resources $M/C_k$, Capacity;
    For $k$: = 1 to $m$
    For $l$: = 1 to $L_k$ initialize Machine Tools, Tool magazine and buffers ($MT_{kl}$, $TMC_{kl}$, $LIB_{kl}$, $LOB_{kl}$);
    For $i$: = 1 to $p$
    For $j$: = 1 to $n_i$
        Initialize queue used to hold part operations ($O_{ij}$ (Process.Queue));
        Read (processing time of each operation ($O_{ij}$, $T_{ij}$, $ST_{ij}$, $MS_{ij}$, $ETT_{ij}$, $RTO_{ij}$);
  InitTime($t$); (Initialize simulation current time)
  RealTimeInitialize($t$); (Initialize inter-process synchronization)
```

The simulation clock is set to the real-time clock of the operating FMS system and all other simulation processes are initiated by *InitProcess*(O_{ij}). Because of the randomness of processing times in each replication, the expectations of system outputs are estimated by sample means. The function $TAVG(ACT_i, TU_{hkl}, BU_{kl})$ records the values of system outputs throughout each replication and finally estimates the expectation of these statistics



```
RTCSim(t):
  RealTimeRecieve(t);
    (Receive real-time actions from the DSS and passes them to simulator)
  Let NREP:= 0;(simulation optimization level)
  Let REPNum:= 0;(replications per simulation counter)
  While $REPNum \prec MaxREP$ ; (Maximum simulation replications)
      For i:= 1 to p
          Create ($P_i$) ; (parts entry in the simulation model)
          Set $APO_i$= 1, $OS_i(t)$= $\phi$ ; ($P_i$ should be processed next)
          For j:= 1 to $n_i$ $\notin OS_i(t)$    (for remaining operations)
              Set $OW_{ij}$= 1; (operation $O_{ij}$ is waiting for process)
                For k:= 1 to m
                                                              Machine flexibility
                    For l: = 1 to $L_k$
                      Read $OE_{ijkl}$; (operative efficiency of $O_{ij}$ on $MT_{kl}$ )
                      DR.Select(t); (select dispatching rule from DSS)
                      RVG ($T(O_{ijkl})$); (random value generator of processing time)
                      InitProcess($O_{ij}$) (beginning of the simulation replication)
                      TAVG ($CT_{kl}$ ,$TBD_{kl}$ ,$IT_{kl}$, $QM_{kl}$,$BU_{kl}$,$MU_{kl}$,$TIT_{kl}$)
                      (records the tally variable throughout this replication)
                      Return TFIN; (final simulation time)
                      REPNum:= REPNum + 1; (increment replication number)
                          }; //end InitProcess
      //end While
  ShutdownIPS; (terminate the simulation replication)
  DAVG ($\hat{E}[ACT_i], \hat{E}[Z_i], \hat{E}[TWT_i]$);
    (Return the average of time-persistent statistics throughout all replications)
```

```
InitProcess($O_{ij}$):
        While   $OW_{ij}$= 1 do ($O_{ij}$ is waiting for process)
         {WriteIPSQueue($O_{ij}$);
                                                  Tool flexibility
          For h:= 1 to $TMC_{kl}$
              Read $TL_{hkl}$;
                  (Tool life of h-th tool of $MT_{kl}$)
              TS. Select(t);
                  (Tooling strategy from DSS )
              Load $TM_{hkl}$;
                  (h-th tool of $MT_{kl}$ tool magazine)
              Assign $O_{ij}$ ;
                  (Process.NumberIn) ;
              Assign $O_{ij}$ (Process. $LIB_{kl}$) ;
              Seize $O_{ij}$ (Process.Queue);
              Delay    $\tau_{ij}$ (Time (kl ));
              Set $OA_{ijkl}$= 1, $MIU_{kl}$= 1;
              Dispose ($P_i$. $LOB_{kl}$ );
              Release $M/C_{kl}$;
              Set $OD_{ijkl}$= 1, $APO_i$= 0, $OW_{ij}$= 0;
              Update $OS_i(t)$,$RE_i$, $ACT_i$ ;
        Return Flags ($OA_{ijkl}$ ,$PC_i$, $TML_{hijkl}$) };
      //end While
```

through the average function $DAVG(\hat{E}[ACT_i], \hat{E}[TU_{hkl}], \hat{E}[BU_{kl}])$ over *MaxREP* simulation replications. The number of replications per simulation (*MaxREP*) should be set to the minimum number necessary to obtain a reliable estimate of performance criteria.

Based on the results obtained at each level of optimization (*NREP*) and exchanging them with IDSS, additional number of replications may be re-simulated for each design. The expected value of FMS performance criteria are extracted under design $\vec{\rho}_{ijkl}, \vec{OA}_{ijkl}$. The $\hat{E}[ACT_i \mid \vec{\rho}_{ijkl}, \vec{OA}_{ijkl}]$, $\hat{E}[TU_{hkl} \mid \vec{\rho}_{ijkl}, \vec{OA}_{ijkl}]$, $\hat{E}[BU_{kl} \mid \vec{\rho}_{ijkl}, \vec{OA}_{ijkl}]$ represent the stochastic effects of system output by sample mean. The ultimate goal is to find the solution that optimizes the value of these performance criteria. The optimization procedure uses the outputs from the simulation model of previous *NREP* to construct a response surface at each simulation optimization level of $\vec{\rho}_{ijkl}, \vec{OA}_{ijkl} |_{NREP=r}$ and to extract the next level of $\vec{\rho}_{ijkl}, \vec{OA}_{ijkl}$ as an input to the model.

To control the external processes of FMS, the simulator block and IDSS are synchronized via simulation data exchange *Sim.Data.eXchange*(*IDSS*). The IDSS analyzes outputs of simulation model to control the real-time status of FMS after receiving these results by *RealTimeSend*() function. The IDSS then sends appropriate control signals of beginning operation to the corresponding entities when an event is occurred. Proposing the adaptive controller with this structure allows modeling of synchronization mechanism between FMS entities and transmission times for messages exchanged between the IDSS and simulator.

Figure 3 schematically describes the inter-process synchronization between different components of co-simulator. The approach for adaptive controller designing is built around the theory of supervisory control based on exchanging simulation outputs with an event-condition-action real time system. The proposed system uses a posteriori adaptive control mechanism that also is an online control method acting after the event occurs versus such popular reactive control method.

The simulator can trigger the rule-based IDSS to generate the appropriate control policy. The simulator block sends messages to the external rule-based system to indicate simulated results from FMS by *RealTimeSend*(). The rule-based IDSS interprets these results and sends appropriate action messages back to the simulator and user to indicate the instructions to be done.

3.3 Rule Production for FMS Real Time Simulation-Based Controller

The IDSS collect the facts into appropriate data base using *CollectFact*(), which is then used for inference by simulation outputs in feed forwarding reasoning. The control framework is implemented by integration of the adaptive control rules and real time simulator for enforcing dynamic strategies of FMS shop floor control. In order to strengthen the expert system reasoning, knowledge-elicit-

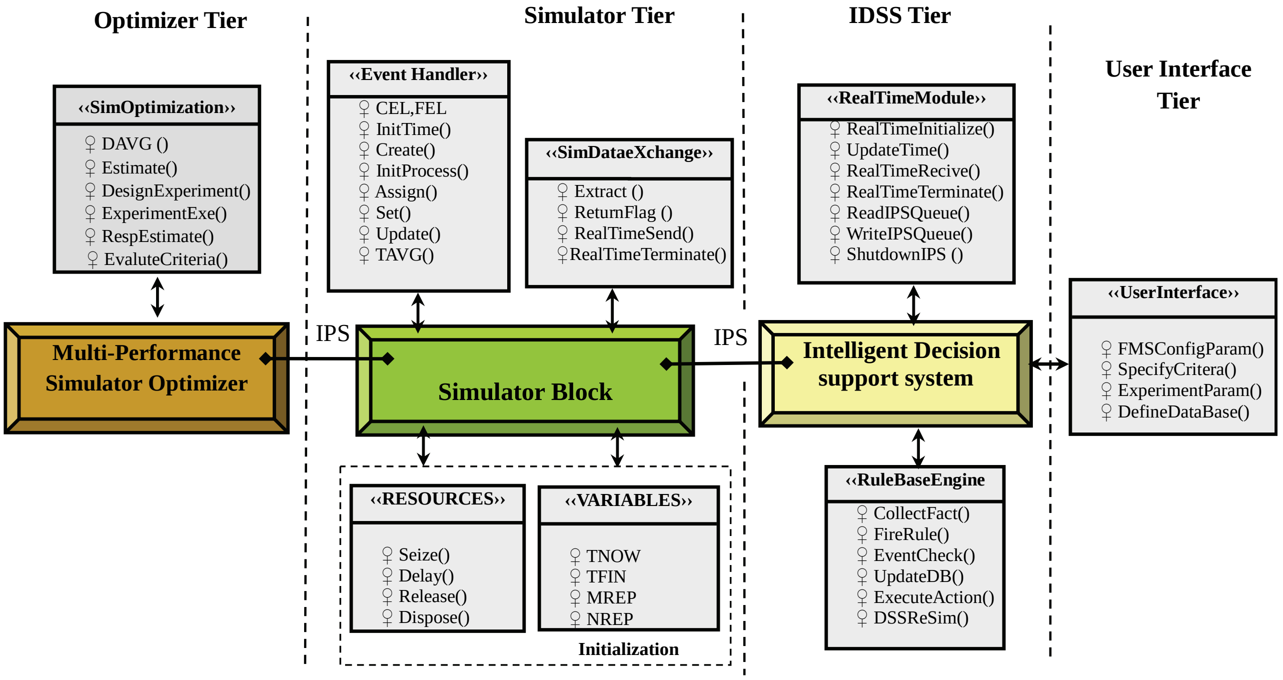


Figure 3. Real time simulation data exchange via inter-process synchronization

tation techniques are used for preventing ineffective redundancy at concurrent firing of rules and high degree of parallelism. This knowledge-based IDSS is aimed at providing a powerful control on different operations of FMS. It acts as a cell manager which may work alongside the operating cell-oriented part and tool management system. These sections describe the knowledge representation through a set of control rules. Design of IDSS controller focuses on the development of appropriate Event-Condition-Action (ECA) rules for tuning control parameters. These rules are formulated by the techniques of data gathering and knowledge elicitation to construct IDSS. The IDSS is able to obtain feedback results from the on-line system of simulator. These results are very significant and let the expert system to re-simulate if the performance criteria are not desirable.

The rules applied in this paper are structured in the following form and consist of three segments: event type, condition and action:

When ‹$Event_1$, $Event_2$,$Event_3$, … ›
If ‹$Condition_1$, $Condition_2$, $Condition_3$, … ›
Then ‹$Action_1$, $Action_2$, $Action_3$,… ›

Event type: This tag specifies that analysis of condition should be done once the events take place.

Condition: This segment of ECA rules specifies a list of conditions. In order to trigger an action rule, all conditions should be satisfied. These conditions refer to a logical assertion of the FMS states extracted by the simulator module *RTCSim*(*t*).

Action: This segment specifies actions which may consist of a list of operations. Whenever an action rule is triggered by an event, the operations being in its action list will be initiated sequentially. The proposed rule-based system for manufacturing execution system provides the parts sequence list to the multi-purpose machines available and then the operation assignment and task proportions of parts on related machines. The output can be manipulated by changing the rules and strategies entered at the expert system query stage. **Table 3** illustrates MES control function about dispatching rules.

For each part P_i the slack index is defined as: $Slack_i = DD_{P_i} - \sum_{j=1}^{n_i} ST_{ij} - t, \forall i$. The function *Sort*(*array*) finds the maximum or minimum value in the array and the binary flag APO_i specifies the next scheduled part. **Table 4** illustrates MES control function for machining rules in the FMS.

The binary flag OA_{ijkl} specifies the assignment of operation O_{ij} to machine M_{kl}. **Table 5** illustrates MES control function for tooling strategy in the FMS.

Operative efficiency of doing operation O_{ij} on M_{kl} is defined as OE_{ijkl} and thus tool usage can be considered as $TU_{hkl} = T_{ij} \times (1 + OE_{ijkl}); \forall h,k,l$. For each executable operation O_{ij}, the proportion of O_{ij} performed on M_{kl} is denoted as $\rho_{ijkl}(t)$, $0 \le \rho_{ijkl} \le 1$. IDSS monitors all events and states transition of FMS by considering ρ_{ijkl} to

Table 3. MES control function (dispatching rules)

MES Control Function: Dispatching Rules			
Dispatching Rule	**When [Event]**	**If (Condition)**	**Action**
Shortest Processing Time	$t \neq 0$ *RealTimeRecieve()*	$DR.Select(t) = SPT$	$Sort(ST_{ij})\ \forall i\ \forall j;\ \ APO\ _i = 1;$
First Come First Serve	$t \neq 0$ *RealTimeRecieve()*	$DR.Select(t) = FCFS$	$Sort(IAT_i)\ \forall i;\ \ APO\ _i = 1;$
Operation with Least Slack	$t \neq 0$ *RealTimeRecieve()*	$DR.Select(t) = SLACK$ $PC_i = 0\ \&\&\ Slack_i \prec 0$	$Sort(\lvert Slack_i \rvert)\ \forall i;\ \ APO\ _i = 1;$
Slack Per Remaining Work	$t \neq 0$ *RealTimeRecieve()*	$DR.Select(t) = S\ /\ RMOP$ $PC_i = 0\ \&\&\ Slack_i \prec 0$	$Sort(\lvert Slack_i \rvert / RE_i)\ \forall i \forall j;\ \ APO\ _i = 1;$
Slack Per Remaining Work	$t \neq 0$ *RealTimeRecieve()*	$DR.Select(t) = S\ /\ RMWK$ $PC_i = 0\ \&\&\ Slack_i \prec 0$	$Sort(\lvert Slack_i \rvert / \sum_j ST_{ij})\ \forall i \forall j;\ \ APO\ _i = 1;$
Earliest Due Date	$t \neq 0$ *RealTimeRecieve()*	$DR.Select(t) = EDD$	$Sort(DD_{Pi})\ \forall i;\ \ APO\ _i = 1;$

Table 4. MES control function (machining rules)

MES Control Function: Machining Rules			
Machining Rule	**When [Event]**	**If (Condition)**	**Action**
Random Selection	$t \neq 0$ *RealTimeRecieve()*	$MR.Select(t) = RAN$ $APO\ _i = 1$	$Sort(Rand\ M_{kl})\ \forall k\ \forall l;\ \ OA\ _{ijkl} = 1;$
Shortest Queue Length	$t \neq 0$ *RealTimeRecieve()*	$MR.Select(t) = SQL$ $APO\ _i = 1$	$Sort(QM_{kl})\ \forall k\ \forall l;\ \ OA\ _{ijkl} = 1;$
Lowest Utilized Buffers	$t \neq 0$ *RealTimeRecieve()*	$MR.Select(t) = LUB$ $APO\ _i = 1$	$Sort(BU_{kl})\ \forall k\ \forall l;\ \ OA\ _{ijkl} = 1;$

Table 5. MES control function (tooling strategy)

MES Control Function: Tooling Strategy			
Tooling Strategy =TS	**When [Event]**	**If (Condition)**	**Action**
Shortest Operation Time	$t \neq 0$ *RealTimeRecieve()*	$TS.Select(t) = SOT$ $OA\ _{ijkl} = 1$	$Sort(ST_{ij})\ \forall i\ \forall j;\ \ AssignTool(TM_{hkl}, O_{ij})\ \forall i\ \forall j;$ $UpdateToolMag\ (M_{kl});$
Shortest Processing Time	$t \neq 0$ *RealTimeRecieve()*	$TS.Select(t) = SPT$ $OA\ _{ijkl} = 1$	$Sort(DD_{Pi})\ \forall i;\ \ AssignTool\ (TM_{hkl}, P_i)\ \forall i;$ $UpdateToolMag\ (M_{kl});$
First Come First Serve	$t \neq 0$ *RealTimeRecieve()*	$TS.Select(t) = FDFS$ $OA\ _{ijkl} = 1$	$Sort(IAT_i)\ \forall i;\ \ AssignTool(TM_{hkl}, P_i)\ \forall i\ \forall j;$ $UpdateToolMag\ (M_{kl});$

dynamically rebuild new configuration and replicate simulation module *RTCSim*(t). **Table 6** contains the rules for control of transition of different states in FMS, bottleneck detection and resolving, assigning operation to a non-bottleneck machining centers.

For each part P_i actual cycle time is defined as: $ACT_i = \sum_{j \in OS_{kl}} \left(ETT_{ij} + \frac{\rho_{ijkl} \times ST_{ij}}{OE_{ijkl}} \right)$ and the penalty is defined as:

$$Z_i = \sum_{i=1}^{p} \left[\alpha_i \lambda_i (DD_{P\,i} - ACT_i) + \beta_i (1 - \lambda_i)(ACT_i - DD_{P\,i}) \right].$$

4. Experimental Study

The problem presented has been adopted in this paper to validate the proposed method by Sarin and Chen [43]. The model presents machine loading and tool allocation problem in FMS with tool life and magazine capacity. The FMS model includes tool and machine alternatives. The experiment was done on a FMS with four machining centers. **Tables 7** and **8** show tool-operation and machine -tool compatibility.

Table 9 represents the machining time of operations on alternative tools.

Table 6. MES control function

MES Control Function: States, Bottleneck, Assigning			
When [Event]	**If (Condition)**	**Action**	
$t \neq 0$ $PA_i = 1; \forall i$	$MIU_{kl} = 0; \forall k \forall l$	$Initialization(initial\ config.\ parameters);$ $DefineDB(); SpecifyCriteria(); RTCSim(t);$ $UpdateTime(t);\ SimDataeXchange(IDSS);$	**States transition control rules**
$t \neq 0$ *ReadIPSQueue(O_{ij})*	$MB_{kl} = 0; \exists! k,l\ ,\ \ PA_i = 1$	RealTimeRecieve $(OS_{kl}(t), OS_i(t), \rho_{ijkl}(t), BCF);$	
$t \neq 0$ *ReadIPSQueue(O_{ij})*	$MIU_{kl} = 1; \forall k,l \quad BSA_{kl} = 0;\ PC_i = 0$	RealTimeRecieve $(OS_{kl}(t), OS_i(t), \rho_{ijkl}(t), BCF);$ $UpdateTime(t);$	
$t \neq 0$ *ReadIPSQueue(O_{ij})*	$MIU_{kl} = 0; \exists k,l \quad BSA_{kl} = 1$ $OA_{ijkl} = 1; \forall j \notin OS_i(t);\ PC_i = 0$	$MIU_{kl} = 1;$ RealTimeRecieve $(OS_{kl}(t), OS_i(t), \rho_{ijkl}(t), BCF);$ $UpdateTime(t);$	
$t \neq 0$ *ReadIPSQueue(O_{ij})*	$MIU_{kl} = 0; \exists k,l \quad BSA_{kl} = 0;\ PC_i = 0$	RealTimeRecieve $(OS_{kl}(t), OS_i(t), \rho_{ijkl}(t), BCF);$ $UpdateTime(t);$	
$t \neq 0$ *RealTimeRecieve()*	$n[OS_{kl}(t)] \succ (\sum_{i=1}^{p} n_i) / L_K; \exists k,l$	$MB_{kl} = 1$	**Bottleneck detection**
$t \neq 0$ *RealTimeRecieve()*	$TBD_{k1} \succ \left(PTH / MinU \right); \forall k\ \ ,\ \ MIU_{kl} = 1$	$MB_{kl} = 1$	
$t \neq 0$ *RealTimeRecieve()*	$[(TBD_{kl} \succ TBD_{k(l-1)})\ \&\&(U_{kl} \prec U_{k(l-1)})]$ $MIU_{kl} = 1$	$MB_{kl} = 1$	
$t \neq 0$ *RealTimeRecieve()*	$MU_{kl} \prec MinU; \forall k,l \qquad MIU_{kl} = 1$	$MB_{kl} = 1$	
$t \neq 0$ *RealTimeRecieve()*	$MB_{kl} = 0\ \&\&\ MIU_{kl} = 0; \exists k,l$ $OA_{ijkl} = 1; \forall j \notin OS_i(t)$	$MIU_{kl} = 1;$ RealTimeSend $(OS_{kl}(t), OS_i(t), \rho_{ijkl}(t), BCF);$ $UpdateTime(t);\ SimDataeXchange(IDSS);$	**Assigning operation to non-bottleneck**
$t \neq 0$ *RealTimeRecieve()*	$PC_i = 0; \forall i$ $MB_{kl} = 1$	$OA_{ijkl'} = 1; \forall l' \neq l$ RealTimeSend $(OS_{kl}(t), OS_i(t), \rho_{ijkl}(t), BCF);$ $UpdateTime(t); SimDataeXchange(IDSS);$	
$t \neq 0$ *RealTimeRecieve()*	$(MU_{kl} \prec MU_{k'l'}; \exists k,k',l,l') \Vert (MU_{kl} \prec MinU);$ $MU_{kl} = MU_{k'l'} = 0; \exists k,k',l,l'; PC_i = 0$	$OA_{ijkl} = 1; \forall j \notin OS_i(t); \quad MIU_{kl} = 1$ RealTimeSend $(OS_{kl}(t), OS_i(t), \rho_{ijkl}(t), BCF);$ $UpdateTime(t);\ SimDataeXchange(IDSS);$	

Table 7. Tool-operation compatibility

Part/Tool		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P_1	**O_{11}**	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	O_{12}	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	O_{13}	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	O_{14}	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
P_2	**O_{21}**	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	O_{22}	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
	O_{23}	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
	O_{24}	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P_3	**O_{31}**	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
	O_{32}	0	0	0		0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
	O_{33}	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
	O_{34}	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P_4	**O_{41}**	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	O_{42}	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	O_{43}	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
	O_{44}	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Tool life		**25**	**21**	**25**	**20**	**22**	**25**	**25**	**22**	**20**	**25**	**18**	**20**	**21**	**25**	**17**	**20**	**20**	**21**	**22**	**24**

Table 8. Machine-tool compatibility

Machine/Tool	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
M_{11}	1	0	1	0	1	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0
M_{21}	1	0	1	1	0	1	1	0	0	1	1	0	0	1	0	0	0	0	1	1
M_{31}	0	1	0	1	0	1	0	1	1	0	0	1	0	0	0	0	1	0	0	0
M_{41}	0	0	0	0	1	0	0	0	1	0	1	0	1	0	1	0	0	1	0	0

Table 9. Machining time on alternative tools (T_{ij}) for parts

Part No	P_1							P_2							
Operation No	O_{11}		O_{12}		O_{13}	O_{14}		O_{21}		O_{22}		O_{23}		O_{24}	
Tool No	1	2	4	7	6	10	13	1	3	8	16	10	17	4	12
M_{11}	104			68		84		114	114		25	106			96
M_{21}	110		120	130	110	76		126	98			66		116	
M_{31}		101	106		118								29	112	84
M_{41}							100			119					
Standard Time	**95**	**98**	**105**	**60**	**104**	**72**	**95**	**112**	**91**	**115**	**18**	**60**	**20**	**107**	**82**

Part No	P_3								P_4							
Operation No	O_{31}		O_{32}		O_{33}		O_{34}		O_{31}		O_{32}		O_{33}		O_{34}	
Tool No	12	15	9	18	11	19	3	14	2	4	5	20	13	14	7	8
M_{11}	67						82				137				68	
M_{21}					117	47	85	110		114		38		115	53	
M_{31}	102		90						49	140						87
M_{41}		134	120	40	132						118		120			
Standard Time	**60**	**127**	**84**	**30**	**115**	**40**	**50**	**100**	**42**	**109**	**113**	**35**	**115**	**115**	**53**	**85**

Table 10. Machining efficiency for parts on alternative tools

Order No	P_1							P_2							
Operation No	O_{11}		O_{12}		O_{13}	O_{14}		O_{21}		O_{22}		O_{23}		O_{24}	
Tool No	1	2	4	7	6	10	13	1	3	8	16	10	17	4	12
OE_{11}(%)	91			88		86		98	80		70	56			85
OE_{21}(%)	86		88	46	95	95		89	93			91		91	
OE_{31}(%)		97	99		88								69	95	98
OE_{41}(%)							95			96					
Standard Time	**95**	**98**	**105**	**60**	**104**	**72**	**95**	**112**	**91**	**115**	**18**	**60**	**20**	**107**	**82**

Order No	P_3								P_4							
Operation No	O_{31}		O_{32}		O_{33}		O_{34}		O_{41}		O_{42}		O_{43}		O_{44}	
Tool No	12	15	9	18	11	19	3	14	2	4	5	20	13	14	7	8
OE_{11}(%)	90						61				82				78	
OE_{21}(%)					98	85	59	91		96		92		100	100	
OE_{31}(%)	59		93						86	78						98
OE_{41}(%)		95	70	75	87						96		96			
Standard Time	**60**	**127**	**84**	**30**	**115**	**40**	**50**	**100**	**42**	**109**	**113**	**35**	**115**	**115**	**53**	**85**

Table 11. Operation assignment and task proportion (ρ_{ijkl}) and tool load

Part/Machine		M_{11}	M_{21}	M_{31}	M_{41}
P_1	O_{11}	0.27 (1)		0.73 (2)	
	O_{12}	0.81 (7)		0.19 (4)	
	O_{13}		0.78 (6)	0.22 (6)	
	O_{14}	0.12 (10)	0.88 (10)		
P_2	O_{21}	0.89 (1)	0.11 (3)		
	O_{22}	0.79 (16)			0.21 (8)
	O_{23}		0.75 (10)	0.25 (17)	
	O_{24}	0.22 (12)		0.78 (12)	
P_3	O_{31}	0.91 (12)		0.09 (12)	
	O_{32}			0.79 (9)	0.21 (18)
	O_{33}		1 (19)		
	O_{34}	0.35 (3)	0.65 (14)		
P_4	O_{41}		0.66 (4)	0.34 (2)	
	O_{42}		0.48 (20)		0.52 (5)
	O_{43}		1 (12)		
	O_{44}	0.45 (7)	0.33 (7)	0.22 (8)	

Table 12. Difference between the proposed method and the heuristic method of [43]

	Total Actual Cycle Time	Total Idle Time	Total Time between Departure	Total Waiting Time	Penalty
Proposed system	2703	441	35	127	48.5 (Earliness)
Classis mathematical method	3108	731	63	463	154 (Tardiness)

Table 13. Statistical analysis of difference between the proposed and mathematical method

		Actual Cycle Time	Idle Time	Time between Departure	Waiting Time
Sample Size = 384	**Mean**	2705.68	442.936	34.360	128.226
	StDev	16.68	9.932	2.029	10.436
	SEMean	0.85	0.507	0.104	0.533
	T-Value	–472.54	–568.36	–276.55	–628.64
	P-Value	0.000	0.000	0.000	0.000

Table 10 represents machining efficiency of operation allocation on alternative tools.

It is assumed that due date (*DD* = 2800), $\alpha_i = \beta_i = 0.5$, and $LIB_{kl} = LOB_{kl} = 15$. Tool magazine capacity and tool life are considered 20 and 100, respectively. Manufacturing execution system also includes dispatching rules (*SPT*), tooling strategies (*FCFS*) and machining rules (*SQL*). **Table 11** shows the operation assignment and task proportion according to the rules of the proposed method.

Table 12 represents the difference of total actual cycle time, total idle time, total time between departures and total waiting time between the proposed rule-based system and the mathematical method.

The solution obtained from proposed method creates a balanced and controlled actual cycle time on machining centers. The proposed approach outperforms the heuristic method in terms of the total actual cycle time, total idle time, total time between departures and total waiting time. The proposed system presents 48.5 units of earliness penalty despite the 154 unit of tardiness penalty of mathematical method. To show the effects of difference between the proposed method outputs and classic mathematical method, statistical analysis is given as shown in **Table 13**.

The aforementioned results verify and validate the FMS shop floor links to the supervisory control of machine and tool flexibility. Different scenarios of performance criteria levels demonstrate effectiveness of the proposed method for the system control. The proposed method is also efficient in terms of the computation time which is highly important for the real time control of a manufacturing system. The proposed real-time simulation-based intelligent decision support system provides a real time control mechanism for improving performance of a flexible manufacturing shop floor.

5. Conclusions

This paper presents an intelligent decision support system to tackle the production control of a FMS. Development of the present knowledge-based system is aimed at integrating an ECA rule-based system and a simulator module to ease the cell adaptive supervisory control. A novel architecture of this intelligent adaptive controller prototype which is based on a real-time simulator core has been developed and presented to validate the proposed approach.

The FMS shop floor data are gathered and stored into the appropriate databases over time. The adaptive control mechanism employs a real time discrete event simulator to predict performance of the given system during the remaining time of planning horizon. The current state of the FMS performance criteria from the simulator is then stored on the appropriate databases. The proposed method provides an applicable and efficient framework for real-time control of the shop floor in flexible manufacturing system. The criteria considered to measure performance of the system shows that the proposed approach is effective and efficient in controlling shop floor. The main contributions of this paper can be summarized as follows.

1) Designing real time ECA rules according to feed forward reasoning with the high degree of granularity.

2) Reinforcement of the expert system reasoning technique using data mining and knowledge-elicitation techniques.

3) Proposed method constitutes the framework of adaptive controller supporting the co-ordination and co-operation relations by integrating a real time simulator and an IDSS for implementing dynamic strategies.

4) Avoiding ineffective redundancy at concurrent firing of rules and high degree of parallelism

5) The simulation based IDSS uses a posteriori adap-

tive control mechanism that also is an online control method acting after the event occurs versus such popular reactive control method.

As a result, the proposed system is suitable for different control frameworks on an existing flexible manufacturing system considering the physical constraints and the production objectives. Furthermore, the system illustrates the potential of using the intelligent rule-based DSS for adaptive control of modern industrial plants. Future researches may concentrate on the application of other types of flexibility in shop floors using simulation-based predictive controllers.

REFERENCES

[1] J. A. Buzacott and D. D. Yao, "Flexible Manufacturing Systems: A Review of Analytical Models," *Management Science*, Vol. 32, No. 7, 1986, pp. 890-905.

[2] J. R. Dixon, "Measuring Manufacturing Flexibility: An Empirical Investigation," *European Journal of Operational Research*, Vol. 60, 1992, pp. 131-143.

[3] R. Jaikumar, "Flexible Manufacturing Systems: Management Perspective," Division of Research, Harvard Business School, 1984.

[4] J. S. Edghill and A. Davies, "Flexible Manufacturing Systems—The Myth and Reality," *International Journal of Advanced Manufacturing Technology*, Vol. 1, No. 3, 1985, pp. 37-54.

[5] M. D. Byrne and P. Chutima, "Real-Time Operational Control of an FMS with Full Routing Flexibility," *International Journal of Production Economics*, Vol. 51, No. 2, 1997, pp. 109-113.

[6] Y. M. Moon, "Reconfigurable Machine Tool Design," In: A. I. Dashchenko, Ed., *Reconfigurable Manufacturing Systems and Transformable Factories*, Springer, 2006, pp. 112-139.

[7] R. Tawegoum, E. Castelain and J. C. Gentina, "Real-Time Piloting of Flexible Manufacturing Systems," *European Journal of Operational Research*, Vol. 78, No. 2, 1994, pp. 252-261.

[8] K. E. Stecke and L. Kim, "A Flexible Approach to Part Type Selection in Flexible Flow Systems Using Part Mix Ratios," *International Journal of Production Research*, Vol. 29, No. 1, 1991, pp. 53-75.

[9] C. Basnet and J. H. Mize, "Scheduling and Control of Flexible Manufacturing Systems: A Critical Review," *International Journal of Computer Integrated Manufacturing*, Vol. 7, No. 6, 1994, pp. 340-355.

[10] J. Ayel, "Supervising Conflicts in Production Management," *International Journal of Computer Integrated Manufacturing*, Vol. 8, No. 1, 1995, pp. 54-63.

[11] H. Seifoddini and J. Zhang, "Application of Simulation and Petri Net Modelling in Manufacturing Control Systems," *International Journal of Production Research*, Vol. 34, No. 1, 1996, pp. 191-207.

[12] E. Szelke and L. Monostori, "Reactive Scheduling in Real-Time Production Control," *Modeling Manufacturing Systems*, Springer, New York, 1999.

[13] Z. Guo, W. Wong, S. Leung, J. Fan and S. Chan, "A Genetic-Algorithm-Based Optimization Model for Scheduling Flexible Assembly Lines," *International Journal of Advanced Manufacturing Technology*, Vol. 36, No. 1-2, 2006, pp. 156-168

[14] D. J. Van der Zee, "Modeling Decision Making and Control in Manufacturing Simulation," *International Journal of Production Economics*, Vol. 100, No. 1, 2006, pp. 155-167.

[15] F. T. S. Chan, R. Bhagwat and S. Wadhwa, "Comparative Performance Analysis of a Flexible Manufacturing System (FMS): A Review-Period-Based Control," *International Journal of Production Research*, Vol. 46, No. 1, 2006, pp. 1-24.

[16] G. Habchi and C. Berchet, "A Model for Manufacturing Systems Simulation with a Control Dimension," *Simulation Modelling Practice and Theory*, Vol. 11, No. 1, 2003, pp. 21-44.

[17] B. P. Douglass, "Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems," Addison-Wesley, 2003.

[18] L. Yao, W. Browne, I. Postlethwaite, T. Ozen, P. Atack, M. Mar and S. Lowes, "Architecture for Intelligent Knowledge-Based Supervisory Control of Rolling Mills," *IFAC Workshop on New Technologies for Automation of Metallurgical Industry*, Shanghai, China, 2003, pp. 162-167.

[19] G. Guariso, M. Hitz and H. Werthner, "An Integrated Simulation and Optimization Modelling Environment for Decision Support," *Decision Support Systems*, Vol. 16, No. 2, 1996, pp. 103-117.

[20] C. Gertosio, N. Mebarki and A. Dussauchoy, "Modeling and Simulation of the Control Framework on a Flexible Manufacturing System," *International Journal of Production Economics*, Vol. 64, No. 1-3, 2000, pp. 285-293.

[21] J. W. Fowler and O. Rose, "Grand Challenges in Modeling and Simulation of Complex Manufacturing Systems," *SIMULATION—Transactions of the Society for Modelling and Simulation International*, Vol. 80, No. 9, 2004, pp. 469-476.

[22] F. T. S. Chan and H. Chan, "A Comprehensive Survey and Future Trend of Simulation Study on FMS Scheduling," *Journal of Intelligent Manufacturing*, Vol. 15, No. 1, 2004, pp. 87-102.

[23] G. R. Drake, J. S. Smith and B. A. Peters, "Simulation as a Planning and Scheduling Tool for Flexible Manufacturing Systems," *Proceedings of the* 1995 *Winter Simulation Conference*, 1995, pp. 805-812.

[24] F. T. S. Chan, H. K. Chan and H. C. W. Lau, "The State of the Art in Simulation Study on FMS Scheduling: A Comprehensive Survey," *International Journal of Ad-*

vanced Manufacturing Technology, Vol. 19, No. 11, 2002, pp. 830-849.

[25] S. Chong, A. Sivakumar and R. Gay, "Simulation-Based Scheduling for Dynamic Discrete Manufacturing," *Proceedings of the* 2003 *Winter Simulation Conference*, New Orleans, Louisiana, USA, 2003.

[26] R. W. Brennan and O. William, "Performance Analysis of a Multi-Agent Scheduling and Control System for Manufacturing," *Production Planning Control*, Vol. 15, No. 2, 2004, pp. 225-235.

[27] R. Schelasin and J. Mauer, "Creating Flexible Simulation Models," *IEE Solutions*, Vol. 5, 1995, pp. 50-67.

[28] A. Anglani, A. Grieco, M. Pacella and T. Tolio, "Object-Oriented Modeling and Simulation of Flexible Manufacturing Systems: A Rule-Based Procedure," *Simulation Modelling Practice and Theory*, Vol. 10, No. 3-4, 2002, pp. 209-234.

[29] D. Arnott and G. Pervan, "A Critical Analysis of Decision Support Systems Research," *Journal of Information Technology*, Vol. 20, No. 2, 2005, pp. 67-87

[30] R. W. Brennan, "Towards Real-Time Distributed Intelligent Control: A Survey of Research Themes and Applications," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 37, No. 5, 2007, pp. 744-765.

[31] G. E. Vieira, J. W. Herrmann and E. Lin, "Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods," *Journal of Scheduling*, Vol. 6, No. 1, 2003, pp. 39-62.

[32] J. A. Ventura, F. F. Chen and C. H. Wu, "Grouping Parts and Tools in FMS Production Planning," *International Journal of Production Research*, Vol. 28, No. 6, 1990, pp. 1039-1056.

[33] R. Beach, A. Muhlemann, D. Price, A. Paterson and J. Sharp, "A Review of Manufacturing Flexibility," *European Journal of Operational Research*, Vol. 122, No. 1, 2000, pp. 41-57.

[34] P. Kouvelis, "An Optimal Tool Selection Procedure for the Initial Design Phase of a Flexible Manufacturing System," *European Journal of Operational Research*, Vol. 55, No. 2, 1991, pp. 201-210.

[35] P. Tomek, "Tooling Strategies Related to FMS Management," *FMS Magazine*, Vol. 4, 1986, pp. 102-107.

[36] D. Veeramani, D. Upton and M. Barash, "Cutting Tool Management in Computer Integrated Manufacturing," *The International Journal of Flexible Manufacturing Systems*, Vol. 3, No. 4, 1992, pp. 237-265.

[37] M. Wahab, "Measuring Machine and Product Mix Flexibilities of a Manufacturing System," *International Journal of Production Research*, Vol. 43, No. 18, 2005, pp. 3773-3786.

[38] J. Browne, D. Dubois, K. Rathmill, S. Sethi and K. Stecke, "Classification of Flexible Manufacturing Systems," *The FMS Magazine*, Vol. 2, No. 2, 1984, pp. 114-117.

[39] J. A. Buzacott, "The Fundamental Principles of Flexibility in Manufacturing Systems," *Proceedings of the* 1*st International Conference on Flexible Manufacturing Systems*, 1982, pp. 13-22.

[40] S. K. Das and P. Nagendra, "Investigation into the Impact of Flexibility on Manufacturing Performance," *International Journal of Production Research*, Vol. 31, No. 10, 1993, pp. 2337-2354.

[41] A. Gray, A. Seidmann and K. Stecke, "A Synthesis of Decision Models for Tool Management in Automated Manufacturing," *Management Science*, Vol. 39, No. 5, 1993, pp. 549-567.

[42] N. Buyurgan, C. Saygina and E. Kilic, "Tool Allocation in Flexible Manufacturing Systems with Tool Alternatives," *Robotics and Computer-Integrated Manufacturing*, Vol. 20, No. 4, 2004, pp. 341-349.

[43] S. C. Sarin and C. S. Chen, "The Machine Loading and Tool Allocation in a Flexible Manufacturing System," *International Journal of Production Research*, Vol. 25, No. 7, 1987, pp. 1081-1094.

J. Software Engineering & Applications, **2010, 3, 674-682**
doi:10.4236/jsea.2010.37077 Published Online July 2010 (http://www.SciRP.org/journal/jsea)

Heuristic Approaches for Cell Formation in Cellular Manufacturing

Shahram Saeedi[1], Maghsud Solimanpur[2], Iraj Mahdavi[1], Nikbakhsh Javadian[1]

[1]Department of Industrial Eng., Mazandaran University of Science and Technology, Babol, Iran; [2]Faculty of Engineering, Urmia University, Urmia, Iran.
Email: shahram.saeedi@gmail.com, irajarash@rediffmail.com, nijavadian@ustmb.ac.ir, m.solimanpur@urmia.ac.ir

Received April 22nd, 2010; revised May 16th, 2010; accepted May 22nd, 2010.

ABSTRACT

Cellular Manufacturing System (CMS) is an application of Group Technology (GT) that allows decomposing a manufacturing system into subsystems. Grouping the machines and parts in a cellular manufacturing system, based on similarities is known as cell formation problem (CFP) which is an NP-hard problem. In this paper, a mathematical model is proposed for CFP and is solved using the Ant Colony Optimization (ACO), Genetic Algorithm (GA) and Simulated Annealing (SA) meta-heuristic methods and the results are compared. The computational results show that the GA method is more effective in solving the model.

Keywords: *Cell Formation Problem, Ant Colony Optimization, Genetic Algorithm, Simulated Annealing, Sequence Data, Production Volume*

1. Introduction

Cellular Manufacturing System (CMS) is an application of the Group Technology (GT) philosophy that allows decomposing a manufacturing system into subsystems which makes its management easier than the entire manufacturing system. It has been shown that CMS is an accepted solution to the problem of productivity in batch production which includes a large portion of world manufacturing [1]. The main idea in CMS is the principle of "Similar things should be done similarly" which means the similar manufacturing processes should be identified and grouped in dedicated manufacturing cells.

Manufacturing systems employing CMS can improve the productivity to a large extent. It has been found that CMS can increase the productivity of manufacturing system by three major factors [2]:

- improvement in quality of the work-force,
- increase in the availability of capital,
- improvement in the production technology.

Based on the simulation results performed by Morris and Tersine [3], the superiority of CM over batch production is significant especially when the setup/operation ratio is high, demand is stable, one-way intercellular flows and considerable materials handling are concerned.

Due to different solution approaches, different grouping solutions may be proposed for a certain problem. Therefore there should be some criteria to compare these solutions and choose the best one. There are several objectives to measure the effectiveness of CMS such as:

- Minimum number of intercellular/intracellular moves,
- Greatest proportion of part operations performed within a single cell,
- Maximum machine utilization,
- Minimal total costs by reducing set-up times, and WIP (Work-in-Process),
- Minimal capital investment,
- Minimum number of voids in the cells.

With respect to the benefits mentioned above, CMS has attracted the attention of researchers for the last decades. Some researches related to the work presented in this paper are reviewed in the following.

Burbidge [4] defines group technology as: "an approach to the organization of work in which the organizational units are relatively independent groups, each responsible for the production of a given family of products". In this approach, the main goal is to form manufacturing groups in which, some machines are located in dedicated cells associated with some similar parts based on a machine-part incidence matrix. In each cell, some operations are done on the parts by machines, so that the main objective is to maximize the intra-cell operations, and to minimize the number of inter-cell movements (exceptional elements). It is shown that the machine-part cell formation (MPCF) is a NP-hard problem [5]. Therefore, it takes a long time to obtain an optimal solution for medium-sized problems while it is computationally intractable for large-sized problems. Thus, development and application of heuristic techniques has attained the interest of researchers in this area.

Joines *et al*. [6] offered a classification of the techniques available for manufacturing cell formation. Individual techniques are aggregated into methodological groups including array-based clustering, hierarchical clustering, non-hierarchical clustering, graph theoretic approach, artificial intelligence, mathematical programming, and heuristic approaches.

Table 1 provides a review of the researches related to the current work in terms of the solution approach or problem perspective. The works pointed out in this table suffer from at least one of the following drawbacks:

1) Intercellular movements have been calculated regardless of production volume though it is directly affected by this parameter.

2) Sequence of operations has only been taken into account in the calculation of similarity between the parts. However, this parameter directly affects the number of movements of parts between the cells.

3) In a large number of researches, the total number of “ones” fell out of diagonal blocks is considered as a measure of the number of intercellular movements between the cells. However, this value is seriously dictated by the sequence through which parts are processed. Suppose a certain operation of a part is processed out of the associated cell. If this is the first or the last operation of the part, a single intercellular movement takes place whereas it is counted twice in otherwise. The mathematical model attempted in this paper provides a formula to calculate the intercellular movements in this way.

In this paper, a mathematical model is proposed for solving the cell formation problem, and the model is solved using Genetic Algorithm (GA), Simulated Annealing (SA) and Ant Colony Optimization (ACO). Performance of these methods is compared using two examples selected from the literature. The comparison shows

Table 1. Summary of literature review

Reference	Applied Methodology	Sequence of operation	Production Volume	Exceptional Elements (Voids)	Intercellular Movements
Islier [7]	Ant algorithm	No	No	No	No
Prabhaharan *et al.* [8]	Ant algorithm	Yes	Yes	No	Yes
Mak *et al*. [9]	Ant algorithm	Yes	No	No	No
Spiliopoulos and Sofianopoulou [10]	Ant algorithm	Yes	No	No	Yes
Kesen *et al*. [11]	Ant algorithm	Yes	No	No	No
Satolgu and Suresh [12]	Goal Programming	No	No	No	No
Kao and Fu [13]	Clustering Algorithm	No	No	No	No
Pandian and Mahapatra [14]	Neural Networks	Yes	No	Yes	Yes
Mahdavi *et al*. [15]	Genetic Algorithm	Yes	No	Yes	No
Mahdavi and Shirazi [16]	Heuristic Algorithm	Yes	No	Yes	No
Arkat *et al*. [17]	Simulated Annealing	No	Yes	No	No
Ahi *et al*. [18]	TOPSIS	Yes	No	Yes	No
Wang *et al*. [19]	Scatter Search	Yes	Yes	No	No
Murugunandam *et al*. [20]	GA + Tabu Search	Yes	Yes	No	No

the effectiveness of GA method.

2. Problem Formulations

2.1 Notations

C, M, P: Total number of Cells, Machines, and Parts.

i,j,c: Index of machines, parts and cells respectively.

Dj: Demand for part j.

L_c: Minimum number of machines which should be assigned to cell c.

y_{jc}: Boolean decision variable, which is 1 if part j is assigned cell c, and 0 otherwise.

x_{ic}: Boolean decision variable, which is 1 if machine i is assigned to cell c, and 0 otherwise.

α_{ij}: Boolean parameter, which is 1 if part j needs machine i for completion, and 0 otherwise.

β_{ij}: Boolean parameter, which is 1if machine i is the first or the last machine needed for part j, and 0 otherwise.

2.2 Mathematical Formulation

The objective function of the proposed model is to minimize the total number of intercellular movements (f_1) and total number of voids (f_2) which can be formulated as below:

$$f_1 = \sum_{c=1}^{C}\sum_{i=1}^{M}\sum_{j=1}^{P} D_j.\alpha_{ij}.\,(2-\beta_{ij})\,.y_{jc}\,.(1-x_{ic}); \tag{1}$$

$$f_2 = \sum_{c=1}^{C}\sum_{i=1}^{M}\sum_{j=1}^{P} D_j.(x_{ic}.y_{jc} - \alpha_{ij}.\,x_{ic}.y_{jc}); \tag{2}$$

Minimize $f = f_1 + f_2$

Subject to:

$$y_{jc} \le \sum_{i=1}^{M} x_{ic} \ ; \qquad \forall\ j,c \tag{3}$$

$$\sum_{c=1}^{C} x_{ic} = 1; \qquad \forall\ i \tag{4}$$

$$\sum_{c=1}^{C} y_{jc} = 1; \qquad \forall\ j \tag{5}$$

$$\sum_{i=1}^{M} x_{ic} \ge L_c; \qquad \forall\ c \tag{6}$$

$$x_{ic},\ y_{jc} \in \{0,1\}; \tag{7}$$

Constraint (3) implies that assignment of a part to a cell is subject to the presence of at least one machine in that cell. Constraints (4) and (5) ensure that any part or machine is assigned to only one cell. Constraint (6) maintains the size of the cells and guarantees that at least a predefined minimum number of parts will be assigned to each cell.

3. The ACO Algorithm

Ant colony optimization was first developed by Dorigo *et al.* [21] based on the behavior of real ants. Real ants which live in colonies leave the nest to find food and come back again at every time. Based on observations, these ants always choose the shortest path to reach the food. As soon as this shortest path is found by some ants, the subsequent ants follow the same path. In fact there is a complicated communication system controlling the movement of ants. The secret of this communication is based on a substance, called *pheromone*. Real ants lay a substance known as pheromone on the ground when they pass through a path. This substance is smelled by other ants which leads them to follow the path traveled by prior ants. The more ants pass on a path, the more pheromone is put on that path. Since the shorter path is traveled fast, the density of pheromone on this path increases faster than other paths. Therefore, a majority of ants intend to travel on the shorter path after a given time. This is the underlying mechanism of ACO which is implemented to solve CFP in the following subsections.

3.1 Solution Representation and Evaluation

Suppose there are *M* machines and *P* parts to be clustered into *C* cells. The relation of machines and parts, which shows machine requirement of parts is normally represented by a matrix named. In this paper, the machine-part incidence matrix indicates the production process data as well. Specifically, an entry $a_{ij} = k$ in the matrix, means that operation *k* of part *j* needs machine *i* for completion. **Table 2** shows an example including fifteen machines and twenty-five parts.

In the proposed algorithm, a solution is represented with a string of length *M* + *P*. The first *M* characters of the string show the cell number of the machines, and the rest are used for the parts. For example, for *M* = 5, *P* = 4, and *C* = 2, a solution can be represented as "212112211" which means assign of machine 1 to 5 to cells 2,1,2,1 and 1 respectively and assignment of parts 1 to 4 to cells 2,2,1 and 1 respectively.

The generated solution may be infeasible, that is, the selected machine and part, are assigned to a wrong cell. In this case, the solution will be deleted, otherwise the goodness (or efficiency) of the solution will be computed.

3.2 Goodness Measurement

One of the most important steps in heuristic techniques is the evaluation of the obtained solutions. In this step, the *goodness* (or *fitness*) of the solution is calculated, and based on the result, the solution may be deleted, kept, or marked as good. The GA technique always keeps a population of feasible best fitted chromosomes (obtained solutions) and tries to achieve better solutions by mating the parents. Similarly, the ACO model keeps a list of best solutions ever found called *elite list*. When a new solution

Table 2. 15 × 25 machine-part matrix and demand for parts (*Dj*) of example 1

	Parts	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	16	18	19	20	21	22	23	24	25
	Dj	59	95	30	46	13	34	12	63	57	74	98	5	93	75	22	24	61	100	26	56	19	67	97	24	47
Machines	**M1**		1				1		1					1	3	3	1		1							
	M2	2			2	2	4	2			5				2			2		8			2			
	M3									6			6							6		1			6	5
	M4			2					5	3	2	2	2							2	1	2		2	2	2
	M5	3			3	3		3										3					3		7	
	M6		3	4			3		2					3		2	3		4							
	M7	1			1			1							1			1					1			
	M8							7		2	4	5	4							4	4	4		4	4	
	M9	4			4	4		4							4								4	5		
	M10		4			1	5		4							4	4		3							
	M11									5	6	4	5							5	5	5		6	5	4
	M12			1						1	1	1	1							1	2			1	1	1
	M13	5			5			5							5			4								
	M14	6		3						4	3	3	3							3	3	3		3	3	3
	M15		2				2	6	3					2		1	2		2	7						

is obtained, the goodness (fitness) function is applied, and based on the result, we decide to add the solution to the elite list, or omit the solution and generate another one.

In this model, considering the objective function defined in Sub-section 2.3, the number of intercell movements and the number of voids in cells is to be minimized. So, the goodness function is defined as below:

$$goodness = f = \frac{1}{f_1 + f_2 + 1} \qquad (8)$$

During the ACO, SA and GA iterations, the goodness of each solution is calculated using Equation (8). The constant value “1” is added to prevent division by zero.

3.3 The Ant Algorithm

Descriptive procedure of the proposed algorithm for solving the attempted mathematical model is as follows:

```
Begin
  1. Initialize
  2. Generate a feasible random solution, and add
     it to the elite list.
  3. Evaluate the efficiency (goodness) of the solution
  4. Repeat
       a. Generate another random solution,
          based on pheromone trails.
       b. Evaluate goodness, if better than the
          worst solution in the elite list, add it
          to elite list and delete the worst solution
          from list, update pheromone trails.
       c. Evaporate pheromone
       d. Alter solutions periodically
  5. Until stopping condition is met
 End
```

This model uses a $\mathbf{P} = [P_{ck}]_{(C) \times (M + P)}$ pheromone matrix in which, C, M, and P, are the number of cells, machines and parts, respectively. The initial value of P_{ck} is 1. So, to generate a random feasible solution in step 3, there is no need to use the pheromone matrix.

4. Genetic Algorithm

In this method, an initial population of solutions (chromosomes) is generated randomly while subsequent populations are generated by choosing good parents and mating them. The mating may cause to worse, better, or even infeasible solutions. By keeping better solutions in population and omitting bad ones, the algorithm converges stage by stage and after a number of iterations, the local or global optimum will be found.

4.1 The GA Algorithm

```
Begin
   1. Generate initial population containing N chromosomes.
   2. Compute the fitness of chromosomes in current population.
   3. Generate the next population
      a. Choose two best parents randomly from the current population.
      b. Mate the parents and generate two children (Crossover operator).
      c. Apply the mutation operator.
      d. Compute the fitness of children.
   4. Repeat step 3 until termination condition is met.
   End
```

In the proposed algorithm, the size of initial population is 1000 and the mating candidate parents are chosen by roulette wheel method. Chromosomes are represented as described in Subsection 3.1. The probability of crossing over and mutation is considered as 0.8 and 0.2 respectively.

5. Simulated Annealing

The Simulated Annealing (SA) algorithm is derived from metallurgy and thermodynamics which incorporated a temperature parameter into the minimization parameter. A high temperature expands the search space, and a lower temperature restricts the exploration. The procedure starts from a high temperature and ends at a low temperature. At each temperature, a number of iterations are done.

Some heuristic algorithms like Hill Climbing technique, may found the *Local Optimum* instead of the *Global optimum* because the movements leading to a new point worse than the current point are not allowed. SA algorithm allows non-improving movements to be taken in the hope of escaping the local optimum with a probability depending on the procedure temperature and the amount of the badness of the solution.

5.1 The SA Algorithm

Using the same representation described for solutions in Subsection 3.1, the SA algorithm can be written as follows:

```
Begin
  1. Initialize Temp.
  2. Find a feasible solution, called x.
  3. Compute its goodness f(x).
  4. Repeat until  frozen
       a. Do 1000 times
            i.   y: = FindNeighbour(x).
            ii.  Delta: = f(x) - f(y).
            iii. if Delta > 0 then x: = y ; Accept y).
            iv.  else if  U(0,1) < $e^{-(Delta)/Temp}$
                 then  x: = y  (Accept  y).
            v.   else reject  y.
       b. End Do.
  5. Temp: = Temp * 0.95.
  6. The Solution will be best so far.
End
```

The Algorithm starts at the temperature of 5000 with a feasible solution. The neighbor of a solution is obtained by making some changes in the solution (some parts or machines are randomly moved from one cell to other one).

6. Examples

The proposed algorithms are applied to solve two benchmark problems (15 × 25 and 20 × 35 sizes) available in the literature. The algorithms are implemented in C Language and are executed on a Pentium IV PC.

Example 1

The first example consists of fifteen machines and twenty-five parts which are grouped in three cells. The machine-part incidence matrix of the problem with is given in **Table 2**. The table also indicates demand of each part generated by a discrete uniform distribution in [1..100]. The solution obtained by the proposed ACO, SA and GA algorithms for this problem is shown in **Table 3**. The obtained values for f_1, f_2 and f are 863, 803 and 1666 respectively in all three methods.

Example 2

The second example is adopted from Boe and Cheng [22] with 35 parts, 20 machines and four cells. **Table 4** and **Table 5** show the machine-cell incidence matrix and the solution obtained by ACO, GA and SA algorithms. Since the demand for parts are not included in [22], these values are randomly generated using a discrete uniform distribution in [1..100]. The final obtained value of the objective function is 4562 in all three methods.

Figures 1 and **2** show the number of iterations and convergence speed of different algorithms. In these figures, the horizontal axis shows the number of iterations (number of generations in GA, number of ants in ACO, and iterations of SA) and the vertical axis shows the value of the objective function.

Table 3. The solution obtained by the proposed ACO, SA, and GA algorithms for example 1

Machines	1	4	5	7	14	17	22	2	6	8	13	15	16	18	3	9	10	11	12	19	20	21	23	24	25
M7	1	1		1	1	1	1																		
M2	2	2	2	2	2	2	2		4								5			8					
M5	3	3	3	3		3	3																	7	
M13	4	4	4	4	4		4																5		
M9	5	5		5	5	4																			
M1					3			1	1	1	1	3	1	1											
M15				6				2	2	3	2	1	2	2						7					
M6								3	3	2	3	2	3	4	4										
M10			1					4	5	4		4	4	3											
M12															1	1	1	1	1	1	2		1	1	1
M4										5					2	3	2	2	2	2	1	2	2	2	2
M14	6														3	4	3	3	3	3	3	3	3	3	3
M8				7												2	4	5	4	4	4	4	4	4	
M11																5	6	4	5	5	5	5	6	5	4
M3																6			6	6		1		6	5

Table 4. The machine-cell incidence matrix of example 2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
1	5					4					5			6	5			1		4			6	2	3					5		4		1	4
2		1								1		3	1					3	6					3							3				
3	1		1		2										1		1												1						
4		2					2					4	2											4			1								
5					1									1			1						1			1									
6								1						2					1			1								3					
7	2		2		3		1					1			2		2		5	1		3	4	1		4			2	4	2	3			
8	3				4				5						3		3			2			5												2
9								2						3					2				2									2			1
10								3						4		1			3			2				2									
11				1		1			1		1										1												1		
12				2		2			2		2										2												2		
13		3										5	3											5											
14		4					3			2		6	4					4						6			2				4				
15				3		3			3		3										3							1		1					
16				5		5			6			2						2								5				6		5			
17	4		3		5										4		4			3					2										3
18		5					4					7	5							5				7							5				
19				4					4		4										4							2		2		1			
20								4						5					4				3			3									

Table 5. The solution obtained by the proposed ACO, SA, and GA algorithms for example 2

	1	3	5	15	17	18	20	23	25	29	32	34	35	2	7	10	12	13	24	27	31	8	14	16	19	22	26	4	6	9	11	21	28	30	33
1	5			5		1	4	6	3		4	1	4						2				6						4		5			5	
3	1	1	2	1	1					1																									
7	2	2	3	2	2		1	4		2	3				1		1		1		2				5	3	4							4	
8	3		4	3	3		2	5					2																	5					
17	4	3	5	4	4		3		2				3																						
2						3								1		1	3	1	3		3				6										
4														2	2		4	2	4	1															
13														3			5	3	5																
14						4								4	3	2	6	4	6	2	4														
18							5							5	4		7	5	7		5														
5			1		1			1															1				1								
6																						1	2		1	1								3	
9								2			2		1									2	3		2										
10																						3	4	1	3	2	2								
20								3														4	5		4		3								
11																												1	1	1	1	1			1
12																												2	2	2	2	2			2
15																												3	3	3	3	3	1	1	
16						2					5						2										5	5	5	6				6	
19											1																	4		4	4	4	2	2	

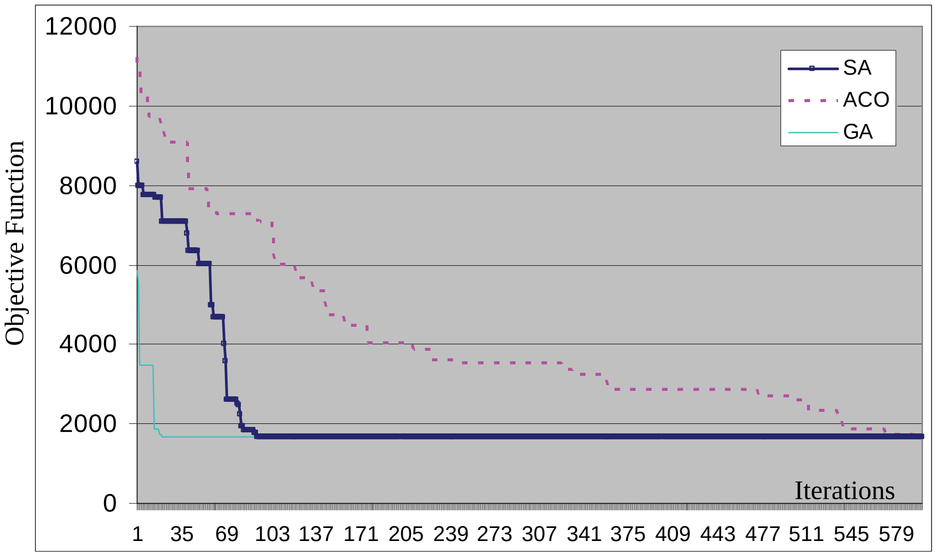


Figure 1. Convergence speed of GA, ACO and SA for example

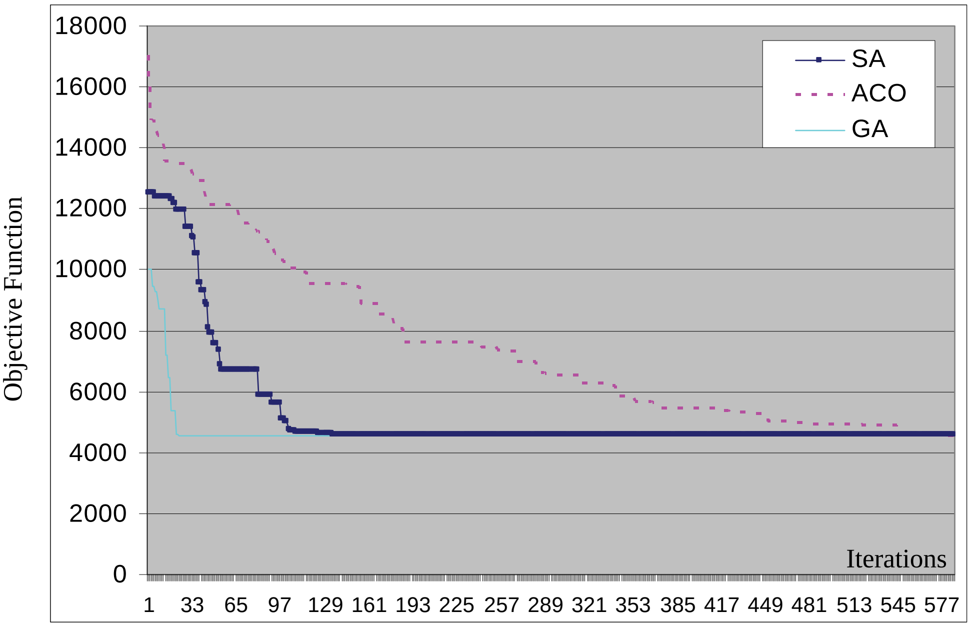


Figure 2. Convergence speed of GA, ACO and SA for example 2

Table 6. Computational time of GA, ACO and SA

Meta-Heuristic Method	Example 1	Example 2
GA	**10**	**14**
SA	**37**	**48**
ACO	**251**	**263**

The computational time (in seconds) of different algorithms for examples 1 and 2 are shown in **Table 6**.

7. Conclusions

This paper discusses that the sequence of operations and the production volume are two major factors to be considered in the design of CMS. Despite this fact, it has not been taken into account in a majority of researches available in the literature. To capture this fact, a new model for solving cell formation problem in CMS is proposed. Due to the NP-hardness of the formulated problem, three solution approaches based on ACO, GA and SA are used to solve the model. The objective function of the model is to minimize the total number of intercellular movements and the number of voids. The total number of cells is defined as a constant parameter in the algorithm.

The computational results show that the proposed algorithms are effective in minimizing the total number of voids and intercellular movements.

As shown in **Figures 1** and **2**, the GA algorithm has obtained the optimum value faster than other techniques.

The attempted mathematical model can be further extended by considering alternate process plans for each part, machine redundancy, processing time of each operation, etc.

REFERENCES

[1] C. Zhao and Z. Wu, "A Genetic Algorithm for Cell Formation with Multiple Routes and Multiple Objectives," *International Journal Production Research*, Vol. 38, No. 2, 2000, pp. 385-395.

[2] C. C. Gallagher and W. A. Knight, "Group Technology Production Methods in Manufacturing," Knight/Ellis Horwood Limited, 1986.

[3] J. S. Morris and R. J. Tersine, "A Simulation Analysis of Factors Influencing the Attractiveness of Group Technol-

ogy and Cellular Layouts," *Management Science*, Vol. 36, No. 12, 1990, pp. 1567-1578.

[4] J. L. Burbidge, "Group-Technology in Engineering Industry," Mechanical Engineering Publication Ltd., UK, 1979.

[5] A. Ballakur and H. J. Steudel, "A within Cell Utilization Based Heuristic for Designing Cellular Manufacturing Systems," *International Journal of Production Research*, Vol. 25, No. 5, 1987, pp. 639-655.

[6] J. A. Joines, R. E. King and C. T. Culbreth, "A Comprehensive Review of Production-Oriented Manufacturing Cell Formation Technique," *International Journal of Flexible Automation and Integrated Manufacturing*, Vol. 3, No. 3-4, 1996, pp. 225-265.

[7] A. Islier, "Group Technology by Ant System Algorithm," *International Journal of Production Research*, Vol. 43, No. 5, 2005, pp. 913-932.

[8] G. Prabhaharan, A. Murugunandam and P. Asokan, "Machine Cell Formation for Cellular Manufacturing Systems Using an Ant Colony System Approach," *International Journal of Advanced Manufacturing Technology*, Vol. 25, 2005, pp. 1013-1019.

[9] K. L. Mak, P. Peng, X. X. Wang and T. L. Lau, " An Ant Colony Optimization Algorithm for Scheduling Virtual Manufacturing Systems," *International Journal of Computer Integrated Manufacturing*, Vol. 20, No. 6, 2007, pp. 524-537.

[10] K. Spiliopoulos and S. Sofianpoulou, "An Efficient Ant Colony Optimization System for the Manufacturing Cells Formation Problem," *International Journal of Advanced Manufacturing Technology*, Vol. 36, No. 5-6, 2008, pp. 589-597.

[11] S. E. Kesen, M. D. Toksari, Z. Gungor and E. Guner, "Analyzing the Behaviors of Virtual Cells (Vcs) and Traditional Manufacturing Systems: Ant Colony Optimization (ACO)-Based Metamodels," *Computers and Operations Research*, Vol. 36, No. 7, 2009, pp. 2275-2285.

[12] S. I. Satoglu and N. C. Surech, "A Goal Programming Approach for Design of Hybrid Cellular Manufacturing System in Dual Resource Constrained Environments," *Computers and Industrial Engineering*, Vol. 56, No. 2, 2009, pp. 560-575.

[13] Y. Kao and S. C. Fu, "An Ant-Based Clustering Algorithm for Manufacturing Cell Design," *International Journal of Advanced Manufacturing Technology*, Vol. 28, 2006, pp. 1182-1189.

[14] R. S. Pandian and S. S. Mahapatra, "Manufacturing Cell Formation with Production Data Using Neural Networks," *Computers and Industrial Engineering*, Vol. 56, No. 4, May 2009, pp. 1340-1347.

[15] I. Mahdavi, M. M. Paydar, M. Solimanpur and A. Heidarzadeh, "Genetic Algorithm Approach for Solving a Cell Formation Problem in Cellular Manufacturing," *Expert Systems with Applications*, Vol. 36, No. 3, 2009, pp. 6598-6604.

[16] I. Mahdavi, B. Shirazi and M. M. Paydar, "A Flow Matrix-Based Heuristic Algorithm for Cell Formation and Layout Design in Cellular Manufacturing System," *International Journal of Advanced Manufacturing Technology*, Vol. 39, No. 9-10, 2008, pp. 943-953.

[17] J. Arkat, M. Saidi and B. Abbasi, "Applying Simulated Annealing to Cellular Manufacturing System Design," *International Journal of Advanced Manufacturing Technology*, Vol. 32, No. 5-6, 2007, pp. 531-536.

[18] A. Ahi, M. B. Aryanezhad, B. Ashtiani and A. Makui, "A Novel Approach to Determine Cell Formation, Intercellular Machine Layout and Cell Layout in the CMS Problem Based on TOPSIS Method," *Computers and Operations Research*, Vol. 36, 2009, pp. 1478-1496.

[19] X. Wang, J. Tang and K. Yung, "Optimization of the Multi Objective Dynamic Cell Formation Problem Using a Scatter Search Approach," *International Journal of Advanced Manufacturing Technology*, Vol. 44, No. 3-4, 2009, pp. 318-329.

[20] A. Muruganandam, G. Prabhaharan, P. Asokan and V. Baskaran, "A Memetic Algorithm Approach to the Cell Formation Problem," *International Journal of Advanced Manufacturing Technology*, Vol. 25, No. 9-10, 2005, pp. 988-997.

[21] M. Dorigo and G. D. Caro, "The Ant Colony Optimization Meta-Heuristic," McGraw-Hill, New York, 1999.

[22] W. J. Boe and C. Cheng, "A Close Neighbour Algorithm for Designing Cellular Manufacturing Systems," *International Journal of Production Research*, Vol. 29, No. 10, 1991, pp. 2097-2116.

J. Software Engineering & Applications, **2010, 3, 683-695**
doi:10.4236/jsea.2010.37078 Published Online July 2010 (http://www.SciRP.org/journal/jsea)

Melody Generator: A Device for Algorithmic Music Construction

Dirk-Jan Povel

Centre for Cognition, Donders Institute for Brain, Cognition and Behaviour, Radboud University Nijmegen, Nijmegen, The Netherlands.
Email: povel@NICI.ru.nl

Received April 15th, 2010; revised May 25th, 2010; accepted May 27th, 2010.

ABSTRACT

This article describes the development of an application for generating tonal melodies. The goal of the project is to ascertain our current understanding of tonal music by means of algorithmic music generation. The method followed consists of four stages: 1) *selection of music-theoretical insights,* 2) *translation of these insights into a set of principles,* 3) *conversion of the principles into a computational model having the form of an algorithm for music generation,* 4) *testing the "music" generated by the algorithm to evaluate the adequacy of the model. As an example, the method is implemented in* Melody Generator, *an algorithm for generating tonal melodies. The program has a structure suited for generating, displaying, playing and storing melodies, functions which are all accessible via a dedicated interface. The actual generation of melodies, is based in part on constraints imposed by the tonal context, i.e. by meter and key, the settings of which are controlled by means of parameters on the interface. For another part, it is based upon a set of construction principles including the notion of a hierarchical organization, and the idea that melodies consist of a skeleton that may be elaborated in various ways. After these aspects were implemented as specific sub-algorithms, the device produces simple but well-structured tonal melodies.*

Keywords: *Principles of Tonal Music Construction, Algorithmic Composition, Synthetic Musicology, Computational Model, Realbasic, OOP, Melody, Meter, Key*

1. Introduction

Research on music has yielded a huge amount of concepts, notions, insights, and theories regarding the structure, organization and functioning of music. But only since the beginning of the 20th century has music become a topic of rigorous scientific research in the sense that aspects of the theory were formally described and subjected to experimental research [1]. Starting with the cognitive revolution of the 1960s [2] a significant increase in the scientific study of music was seen with the emergence of the disciplines of cognitive psychology and artificial intelligence. This gave rise to numerous experimental studies investigating aspects of music perception and music production, for overviews see [3,4], and to the development of formal and computational models describing and simulating various aspects of the process of music perception, e.g., meter and key induction, harmony induction, segmentation, coding, and music representation [5-13].

Remembering Richard Feynman's adage "What I can't create, I don't understand", this article is based on the belief that the best estimate of our understanding of music will be obtained from attempts to actually create music. For that purpose we need a computer algorithm that generates music.

1.1 Algorithmic Music Construction

The rationale behind the method is simple and straightforward: if we have a theory about the mechanism underlying some phenomenon, the best way to establish the validity of that theory is to show that we can reproduce the phenomenon from scratch. Applied to music: if we have a valid theory of the structure of music, then we should be able to construct music from its basic elements (sounds differing in frequency and duration), at least in some elementary fashion. The core of the method therefore consists in the generation of music on the basis of insights accumulated in theoretical and experimental music research.

In essence, the method includes four stages 1) specification of the theoretical basis; 2) translation of the theory into a set of principles; 3) implementation of the principles as a generative computer algorithm; 4) test of the output. This article only describes the three former stages:

the theoretical background and the actual development of the algorithmic music construction device. The testing of the device which calls for a separate dedicated study will be reported in a separate article.

Compared to the experimental method, the algorithmic construction method has two advantages: first, it studies all structural aspects of music in a comprehensive way (because all aspects have to be taken into account if one wants to generate music) thus exhibiting the "big picture": the working and interaction of all variables; second, it requires utmost precision, enforced by the fact that the model is implemented as a computer algorithm. Experimental research in music, because of inherent limitations of the method, typically focuses on a restricted domain within the field of music, taking into account just one or two variables. As a result of this, the interpretation of the experimental results and their significance for the understanding of music as a whole, is often open to discussion.

1.2 Related Work

The possibility to create music by means of computer algorithms, so-called algorithmic composition, has attracted a lot of interest in the last few decades [14-16]. Algorithmic composition employs various methods [17]: Markov chains [18,19]; Knowledge-based systems [20]; Grammars [14,21-24]; Evolutionary methods divided into genetic algorithms, [25] and interactive genetic algorithms [26]; and learning systems [27,28].

Because the method presented here generates music it could be seen as an instance of algorithmic composition. But, since its purpose is to serve as a means to test music theory, it does not merely rely on classical AI methods but capitalizes on music theoretical insights to guide the implementation of the music generating device. As such it resembles the approaches taken by [9,29-31].

1.3 Organization of the Paper

The present article is organized as follows. In Section 2 the theoretical foundation of the method is described: the theoretical starting point, the basic assumptions, the insights regarding the configuration of the time dimension and the ensuing constraints, the configuration of the pitch dimension and ensuing constraints, and the basic construction rules for generating melodies. In Section 3, the implementation of the algorithm is discussed: the various functions of the program, its underlying structure, its user interface, and the implementation of the tonal context and the construction principles.

2. Theoretical Foundation

Before describing the theoretical basis of the project, I should point out that what is being presented here is a particular set of theoretical notions based on a specific interpretation of the findings in the literature. Thus, I do not claim that this is the only possible theoretical basis, and certainly not that it is complete. The main purpose of this paper is to study the adequacy of a melody generating device based on specific theoretical ideas about the construction of tonal music.

Starting point of the project is the conception of music as a psychological phenomenon, *i.e.*, as the result of a unique perceptual process carried out on sequences of pitches. In the case of tonal music this process has two distinct aspects: 1) Discovering the context in which the music was conceived (meter, key, and harmony) and representing the input within that context. By this representation the input, consisting of sounds varying in pitch, is transformed into tones and chords having specific musical meanings; 2) Discovering the structural regularities in the input (e.g., repetition, alternation, reversal) and using these to form a mental representation [5,7]. These processes evolve largely unconsciously: What the listener experiences are the effects of these processes, specifically the experiences that accompany the expectations arising while the music develops and the ways in which these expectations are subsequently resolved. Sometimes these experiences are described by the terms tension and relaxation, but these hardly seem to cover the subtle and varied ways humans may respond to music.

This basic conception of the process of music perception has guided the choice of assumptions, and the development of the models of music construction described below, as well as the shaping of the interface of the computer algorithm.

2.1 Basic Assumptions

The model is based on the following assumptions: 1) Tonal music is conceived within the context of time and pitch, where time is configured by meter (imposing constraints as to *when* notes may occur), and pitch by key and harmony (imposing constraints as to *what* notes may occur). 2) Within that context tone sequences are generated using construction rules specifying the (hierarchical) organization of tones into parts, and of parts into larger parts etc., relating to concepts such as motives, phrases, repetition and variation, skeleton, structural and ornamental tones, etc.

In line with these assumptions two components may be discerned in the program: one component that manages the context, and another that handles the construction rules.

Below, we discuss the configuration of the time dimension, the configuration of the pitch dimension, the interaction between these dimensions, the basic principles of music construction, and the resulting constraints and rules.

2.2 The Time Dimension of Tonal Music

The time dimension in tonal music is configured by meter. Meter is a temporal framework in which a rhythm is cast. It divides the continuum of time into discrete periods. Note onsets in a piece of tonal music coincide with the beginning of one of those periods.

Meter divides time in a hierarchical and recurrent way: time is divided into time periods called measures that are repeated in a cyclical fashion. A measure in turn is subdivided in a number of smaller periods of equal length, called beats, which are further hierarchically subdivided (mostly into 2 or 3 equal parts) into smaller and smaller time periods. A meter is specified by means of a fraction, e.g., 4/4, 3/4, 6/8, in which the numerator indicates the number of beats per measure and the divisor the duration of the beat in terms of note length (1/4, 1/8 etc.), which is a relative duration.

The various positions defined by a meter (the beginnings of the periods) differ in metrical weight: the higher the position in the hierarchy (*i.e.* the longer the delimited period), the larger the metrical weight. **Figure 1** displays two common meters in a tree-like representation indicating the metrical weight of the different positions (the weights are defined on an ordinal scale). As shown, the first beat in a measure (called the down beat) has the highest weight. Phenomenally, metrical weight is associated with perceptual markedness or accentuation: the higher the weight, the more accented it will be.

Although tonal music is commonly notated within a meter, it is important to realize that meter is not a physical characteristic, but a perceptual attribute conceived of by a listener while processing a piece of tonal music. It is a mental temporal framework that allows the accurate representation of the temporal structure of a rhythm (a sequence of sound events with some temporal structure). Meter is obtained from a rhythm in a process called metrical induction [32,33]. The main factor in the induction of meter is the distribution of accents in the rhythm: the more this distribution conforms to the pattern of metrical weights of some meter, the stronger that meter will be induced. The degree of accentuation of a note in a rhythm is determined by its position in the temporal structure and by its relative intensity, pitch height, duration, and spectral composition, the temporal structure being the most powerful [34]. The most important determiners of accentuation are 1) Tone length as measured by the inter-onset-interval (IOI); 2) Grouping: the last tone of a group of 2 tones and the first and last tone of groups of three or more tones are perceived as accentuated [33,35].

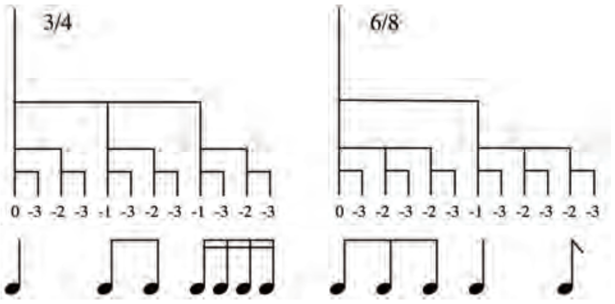


Figure 1. Tree-representations of one measure of 3/4 and 6/8 meter with the metrical weights of the different levels. For each meter a typical rhythm is shown

2.2.1 Metrical Stability

The term metrical stability denotes how strongly a rhythm evokes a meter. We have seen that the degree of stability is a function of how well the pattern of accents in a rhythm matches the pattern of weights of a meter. This relation may be quantified by means of the coefficient of correlation. If the metrical stability of a rhythm (for some meter) falls below a critical level, by the occurrence of what could be called “anti-metric” accents, the meter will no longer be induced, leading to a loss of the temporal framework and, as a consequence, of the understanding of the temporal structure.

2.2.2 Basic Constraints Regarding the Generation of Tonal Rhythm

Meter imposes a number of constraints on the use of the time dimension when generating tonal music: 1) it determines the moments in time, the locations, at which a note may begin; 2) it requires that the notes are positioned such that the metrical stability is high enough to induce the intended meter.

2.3 The Pitch Dimension of Tonal Music

The pitch dimension of tonal music is organized on three levels: key, harmony, and tones. These constituents maintain intricate mutual relationships represented in what is called the tonal system. This system describes how the different constituents relate and how they function: e.g., how close one key is to another, which are the harmonic and pitch elements of a key, how they are related, how they function musically, etcetera. For a review of the main empirical facts see [4]. It should be noted that these relations between the elements of the tonal system do not exist in the physical world of sounds (although they are directly associated with it), but refer to knowledge in the listener’s long-term memory acquired by listening to music.

Key is the highest level of the system and changes at the slowest rate: in shorter tonal pieces like hymns, folksongs, and popular songs there is often only one key which does not change during the whole piece.

A key is comprised of a set of 7 tones, the diatonic scale. The tones in a scale differ in stability and the degree in which they attract or are attracted [36]. For instance, the first tone of the scale, the tonic, is the most stable and attracts the other tones of the scale, directly or indirectly, to different degrees. The last tone of the scale, the leading tone, is the least stable and is strongly attracted by the tonic. Phenomenally, attractions are experienced by a listener as expectations.

A key also contains a set of harmonies, basically instantiated by triads built on the 7 degrees of the scale. The triads on the 1st, 4th, and 5th degree are the primary ones. Like the tones, the harmonies, depending on their stability, may either attract or be attracted by other harmonies [37-39]. We start from the assumption that tonal melodies are built upon an underlying harmonic progression [40].

These musical functions only become available after a listener has recognized the key of the piece in a process analogous to finding the meter, called key induction. For an overview see [13].

2.3.1 Basic Constraints Associated with Key and Harmony

Once a selection for a particular key has been made, the tones of the diatonic scale are the primary candidates for the construction of a melody. The non-key or chromatic tones play a secondary role and can only be used under strict conditions and mainly as local ornaments. The selection of a specific harmonic progression reduces the selection of tones even more since the tones within some harmony must, in principle, be in accordance with that harmony: only the tones that are part of the harmony can be sounded together with it. These tones are called harmonic tones or chord tones. There are melodies only consisting of chord tones (e.g., Mozart, Eine kleine Nachtmusik), but most melodies also contain non-chord tones (e.g., suspensions, appoggiatura's, passing tones, neighbor tones etc.) (**Figure 2**).

As non-chord tones form dissonances with the harmony their use is restricted: they must be "anchored" by (resolved to) a succeeding chord tone close in pitch [41, 42]. Most often a non-chord tone is anchored immediately, *i.e.*, by the next succeeding tone, but it can also be anchored with some delay, as in the case of F A G, in which G is a chord tone and F and A are non-chord tones.

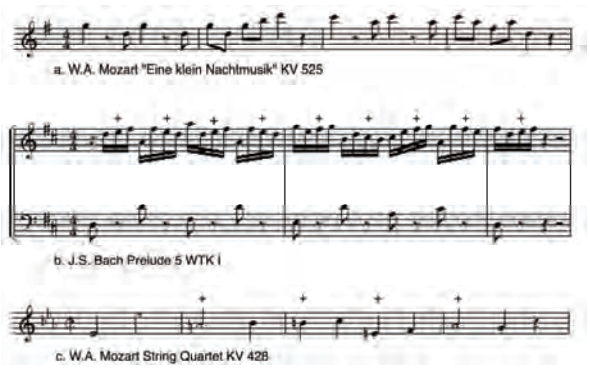

a. W.A. Mozart "Eine klein Nachtmusik" KV 525
b. J.S. Bach Prelude 5 WTK I
c. W.A. Mozart String Quartet KV 428

Figure 2. Three melodic fragments with different distributions of chord tones and non-chord tones (indicated +). (a) only chord tones; (b) non-chord tones on weak metrical positions; (c) non-chord tones on strong metrical positions

2.4 Relations between Meter and Key

Above it was mentioned that the notes within a key differ in stability. In general, the more stable a note, the more structurally important it will be in the melody. From this it follows that stable notes, in order to enforce their prominent role, tend to appear at metrically strong positions.

This, however, is not always the case: sometimes a non-chord tone is placed on a beat (a strong position) and resolved by a chord tone on a weaker metrical position (appoggiatura). See note example c in **Figure 2.** These different uses of non-chord tones is style dependent, see [43].

2.5 Principles of Melody Construction

So far I have described the major characteristics of the context within which tonal music is constructed: Meter which defines the positions at which notes may begin and the perceptual salience of these positions. Key defining the basic units of a tonal piece: tones and chords, their relation and their musical function. And lastly the harmonic progression underlying a melody that defines the chord tones and non-chord tones. Now the question must be answered how a melody is generated within this context.

We start with the following basic principles: 1) A melody consists of parts that may consist of subparts, which in turn may contain subparts, etc., thus forming a hierarchical organization; 2) Parts are often created from other parts by means of variation; 3) A part consists of a skeleton of structural tones, which may be elaborated (ornamented) to different extents.

The hierarchical organization of tonal music has been described by many authors. Bamberger [44] describes the hierarchical organization of tunes in terms of trees consisting of motives, phrases, and sections connected by means of 3 organizing principles: repetition, sequential relations, and antecedent-consequent relations. Lerdahl & Jackendoff [7] conceive of the organization of tonal pieces in terms of hierarchical trees, resulting from a time-span reduction (based on the rhythmical structure) and a prolongational reduction expressing harmonic and melodic patterns of tension and relaxation. Schoenberg [45] describes the hierarchical organization of classical music on several levels: large forms (e.g., sonata form), small forms (e.g., minuet), themes (e.g., the period and the sentence consisting of phrases and motives).

The idea that the surface structure of a melody can be reduced by a stepwise removal of less important tones thereby revealing the underlying framework or skeleton has a long history in music theory. Schenker [46] laid the theoretical foundation for the notion which was formalized in [7]. Similar ideas are found in [9,29,31,43,47]. Baroni *et al*. [30] proposed a set of transformations by which a melodic skeleton can be elaborated into a full-fledged melody (see Subsection 2.5.2). Marsden [31] proposed a framework for representing melodic patterns yielding a set of elaborations to produce notes between two parent notes.

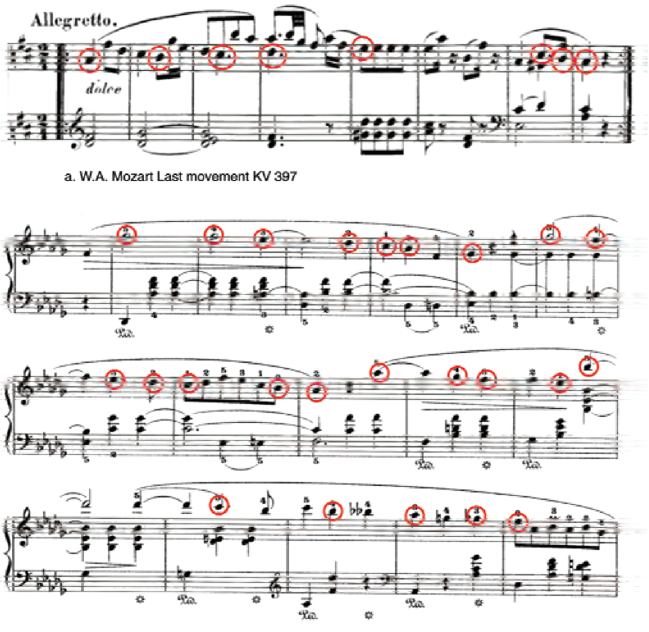
In view of the foregoing, we must first decide how to form a skeleton and next how to elaborate it. In line with the two major units of tonal music, the (broken) chord and the scale, and based upon analyses of large samples of tonal melodies, two models emerged: the “Chord-based model” and the “Scale-based model”. The Chord-based model builds a melodic skeleton from chord tones, whereas the skeleton in the Scale-based model consist of a scale fragment. The application described below contains a third model, called Basic, which is useful to compare tone sequences that are created either within or without a tonal context. The latter model will not further be discussed here.

2.5.1 The Chord-Based Model

This model is based on the following assumptions: 1) A melody is superimposed upon an underlying harmonic progression; 2) A melody consists of structural tones (a skeleton) and ornamental tones; 3) The skeleton is assembled from chord tones; 4) Skeleton tones are primarily located at metrically strong positions. To actually generate a skeleton melody within this model, apart from the parameters related to the rhythm and the harmonic progression, a few additional parameters are needed to further shape the skeleton, namely: *Range* (determining the highest and lowest note), *Contour* (the up-down movement of the melody), *Location* (on beats or down-beats), and *Interval* (the size of the steps between the successive notes of the skeleton). After generation of the skeleton it may be elaborated or ornamented to different degrees by means of an algorithm that interpolates notes between the notes of the skeleton. More information will follow in Section 3.

2.5.2 The Scale-Based Model

The Scale-based model is largely based on the same assumptions as the Chord-based model, with one essential difference: the skeleton is not formed of a series of chord tones, but consists of a scale fragment. The idea that the skeleton of tonal melodies consist of scale fragments has been proposed, among others, by [30,48]. The latter authors studied melodies from a period of almost 10 centuries and concluded that “...every melodic phrase can be reduced, at its deep level, to a kernel which not only progresses by conjunct step, but which is also monodirectional”. This rule applies to the “body” of the phrase, not to a possible anacrusis, or “feminine ending”. The variability on the surface is seen as the result of the application of two types of transformations to the kernel: linear transformations (repetition, neighbor, (filled) skip), and a harmonic transformation. The latter transformation, called “chord transposition”, substitutes a note for an other note of the underlying harmony. The relative frequency of harmonic transformations increases over time (Händel, Mozart, Liszt). **Figure 3** presents the reverse



a. W.A. Mozart Last movement KV 397

b. F. Chopin 3rd part of Waltz op. 64 No 2

Figure 3. The reduction of two melodies to their kernel. The kernel of melody a consists of an ascending scale fragment followed by a descending fragment. In the descending fragment the D is not realized in the surface. The highest level of melody b. consists of the scale fragment F Gb Ab (bars 1, 6, 9 respectively), of which each tone is augmented with a descending scale fragment. In bar 4 the Ab is replaced by F and in bar 11 there is an downward octave jump

process in which melodies are reduced to their kernel.

After a skeleton has been generated (either on the beats or the down-beats), it may be elaborated by one of the mentioned transformations or a combination of these: repetition, neighbor note, skip, or chord transposition. Implementation is detailed in the next section.

3. Implementation: Melody Generator

The next step consisted in the implementation in an algorithmic form of the notions and models for melody generation detailed above. This has led to the program Melody Generator that, as its name suggests, generates melodies. We chose for melodies rather than complete pieces mainly to keep the problem manageable. Moreover, melodies form a core aspect of tonal music, and many aspects of music construction are involved in their generation.

The program has a structure suited to generate, modify, play, display, store and save tonal melodies, and an interface suited to control these various aspects of the process. Consequently, Melody Generator evolved into an aid to gain an in-depth, hands-on understanding of tonal melody and its construction, enabling the user to study closely the consequences of changing the various parameters while building a melody. Melody Generator is programmed in REALbasic^{TM} [49], an object-oriented language that builds applications for Macintosh, Windows, and Linux platforms from a single source code.

The program can be downloaded free of charge at http://www.socsci.ru.nl/~povel/Melody/.

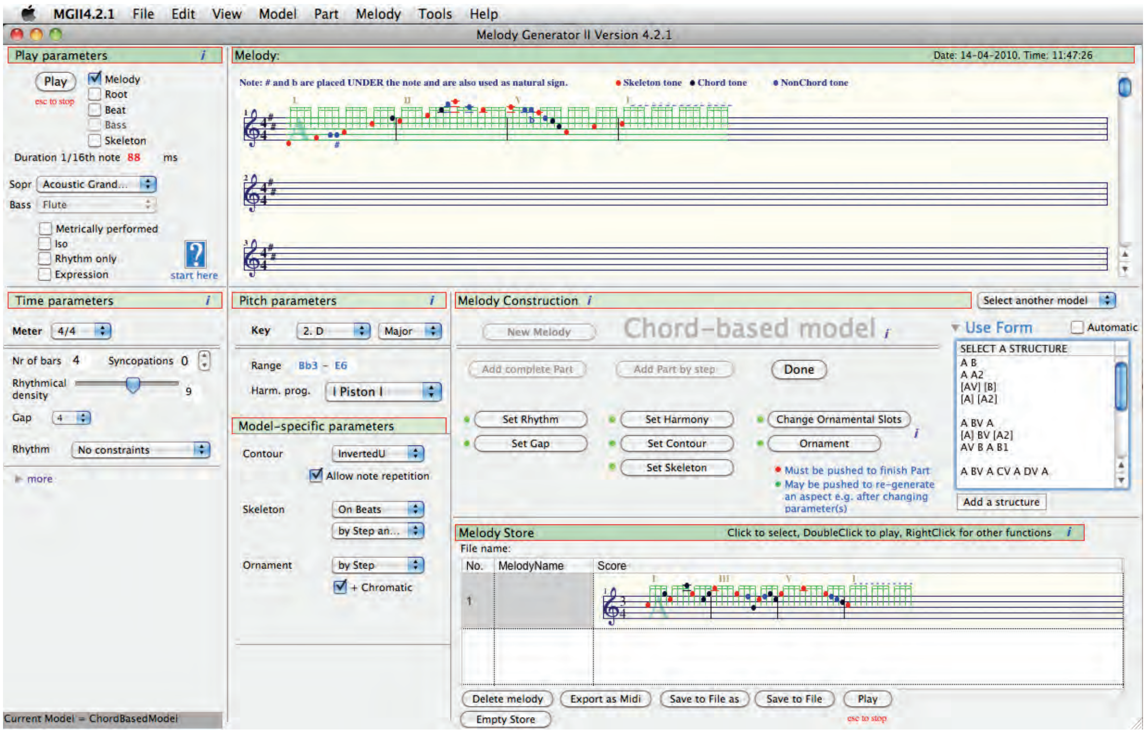
In this context only a concise description is presented. A proper understanding of the application can only be obtained from a hands-on experience. More detailed information about the implementation, the functioning and use of the parameters can be obtained by clicking the Help menu on the interface or the various info buttons on the interface of the program. A user guide can be found at http://www.socsci.ru.nl/~povel/Melody/InfoFilesMGII 2008/User Guide.html. The interface of Melody Generator, shown on **Figure 4**, comprises a number of panes serving its three main functions: Generation, Display/ Play, and Storage. These three functions are described in some detail below.

Following the distinction introduced above, I will successively describe the implementation of the tonal context and that of the construction principles.

3.1 Context

3.1.1 The Time Dimension

The time aspect is controlled by means of the following parameters: *Duration* (No. of bars), *Meter* (4/4, 3/4, 2/4, 6/8), *Gap* (between Parts), *Density* (relative number of notes per bar), *Syncopation*, and *Rhythmical constraint*. The latter parameter determines the degree of rhythmical similarity among bars. A description of the use of these parameters can be obtained by pressing the info button on



Note: # and b are placed UNDER the note and are also used as natural sign.

☑ Melody
□ Root
□ Beat
□ Bass
□ Skeleton

Duration 1/16th note 88 ms

Sopr Acoustic Grand...
Bass Flute

□ Metrically performed
□ Iso
□ Rhythm only
□ Expression

Meter 4/4
Nr of bars 4 Syncopations 0
Rhythmical density 9
Gap 4
Rhythm No constraints

Key 2. D Major
Range Bb3 – E6
Harm. prog. I Piston I

Contour InvertedU
☑ Allow note repetition
Skeleton On Beats
by Step an...
Ornament by Step
☑ + Chromatic

● Must be pushed to finish Part
● May be pushed to re-generate an aspect e.g. after changing parameter(s)

SELECT A STRUCTURE
A B
A A2
[AV] [B]
[A] [A2]

A BV A
[A] BV [A2]
AV B A B1

A BV A CV A DV A

Click to select, DoubleClick to play, RightClick for other functions
File name:

No.	MelodyName	Score
1		

Current Model = ChordBasedModel

Figure 4. Interface of melody generator version 4.2.1, showing a melody generated using the Chord-based model

the pane "Time Parameters" of the program. A detailed description of the algorithm to generate rhythms, varying in density and metrical stability, is also presented there.

3.1.2 The Pitch Dimension: Key and Harmony

Melody Generator has parameters for selecting a key (and mode) and for selecting a harmonic progression. Various template progressions are provided, partly based upon Schenker's notion that tonal music is constructed by applying transformations on a "Background" consisting of a I-V-I harmonic progression (Ursatz), [50-52]. Basic harmonic templates typically begin with I and end with V-I (full cadence) or V (half cadence). The intermediate harmonies are derived from Piston's "Table of usual root progressions" [53].

This table, shown here as **Table 1**, lists the transition probabilities between harmonies in tonal music. These probabilities are loosely defined by the terms "usually", "sometimes" and "less often". Since the choice of a chord is only determined by the immediately preceding chord, it only provides a first-order approximation of harmonic progressions in tonal music.

Also needed is a structure representing the various elements of the selected key and their mutual relations (mentioned above in Subsection 2.3). This includes information concerning the specification of the scale degrees, their stability and their spelling, the various chords on these degrees, etc. This information is accumulated in the object KeySpace. **Figure 5** displays one octave of the KeySpace of C-major.

3.2 Construction

To effectively construct melodies we need a structure allowing the flexible generation and modification of a hierarchically organized temporal sequence consisting of parts and subparts, divided into bars, in turn divided into beats and "slots" (locations where notes may be put). For this purpose an object called *Melody*, was designed comprising one or more instances of the objects *Piece*, *Part*, *Bar*, *Beat*, *Slot*, and *Note*. See **Figure 6**. Since any part may contain a subpart, the hierarchy of a piece can be extended indefinitely.

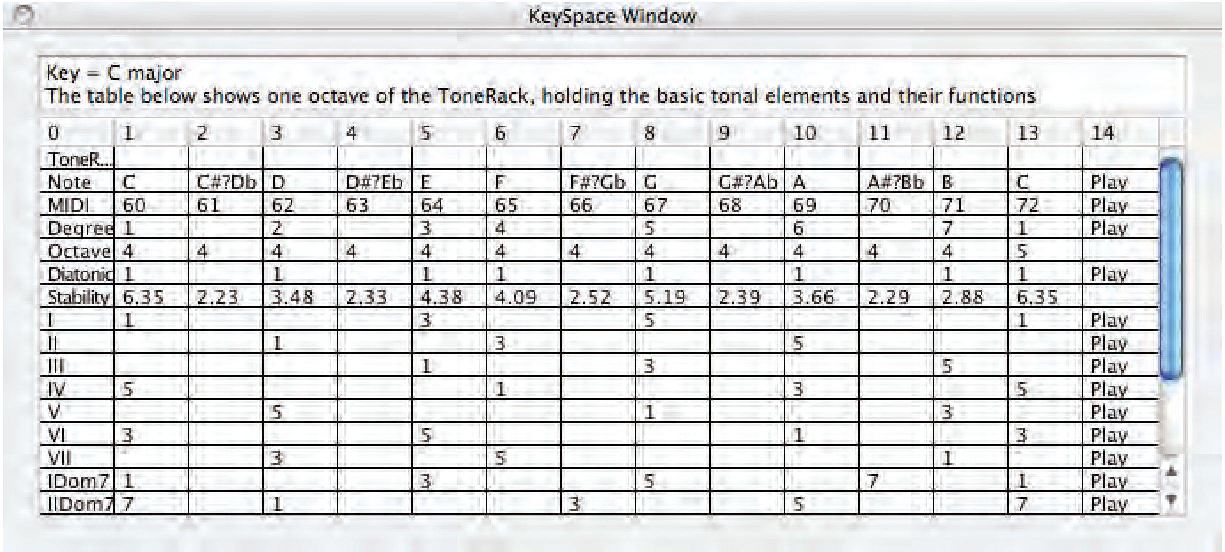
The generation of a melody progresses according to the following stages: Construction, Editing, (Re)arrangement, and Transformation.

3.2.1 Part Construction

Construction is performed in the "Melody Construction" pane: after setting the Meter and Key parameters and pushing the "New Melody" button, the first Part of a melody may be generated either stepwise or at once. In the stepwise mode, the various aspects of a melody: its *Rhythm*, *Gap*, *Harmony*, *Contour*, *Skeleton*, and *Ornamentation* are generated in separate steps. Prior to each

Table 1. The table of usual root progressions for the Major mode from Piston & DeVoto. Given a chord in the first column, the table shows the chords that succeed that chord, usually (1st column), sometimes (2nd column), or less often (3rd column)

Chord	Usually followed by	Sometimes by	Less often by
I	IV or V	VI	II or III
II	V	IV or VI	I or III
III	VI	IV	I, II or V
IV	V	I or II	III or IV
V	I	VI or IV	III or II
VI	II or V	III or IV	I
VII	III	I	-



Key = C major
The table below shows one octave of the ToneRack, holding the basic tonal elements and their functions

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
ToneR...														
Note	C	C#?Db	D	D#?Eb	E	F	F#?Gb	G	G#?Ab	A	A#?Bb	B	C	Play
MIDI	60	61	62	63	64	65	66	67	68	69	70	71	72	Play
Degree	1		2		3	4		5		6		7	1	Play
Octave	4	4	4	4	4	4	4	4	4	4	4	4	5	
Diatonic	1		1		1	1		1		1		1	1	Play
Stability	6.35	2.23	3.48	2.33	4.38	4.09	2.52	5.19	2.39	3.66	2.29	2.88	6.35	
I	1				3			5					1	Play
II			1			3				5				Play
III					1			3				5		Play
IV	5					1				3			5	Play
V			5					1				3		Play
VI	3				5					1			3	Play
VII			3			5						1		Play
IDom7	1				3			5			7		1	Play
IIDom7	7		1				3			5			7	Play

Figure 5. Part of the object KeySpace used in the construction of melodies

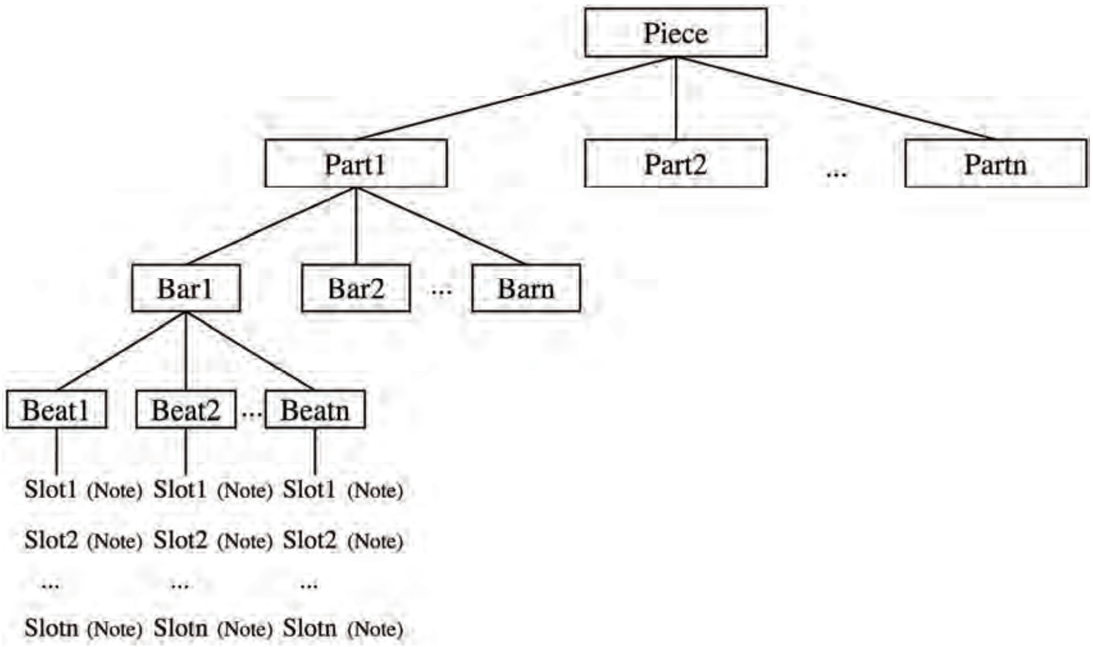

Piece
Part1
Part2
...
Partn
Bar1
Bar2
...
Barn
Beat1
Beat2
...
Beatn
Slot1 (Note) Slot1 (Note) Slot1 (Note)
Slot2 (Note) Slot2 (Note) Slot2 (Note)
...
Slotn (Note) Slotn (Note) Slotn (Note)

Figure 6. The object melody

step the relevant parameters (indicated by highlighting) may be set. Each aspect can at each time be generated anew, for instance with a different parameter setting to study its effect on the resulting melody. The order in which the aspects can be constructed is controlled by enabling and disabling the construction buttons and by means of little "lights" located left of the buttons (see **Figure 4**). By pushing the "Done" button the generation of a Part is terminated.

3.2.2 Structure-Based Construction

In the chord-based model it is possible to construct a melody with a pre-defined structure of parts. Examples of such structures are: A B A, A A1 A2 B, Ant Cons, [A VBV] [Dev AI BI], in which the capital letters A-G are used to identify parts, a digit following a letter to indicate a variation, and the Roman numerals I, II ... VII to identify the initial or final harmony of a part; Ant and Cons are used in the Antecedent-Consequent formula. Other reserved words are Dev for Development, and Coda indicating a part at the end of a structure. The user can add custom structures using the above coding.

The default way of constructing is part by part in which the user separately sets the parameters for each part. However, it is also possible to construct a structured melody automatically in which case the different parts are all constructed consecutively using randomly selected Time and Pitch parameters and default variations (if applicable). A detailed description is provided on the interface.

An example of a structure based melody is shown below.

3.2.3 Editing

By right-clicking in the melody a drop-down menu appears that enables to add a note or to change the pitch of an existing note (**Figure 7**).

A melody can also be transposed or elaborated. The degree of elaboration is determined by the setting of the density parameter in the "Time parameters" pane. Elaborations may again be removed.

3.2.4 (Re)arrangement

After one or more Parts have been generated, Parts can be removed, moved and duplicated using the drop-down menu shown in **Figure 7**.

3.2.5 Transformation

After a Part has been finished (the Done button having been pushed) the user may apply one or more transformations. Such transformations are most useful to create variations of a Part. The Transformation panel is opened either by right clicking on the Part and selecting "Transform Part", or by left-clicking and pushing the appearing "Transform?" button. Next, the Transform pane will be shown and the Part being transformed will be highlighted in red. At present the following transformations can be applied to a Part: *Increase elaboration*, *Decrease elaboration*, *Transpose 1* (applies a transposition of 1, 2, or 3 steps within the current harmony), *Transpose 2* (applies a transposition of 1-7 degree steps up or down, thereby adjusting the harmony), *Change Pitch* (changes the pitches of the Part keeping rhythm and harmony intact), *Change Rhythm* (changes (only) the rhythm of the Part). More options will be added in the future. Applied transformations can be undone by clicking Edit: Undo Editing

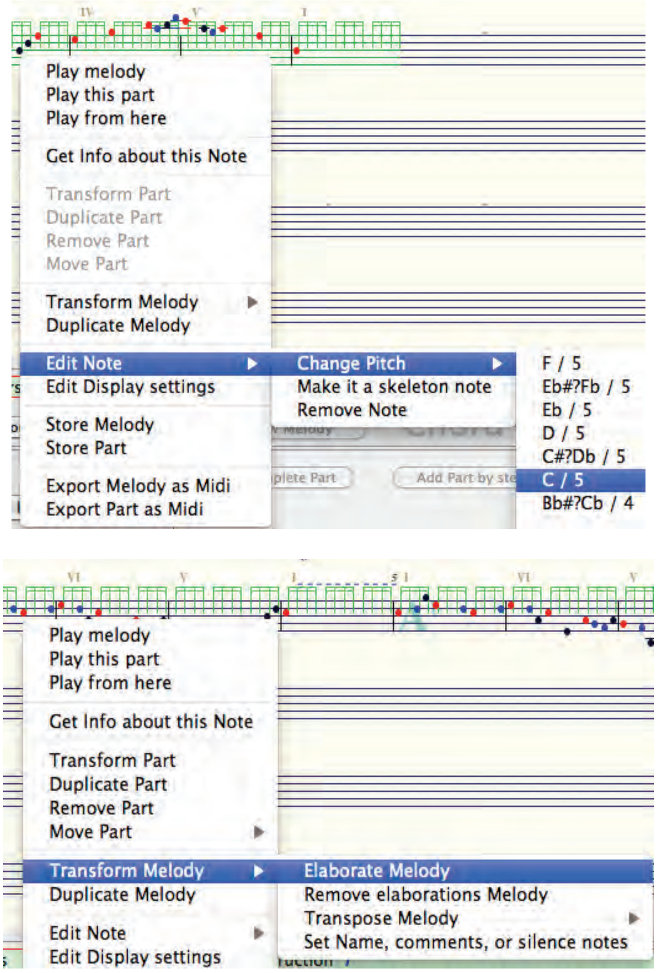


Figure 7. The drop-down menu showing various possibilities to manipulate aspects of the melody

(ctrl z, or cmd z).

3.2.6 Example of a Melody Generated within the Chord-Based Model

Once the parameters for rhythm, harmony and contour are set, a skeleton consisting of chord tones is generated. For the contour a choice can be made between the following shapes: Ascending, Descending, U-shaped, Inverted U-shaped, Sinusoid, and Random (based on [43, 54]). Subsequently, the skeleton may be elaborated using an algorithm that interpolates notes between the notes of the skeleton. **Figure 8** presents a skeleton melody **(a)** and three ornamentations differing in density **(b, c, d)**.

3.2.7 Example of a Melody Generated within the Scale-Based Model

After a rhythm has been generated a skeleton consisting of an ascending or descending scale fragment is created. Next a harmony fitting with the skeleton is set, after which the skeleton may be ornamented. As explained in Subsection 2.5.2, four types of ornamentation have been implemented: Repetition, Neighbor note, Skip, and Chord transposition. In addition, the user may select a "Random" ornamentation in which case an ornamentation is randomly chosen for each bar. A detailed description of this model can be found on the interface of the program. **Figure 9** presents a melody based on a skeleton consisting of a descending scale fragment that is subsequently ornamented in different ways.

3.2.8 Example of a Multi-Part Melody

Figure 10 shows a multi-part melody based on an A B A1 B1 A B structure in which A and B are an Antecedent and Consequent part respectively, and A1 and B1 variations of A and B.

A few more examples of multi-part melodies can be found at http://www.socsci.ru.nl/~povel/Melody/.

3.3 Display and Play Features

Each step in the construction of a melody is displayed in the "Melody" pane and can be made audible by clicking the Play button in the "Play parameters" pane. The main parameters in this pane refer to: *what* is being played (Melody only, or combined with Root and/or Beat; Rhythm only), *Tempo*, and *Instrument* (both for Melody and Root).

3.4 Storing and Saving Melodies

A melody can be stored temporarily in the "Melody store" on the interface (right click in the Melody). Melodies in the Melody store can be played, sent back to the Melody pane, or pre- or post-fixed to the melody in the Melody pane (by right clicking in the Melody store). A melody can also be exported in MIDI format (right click in the Melody). The melodies stored in the Melody Store can be saved to Disk in so-called mg2 format by clicking the "Save to File" or "Save to File As" buttons. Upon clicking the "Export as MIDI" button, all melodies in the "Melody store" pane are stored to Disk in MIDI format.

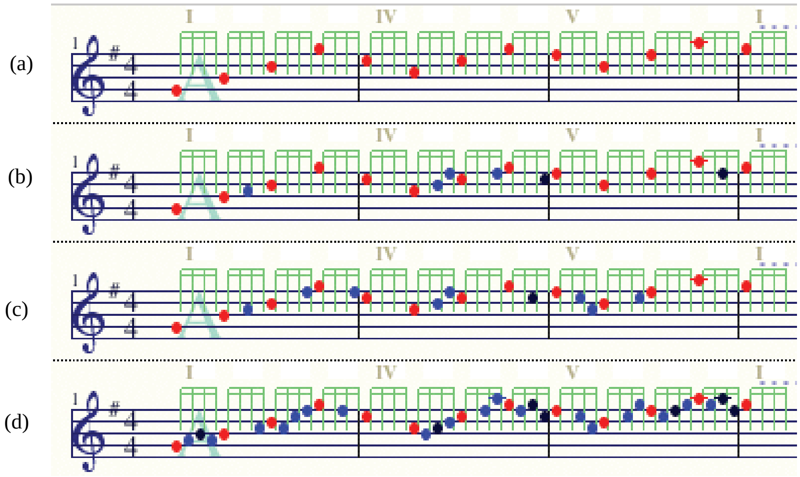


Figure 8. Example of a melody generated within the chord-based model. (a) A skeleton melody; (b)-(d) the skeleton elaborated with an increasing number of ornamental notes. Color code: red: skeleton note, black: chord-note; blue: non-chord note

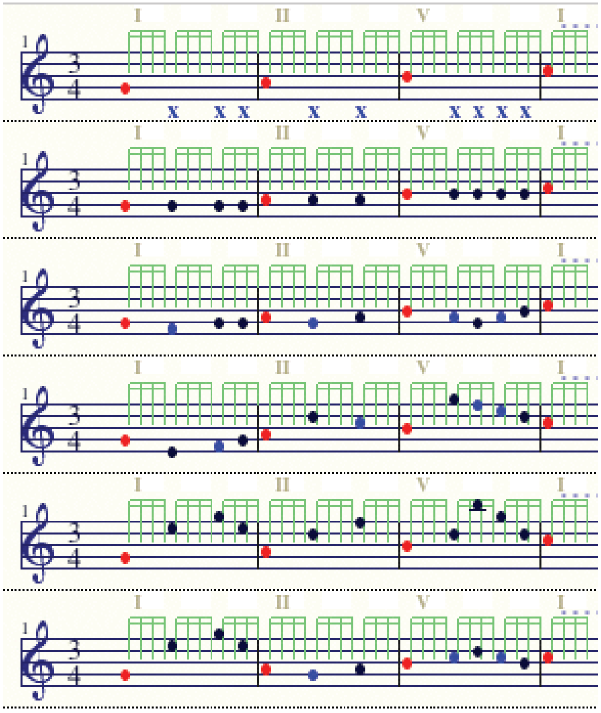


Figure 9. Example of a scale-based melody, showing a skeleton consisting of an ascending scale fragment, followed by 5 different elaborations: repetition, neighbor, skip, chord transposition, and random (chord transposition in the first bar and neighbor in bars 2 and 3). Color code: red: skeleton note, black: chord-note; blue: non-chord note

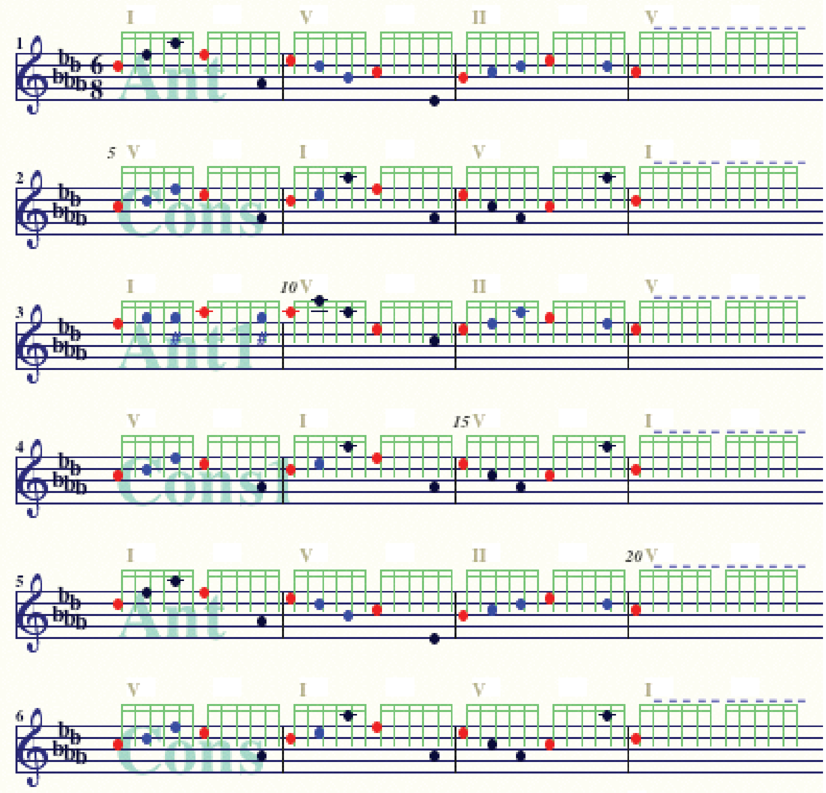


Figure 10. Example of a multipart melody built on the structure Ant, Cons, Ant1, Cons2, Ant, Cons, *i.e.* an Antecedent-Consequent formula repeated twice in which the first repetition is slightly varied (ornamented)

3.5 Additional Functions

Apart from the functions that may be invoked from the main interface, a few additional functions are available through the Menu, such as displaying the current KeySpace, displaying properties of the notes in the melody, changing the display parameters, displaying information about the operation of the program, etc.

4. Conclusions

This article describes a device for the algorithmic construction of tonal melodies, named Melody Generator. Starting from a few basic assumptions, a set of notions were formulated concerning the context of tonal music and the construction principles used in its generation. These notions served as the starting points for the development of a computer program that generates tonal melodies the characteristics of which are controlled by means of adjustable parameters. The architecture and implementation of the program, including its multipurpose interface, are discussed in some details.

Presently, we are working on a formal and systematic validation of the melodies produced by the device in order to obtain a reliable estimation of the positive and negative qualities of its various aspects. This evaluation will provide the necessary feedback to further develop and refine the algorithm and its underlying theoretical notions. Results of this evaluation will be reported in a forthcoming publication.

5. Acknowledgements

My former students Hans Okkerman, Peter Essens, René van Egmond, Erik Jansen, Jackie Scharroo, Reinoud Roding, Jan Willem de Graaf, Albert van Drongelen, Thomas Koelewijn, and Daphne Albeda, all played an indispensable role in the development of the ideas underlying the project described in this paper.

I am most grateful to David Temperley for his support over the years, to Hubert Voogd for his help in tracing and crushing numerous tenacious bugs, and Herman Kolk and Ar Thomassen for suggestions to improve the text.

REFERENCES

[1] W. V. D. Bingham, "Studies in Melody," *Psychological Review*, Monograph Supplements, Vol. 12, Whole No. 50, 1910.

[2] H. Gardner, "The Mind's New Science: A History of the Cognitive Revolution," Basic Books, 1984.

[3] D. Deutsch, (Ed.) "The Psychology of Music," Academic Press, 1999.

[4] C. L. Krumhansl, "Cognitive Foundations of Musical Pitch," Oxford University Press, 1990.

[5] D. Deutsch and J. Feroe, "The Internal Representation of Pitch Sequences," *Psychological Review*, Vol. 88, No. 6, 1981, pp. 503-522.

[6] K. Hirata and T. Aoyagi, "Computational Music Representation Based on the Generative Theory of Tonal Music and the Deductive Object-Oriented Database," *Computer Music Journal*, Vol. 27, No. 3, 2003, pp. 73-89.

[7] F. Lerdahl and R. Jackendoff, "A Generative Theory of Tonal Music," MIT Press, Cambridge, 1983.

[8] H. C. Longuet-Higgins and M. J. Steedman, "On Interpreting Bach," In: B. Meltzer and D. Michie, Eds., *Machine Intelligence*, University Press, 1971.

[9] A. Marsden, "Generative Structural Representation of Tonal Music," *Journal of New Music Research*, Vol. 34, No. 4, 2005, pp. 409-428.

[10] D. J. Povel, "Internal Representation of Simple Temporal Patterns," *Journal of Experimental Psychology*: *Human Perception and Performance*, Vol. 7, No. 1, 1981, pp. 3-18.

[11] D. J. Povel, "A Model for the Perception of Tonal Music," In: C. Anagnostopoulou, M. Ferrand and A. Smaill, Eds., *Music and Artificial Intelligence*, Springer Verlag, 2002, pp. 144-154.

[12] D. Temperley, "An Algorithm for Harmonic Analysis," *Music Perception*, Vol. 15, No. 1, 1997, pp. 31-68.

[13] D. Temperley, "Music and Probability," MIT Press, Cambridge, 2007.

[14] D. Cope, "Experiments in Music Intelligence," MIT Press, 1996.

[15] E. Miranda, "Composing Music with Computers," Focal Press, 2001.

[16] R. Rowe, "Machine Musicianship," MIT Press, 2004.

[17] G. Papadopoulos and G. Wiggins, "AI Methods for Algorithmic Composition: A Survey, a Critical View and Future Prospects," *Proceedings of the AISB*'99 *Symposium on Musical Creativity*, Edinburgh, Scotland, 6-9 April 1999.

[18] C. Ames, "The Markov Process as a Compositional Model: A Survey and Tutorial," *Leonardo*, Vol. 22, No. 2, 1989, pp. 175-187.

[19] E. Cambouropoulos, "Markov Chains as an Aid to Computer Assisted Composition," *Musical Praxis*, Vol. 1, No. 1, 1994, pp. 41-52.

[20] K. Ebcioglu, "An Expert System for Harmonizing Four Part Chorales," *Computer Music Journal*, Vol. 12, No. 3, 1988, pp. 43-51.

[21] M. Baroni, R. Dalmonte and C. Jacoboni, "Theory and Analysis of European Melody," In: A. Marsden and A. Pople, Eds., *Computer Representations and Models in Music*, Academic Press, London, 1992.

[22] D. Cope, "Computer Models of Musical Creativity," A-R Editions, 2005.

[23] M. Steedman, "A Generative Grammar for Jazz Chord Sequences," *Music Perception*, Vol. 2, No. 1, 1984, pp. 52-77.

[24] J. Sundberg and B. Lindblom, "Generative Theories in Language and Music Descriptions," *Cognition*, Vol. 4, No. 1, 1976, pp. 99-122.

[25] G. Wiggins, G. Papadopoulos, S. Phon-Amnuaisuk and A. Tuson, "Evolutionary Methods for Musical Composition," Partial *Proceedings of the* 2*nd International Conference CASYS*'98 *on Computing Anticipatory Systems*, Liège, Belgium, 10-14 August 1998.

[26] J. A. Biles, "Genjam: A Genetic Algorithm for Generating Jazz Solos," 1994. http://www.it.rit.edu/~jab/

[27] P. Todd, "A Connectionist Approach to Algorithmic Composition," *Computer Music Journal*, Vol. 13, No. 4, 1989, pp. 27-43.

[28] P. Toiviainen, "Modeling the Target-Note Technique of Bebop-Style Jazz Improvisation: An Artificial Neural Network Approach," *Music Perception*, Vol. 12, No. 4, 1995, pp. 399-413.

[29] M. Baroni, S. Maguire and W. Drabkin, "The Concept of a Musical Grammar," *Musical Analysis*, Vol. 2, No. 2, 1983, pp. 175-208.

[30] M. Baroni, R., Dalmonte and C. Jacoboni, "A Computer-Aided Inquiry on Music Communication: The Rules of Music," Edwin Mellen Press, 2003.

[31] A. Marsden, "Representing Melodic Patterns as Networks of Elaborations," *Computers and the Humanities*, Vol. 35, No. 1, 2001, pp. 37-54.

[32] H. C. Longuet-Higgins and C. S. Lee, "The Perception of Musical Rhythms," *Perception*, Vol. 11, No. 2, 1982, pp. 115-128.

[33] D. J. Povel and P. J. Essens, "Perception of Temporal Patterns," *Music Perception*, Vol. 2, No. 4, 1985, pp. 411-440.

[34] F. Gouyon, G. Widmer, X. Serra and A. Flexer, "Acoustic Cues to Beat Induction: A Machine Learning Approach," *Music Perception*, Vol. 24, No. 2, 2006, pp. 177-188.

[35] D. J. Povel and H. Okkerman, "Accents in Equitone Sequences," *Perception and Psychophysics*, Vol. 30, No. 6, 1981, pp. 565-572.

[36] D. J. Povel, "Exploring the Elementary Harmonic Forces in the Tonal System," *Psychological Research*, Vol. 58, No. 4, 1996, pp. 274-283.

[37] F. Lerdahl, "Tonal Pitch Space," *Music Perception*, Vol. 5, No. 3, 1989, pp. 315-350.

[38] F. Lerdahl, "Tonal Pitch Space," Oxford University Press, Oxford, 2001.

[39] E. H. Margulis, "A Model of Melodic Expectation," *Music Perception*, Vol. 22, No. 4, 2005, pp. 663-714.

[40] D. J. Povel and E. Jansen, "Harmonic Factors in the Perception of Tonal Melodies," *Music Perception*, Vol. 20, No. 1, 2002, pp. 51-85.

[41] J. J. Bharucha, "Anchoring Effects in Music: The Resolution of Dissonance," *Cognitive Psychology*, Vol. 16, No. 4, 1984, pp. 485-518.

[42] D. J. Povel and E. Jansen, “Perceptual Mechanisms in Music Processing,” *Music Perception*, Vol. 19, No. 2, 2001, pp. 169-198.

[43] E. Toch, “The Shaping Forces in Music: An Inquiry into the Nature of Harmony, Melody, Counterpoint, Form,” Dover Publications, NY, 1977.

[44] J. Bamberger, “Developing Musical Intuitions,” Oxford University Press, Oxford, 2000.

[45] Schoenberg, “Fundamentals of Musical Composition,” Faber and Faber, 1967.

[46] H. Schenker, “Harmony,” Chicago University Press, Chicago, 1954.

[47] V. Zuckerkandl, “Sound and Symbol,” Princeton University Press, Princeton, 1956.

[48] H. Schenker, “Free Composition (Der freie Satz),” E. Oster, Translated and Edited, Longman, New York, 1979.

[49] M. Neuburg, “REALbasic. The Definitive Guide,” O’Reilly, 1999.

[50] M. G. Brown, “A Rational Reconstruction of Schenkerian Theory,” Thesis Cornell University, 1989.

[51] M. G. Brown, “Explaining Tonality,” Rochester University Press, Rochester, 2005.

[52] Jonas, “Das Wesen des musikalischen Kunstwerks : eine Einführung in die Lehre Heinrich Schenkers,” Saturn, 1934.

[53] W. Piston and M. Devoto, “Harmony,” Victor Gollancz, 1989.

[54] D. Huron, “The Melodic Arch in Western Folksongs,” *Computing in Musicology*, Vol. 10, 1989, pp. 3-23.

J. Software Engineering & Applications, **2010, 3, 696-703**
doi:10.4236/jsea.2010.37079 Published Online July 2010 (http://www.SciRP.org/journal/jsea)

Lightness Perception Model for Natural Images

Xianglin Meng, Zhengzhi Wang

College of Mechatronics Engineering and Automation, National University of Defense Technology, Changsha, China.
Email: xlmeng@nudt.edu.cn

Received May 11th, 2010; revised May 28th, 2010; accepted June 12th, 2010.

ABSTRACT

A perceptual lightness anchoring model based on visual cognition is proposed. It can recover absolute lightness of natural images using filling-in mechanism from single-scale boundaries. First, it adapts the response of retinal photoreceptors to varying levels of illumination. Then luminance-correlated contrast information can be obtained through multiplex encoding without additional luminance channel. Dynamic normalization is used to get smooth and continuous boundary contours. Different boundaries are used for ON and OFF channel diffusion layers. Theoretical analysis and simulation results indicate that the model could recover natural images under varying illumination, and solve the trapping, blurring and fogging problems to some extent.

Keywords: *Perception, Lightness Anchoring, Filling in, Adaptation, Multiplex Contrast*

1. Introduction

Studies show that human could perceive a wide dynamic range of lightness from dim moonlight to dazzling sunlight. Human visual system has stable perceptual capability for scenes under variable illuminations, which is referred to as lightness constancy. First of all, two concepts must be clarified: luminance and lightness. Luminance values denote light intensities within the retinal image while lightness is related to our perceived world. Luminance values in the retinal image are a product, not only of the actual physical shade of gray of the imaged surfaces, but also, and even more so, of the intensity of the light illuminating those surfaces. The luminance of any region of the retinal image can vary by a factor of no more than thirty to one as a function of the physical reflectance of that surface. However, it can vary as a factor of a billion to one as a function of the amount of illumination on that surface. The net result is that any given luminance value can be perceived as literally any shade of gray, depending on its context within the image. Despite the challenge, human perceive shades of surface grays with rough accuracy [1]. Lightness represents the cortical perceptual result of retinal stimuli and the process of lightness perception can be understood as a mapping from natural image input to visual percept output.

BCS/FCS (Boundary Contour System/Feature Contour System) proposed by Grossberg *et al*. is representative of visual lightness perception model. Its extended versions have explained a mass of psychological experimental data [2,3]. BCS/FCS model discounts the illuminant to obtain contrast information through retinal preprocessing. Further processing is made by visual cortex to get boundary contour and surface. However, such illumination discounting information can just estimate relative measurements of reflectance of the surface. Visual cortex needs to compute the absolute lightness values that exploit the full dynamic range to perceive effectively. That's just the so-called anchoring problem.

Grossberg believed that boundaries and surfaces are visual perceptual units, and proposed aFILM model in 2006 [4]. The model augments a lightness anchoring stage in the framework of BCS/FCS. Sepp *et al*. proposed a multi-scale filling-in model to reconstruct the image surface lightness [5]. It extends the confidence-based filling-in model to multi-scale processing, thus speeding up the filling-in process. A key aspect of visual cognition research is to process natural images effectively while explaining psychological data, which facilitates higher cognition process such as object recognition and provides inspirations to machine vision.

This paper presents a neural dynamic model to simulate the mapping process from luminance to lightness according to recent neurophysiological and psychological experimental findings. The proposed model could recover natural images under varying levels of illumination, and solve the trapping, blurring and fogging problems to some extent.

2. Model Description

After retinal photoreceptors receive the input image, retinal adaptation occurs firstly. The light-adapted signal goes to multiplex coding and boundary detection. The contrast information from multiplex coding is used to recover the absolute surface lightness, and the boundary contour is to block activity diffusion between surfaces. Final perceptual lightness output is obtained through surface filling-in mechanism. The overall model architecture is depicted in **Figure 1**.

2.1 Retinal Adaptation

The model retina calculates the steady-state of retinal adaptation to a given input image [4]. It adapts the response of photoreceptors to varying levels of incoming light, since otherwise the visual process could be desensitized by saturation right at the photoreceptor. Light adaptation, at the photoreceptor outer segment, protects each photoreceptor from saturation by using intracellular temporal adaptation that shifts the photoreceptor sensitivity curve [6]. As illustrated in **Figure 2**, the light-adapted signal is further processed at the photoreceptor inner segment where it gets feedback from a horizontal cell (HC) that is connected with other HCs by gap junctions [7], forming a syncytium that is sensitive to spatial contrast. HC inhibition further adjusts the sensitivity curve to realize spatial contrast adaptation. It is assumed that the permeability of gap junctions between HCs decreases as the difference of the inputs to HCs from coupled photoreceptors increases. The model retina hereby segregates and selectively suppresses signals in regions that have strong contrasts, such as a light source. HCs connections are not constrained to nearest neighbors but reach farther regions. Inhibition of the HC on the photoreceptor controls the output of the photoreceptor by modulating Ca^{2+} influx at the photoreceptor inner segment. This feedback prevents the output from being saturated by localized high-contrast input signals.

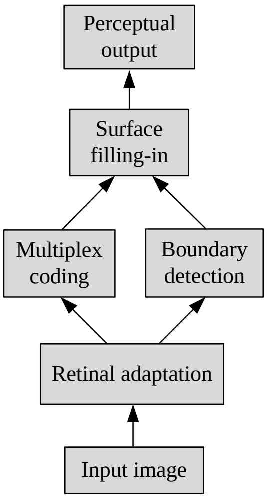


Figure 1. Model block diagram

The outer segment of retinal photoreceptor can be modeled by the equation:

$$s_{ij}(t) = I_{ij} \bullet g_{ij}(t) \tag{1}$$

where (i, j) denotes spatial position, $s_{ij}(t)$ represents the output of outer segment, I_{ij} is the input image, $g_{ij}(t)$ is an automatic gain control term simulating negative feedback mediated by Ca^{2+} ions and follows:

$$\frac{dg_{ij}}{dt} = (A_g - g_{ij}) - g_{ij}(B_I I_{ij} + B_{\bar{I}} \bar{I}) \tag{2}$$

where A_g, B_I and $B_{\bar{I}}$ are constants. $\bar{I}$ denotes spatialaverage of input image that approximates the effect of recent image scanning by sequences of eye movements. The second term describes the inactivation of $g_{ij}(t)$ by

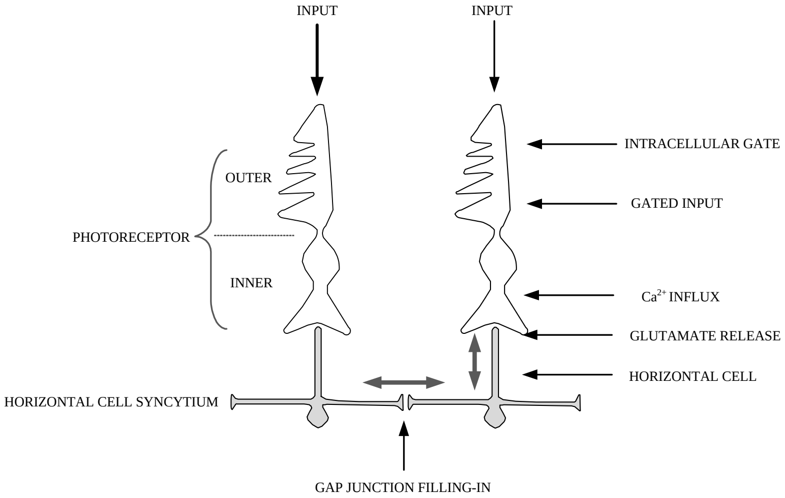


Figure 2. Circuit of retinal adaptation

the present I_{ij} and $\bar{I}$. The inner segment of the photoreceptor receives the signal s_{ij} from the outer segment and also gets feedback H_{ij} from the horizontal cell. HC modulation of the output of the inner segment is modeled by the equation:

$$S_{ij} = \frac{s_{ij}}{C_H \exp(H_{ij}) \cdot (C_s - s_{ij}) + 1} \tag{3}$$

where $C_s = A_g / B_I$, C_H is a constant. The relationship between H_{ij} and HC activity h_{ij} follows:

$$H_{ij} = \frac{a_H h_{ij}^2}{b_H^2 + h_{ij}^2} \tag{4}$$

where a_H and b_H are constants. The activity of an HC connected to its neighbors through gap junctions is defined as diffusion equation:

$$\frac{dh_{ij}}{dt} = -h_{ij} + \sum_{(p,q)\in N_{ij}} P_{pqij}(h_{pq} - h_{ij}) + S_{ij} \tag{5}$$

where P_{pqij} is the permeability between cells at (i,j) and (p,q),

$$P_{pqij} = 1 - \frac{1}{1 + \exp[-(|S_{pq} - S_{ij}| - \beta)/\lambda]} \tag{6}$$

β, λ are constants. N_{ij} is the neighborhood of size r to which the HC at (i,j) is connected:

$$N_{ij} = \{(p,q) : \sqrt{(i-p)^2 + (j-q)^2} \le r,\ (p,q) \ne (i,j)\} \tag{7}$$

2.2 Multiplex Contrast Code

Lightness anchoring model generally incorporates an extra luminance-driven channel to recover absolute lightness in addition to retinal contrast channels. Maybe multi-scale band-pass filters are used to get contrast and luminance information, such as aFILM model which acquires contrast information through small scale filtering and obtains luminance information through large scale filtering [4]. There is evidence showing that a larger disinhibitory surround exists outside of the classical receptive field of retinal ganglion cells. Accordingly, a multiplex retinal code is proposed to solve the anchoring problem. The code is composed of retinal contrast responses, where contrast responses are locally modulated by brightness (ON cell) or darkness (OFF cell). The modulation is implemented by an extensive disinhibitory surround or outer surround (OS), an annulus which is situated beyond the classical center-surround receptive of retinal ganglion cells as shown in **Figure 3** [8]. The classical receptive field is sensitive to contrast information and outer surround is sensitive to local luminance. Then luminance-correlated contrast information can be obtained through multiplex encoding without additional luminance channels. Also, it is plausible from a neuron-physiological point of view.

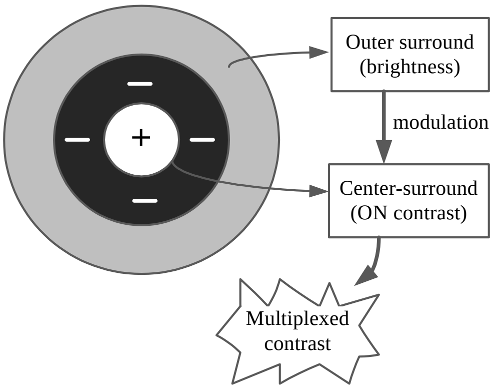


Figure 3. Multiplex retinal code

Due to the asymmetry phenomenon of ON cell and OFF cell responses for the classical center-surround receptive field of retinal ganglion cells, a self-inhibition mechanism is adopted:

$$\frac{dx_{ij}^+(t)}{dt} = -\alpha x_{ij}^+ + I_{ij}^c - I_{ij}^s - [I_{ij}^c - I_{ij}^s]^+ \cdot x_{ij}^+ \tag{8}$$

$$\frac{dx_{ij}^-(t)}{dt} = -\alpha x_{ij}^- + I_{ij}^s - I_{ij}^c - [I_{ij}^s - I_{ij}^c]^+ \cdot x_{ij}^- \tag{9}$$

where x_{ij}^+ and x_{ij}^- are ON cell and OFF cell responses respectively, $[\bullet]^+ \equiv \max(\bullet, 0)$, and α is a decay factor. $I_{ij}^c = S_{ij}$ is center input. $I_{ij}^s = (S \otimes G_s)_{ij}$ is surround input. G_s is a Gaussian kernel. "$\otimes$" denotes convolution operator. The last term of the right side of above equations denotes self-inhibition which is used to solve the response asymmetry problem [9].

Let m^+ and m^- denote local luminance-correlated multiplexed contrast response of ON and OFF channels respectively. The activity of the outer surround is computed by convolution with a Gaussian kernel: $O = Norm[S] \otimes G_{\sigma_O}$. $Norm[\bullet]$ implements the normalization operator which maps the input into [0 1]. Outer surround activity acts to multiplicatively gate the classical center-surround responses of retinal ganglion cells. So the multiplexed contrast responses are defined as

$$m_{ij}^+ = \frac{O_{ij}}{\beta_o + O_{ij}} \bullet [x_{ij}^+]^+, \quad m_{ij}^- = \frac{1 - O_{ij}}{\beta_o + 1 - O_{ij}} \bullet [x_{ij}^-]^+ \tag{10}$$

where β_o is a saturation constant.

2.3 Boundary Detection

The same retinal contrast information is used in both boundary detection and filling-in mechanism in BCS/

FCS model. In this paper, we use different contrast information. Boundary detection is completed through a dynamic normalization network [10]. First, we define an operator $\Re_\lambda[\bullet]$:

$$\Re_\lambda[x] = \frac{1+e^{-|\lambda|}}{1+e^{-\lambda\cdot x}}x \tag{11}$$

We can get

$$\begin{aligned}\lim_{\lambda\to\infty}\Re_\lambda[x] &= \lim_{\lambda\to\infty}\frac{1+e^{-|\lambda|}}{1+e^{-\lambda\cdot x}}x \\ &= \begin{cases} x, & x>0 \\ 0, & x=0 \\ 0, & x<0 \end{cases} = \max(x,0)\end{aligned} \tag{12}$$

similarly,

$$\lim_{\lambda\to-\infty}\Re_\lambda[x] = \min(x,0) \tag{13}$$

$$\Re_\lambda[x]\,|_{\lambda=0} = x \tag{14}$$

for simplicity we denote:

$$\Re_\infty = \lim_{\lambda\to\infty}\Re_\lambda,\quad \Re_{-\infty} = \lim_{\lambda\to-\infty}\Re_\lambda,\quad \Re_0 = \Re_\lambda\,|_{\lambda=0}.$$

Subsequently, we define nonlinear diffusion operators:

$$\Phi^+ f_{ij} = \sum_{(p,q)\in N_{ij}^4}\Re_\infty[f_{pq}-f_{ij}] \tag{15}$$

$$\Phi^- f_{ij} = \sum_{(p,q)\in N_{ij}^4}\Re_{-\infty}[f_{pq}-f_{ij}] \tag{16}$$

where $N_{ij}^4 = \{(i+1,j),(i-1,j),(i,j-1),(i,j+1)\}$ is a four nearest neighborhood. The dynamic normalization equations are:

$$\frac{da_{ij}(t)}{dt} = \kappa\cdot\Phi^- a_{ij}(t) + S_{ij}\delta(t-t_0) \tag{17}$$

$$\frac{db_{ij}(t)}{dt} = \kappa\cdot\Phi^+ b_{ij}(t) + S_{ij}\delta(t-t_0) \tag{18}$$

$$\frac{dc_{ij}(t)}{dx} = b_{ij}(0-c_{ij}) - a_{ij}(1-c_{ij}) + S_{ij} \tag{19}$$

$$\frac{dd_{ij}(t)}{dx} = b_{ij}(1-d_{ij}) - a_{ij}(0-d_{ij}) - S_{ij} \tag{20}$$

κ is a diffusion coefficient, $a_{ij}, b_{ij}, c_{ij}, d_{ij}$ are min-syncytium, max-syncytium, normalized ON type and OFF type cell activity respectively. δ denotes Dirac's delta function. Seen from (17), a cell a_{ij} of the min-syncytium may only decrease its activity along with time, as long as there exists any activity gradient between this cell and its four nearest neighbors. The result of diffusion is that a_{ij} decreases to the global minimum value. In an analogous fashion, a cell b_{ij} of the max-syncytium finally gets the global maximum value of activity. We can get smooth and continuous boundary contour by early dynamics of the dynamic normalization network. We denote the resultant responses as y^+, y^-. ON contour and OFF contour representing diffusion barriers are defined as

$$w_{ij}^+ = \frac{thresh_{\Theta_w}(y_{ij}^+)}{\beta_w + thresh_{\Theta_w}(y_{ij}^+)} \tag{21}$$

$$w_{ij}^- = \frac{thresh_{\Theta_w}(y_{ij}^-)}{\beta_w + thresh_{\Theta_w}(y_{ij}^-)} \tag{22}$$

where β_w is a saturation constant,

$$thresh_{\Theta_w}(x) = [Norm[x]-\Theta_w]^+,\quad \Theta_w\in[0,1].$$

The detected boundaries are always discontinuous due to noise or other factors, which allows for activity exchange between adjacent surface representations. Consequently, perceived luminance contrasts between surfaces decrease, because they eventually adopt the same value of perceptual activity. It's the so-called fogging problem. In order to counteract fogging, an interaction zone around contours is defined. Within this zone, brightness activity and darkness activity undergo mutual inhibition, leading to a slow-down of diffusion rate at boundary leaks. Thus fogging is decelerated and surface edges will appear blurry at boundary gaps. Let

$$z_{ij} = thresh_{\Theta_z}(w_{ij}^+ + w_{ij}^-) \tag{23}$$

with a threshold value Θ_z. Interaction zone activity Z is defined as

$$Z = \frac{z}{\beta_z + z}\otimes G_{\sigma_z} \tag{24}$$

where β_z is a saturation constant, G_{σ_z} is a Gaussian kernel with standard deviation σ_z.

2.4 Surface Filling-in

The multiplex contrast responses diffuse within regions formed by boundary contours to form perceptual surface representations. Diffusion layers are computed by dynamic equations:

$$\begin{aligned}\frac{df_{ij}^+(t)}{dt} &= \gamma w_{ij}^-(E_{in}-f_{ij}^+) + \sum_{(p,q)\in N_{ij}^4}P_{ijpq}^+\Re_\infty[f_{pq}^+ - f_{ij}^+] \\ &\quad + \delta(t-t_0)m_{ij}^+\end{aligned} \tag{25}$$

$$\begin{aligned}\frac{df_{ij}^-(t)}{dt} &= \gamma w_{ij}^+(E_{in}-f_{ij}^-) + \sum_{(p,q)\in N_{ij}^4}P_{ijpq}^-\Re_\infty[f_{pq}^- - f_{ij}^-] \\ &\quad + \delta(t-t_0)m_{ij}^-\end{aligned} \tag{26}$$

where $f_{ij}^{+/-}$ represent brightness and darkness activity. γ is a constant, E_{in} is an inhibitory reversal potential. Diffusion coefficients:

$$P^{+}_{ijpq} = \frac{1}{1+\varepsilon(Z_{ij}[f^{-}_{ij}]^{+} + Z_{pq}[f^{-}_{ij}]^{+})} \qquad (27)$$

$$P^{-}_{ijpq} = \frac{1}{1+\varepsilon(Z_{ij}[f^{+}_{ij}]^{+} + Z_{pq}[f^{+}_{ij}]^{+})} \qquad (28)$$

where ε is a constant. From equations above we can see that OFF contours are used to block brightness activity diffusion while ON contours are employed to block darkness activity diffusion. In this way, we can alleviate the activity trapping problem. The perceptual activity of surface representations

$$p_{ij} = \frac{g_{leak}V_{rest} + [f^{+}_{ij}]^{+} - [f^{-}_{ij}]^{+}}{g_{leak} + [f^{+}_{ij}]^{+} + [f^{-}_{ij}]^{+}} \qquad (29)$$

where g_{leak} is leakage conductance, and V_{rest} is the resting potential.

3. Simulation Results

In order to validate the effectiveness of the proposed model, we first evaluate the performance of each stage, then the lightness perception performance of the whole model is tested using natural images. The size of test images in simulation is 256 × 256.

3.1 Retinal Adaptation

Firstly, we evaluate the performance of retinal adaptation. The simulation results are illustrated in **Figure 4**. The original input image, HC activity and retinal adaptation output are shown in **Figures 4(a)-(c)** respectively. The input image has intensive contrast, thus we can hardly see the dark scene. Through retinal adaptation processing, we cannot only see the clouds in sky of bright region, but also see the sand of dark region, even small rocks on the ground. It's the result of light adaptation and spatial contrast adaptation of the retina.

3.2 Boundary Contour Detection

In this section, we compare the boundary detection performance of dynamic normalization with that of classical laplacian linear filtering method. **Figures 5(a) and (b)** are ON and OFF boundaries of Lena image obtained by laplacian method. **Figures 5(c) and (d)** are detection results from dynamic normalization. As we can see from **Figure 5**, the boundary contours obtained by dynamic normalization are smoother and more continuous, which will block the diffusion between different regions more effecttively in later filling-in processing, thus alleviating

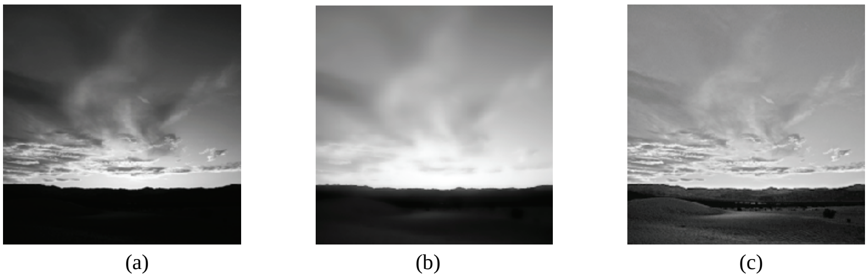


Figure 4. Retinal adaptation simulation. (a) stimulus; (b) HC activity; (c) adaptation output

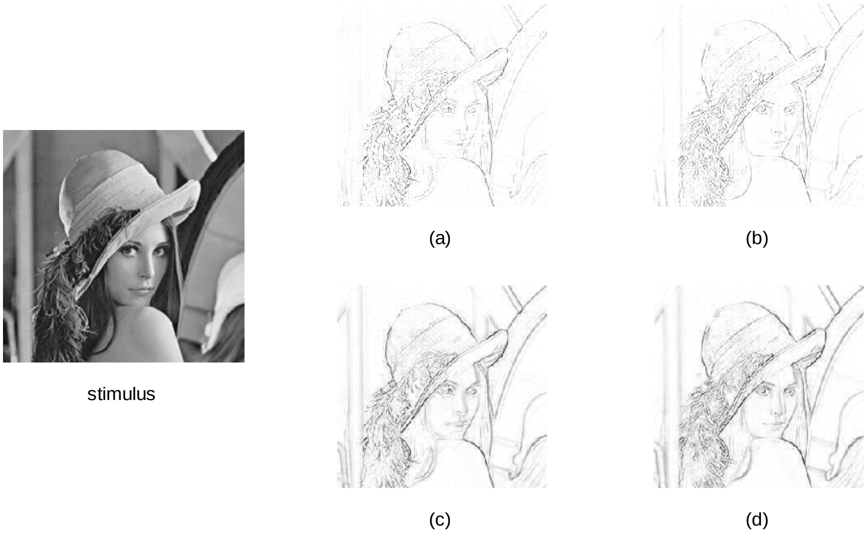


Figure 5. Boundary contour detection

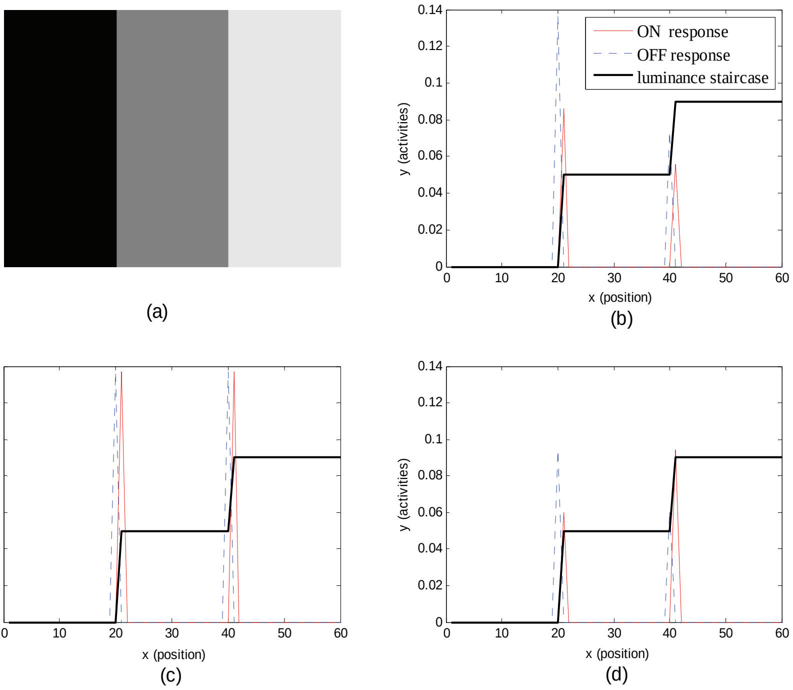
the fogging problem.

3.3 Multiplex Contrasts

For the synthetic stimulus in **Figure 6(a)**, one-dimensional luminance staircase is shown in **(b)-(d)** by the black solid lines. The red solid lines denote ON responses while the blue dashed lines correspond to OFF responses. **Figure 6(b)** shows the responses of classical center-surround receptive field. OFF responses are always higher than ON responses around edges, and both decrease with luminance increase. The asymmetry problem of ON and OFF responses is solved by self-Inhibition mechanism illustrated in **Figure 6(c)**, where ON and OFF responses are only sensitive to contrast and insensitive to luminance. Therefore, we could modulate ON responses with local brightness and OFF with local darkness, thus getting luminance-correlated multiplex contrast responses. As seen in **Figure 6(d)**, ON responses increase while OFF responses decrease as the luminance increases.

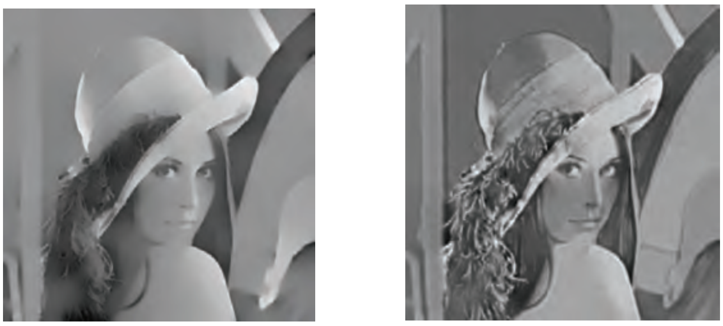
3.4 Surface Filling-in

Nonlinear diffusion is used to implement surface filling-in and recover absolute perceived lightness. Most filling-in models have blurring, trapping and fogging problems such as confidence-based filling-in. As described previously, in this paper, different boundaries are used for the brightness and the darkness diffusion layers. As a consequence, trapping can be weakened to some extent. Besides, the adoption of interaction zone could alleviate fogging. **Figure 7(a)** shows the result of confidence-based filling-in, demonstrating serious fogging problem. **Figure 7(b)** is the filling-in result of the proposed model. It can be seen that absolute lightness is recovered while



(a) (b) (c) (d)

Figure 6. Multiplex contrast responses



(a) (b)

Figure 7. Surface filling-in results. (a) confidence-based filling-in; (b) nonlinear diffusion

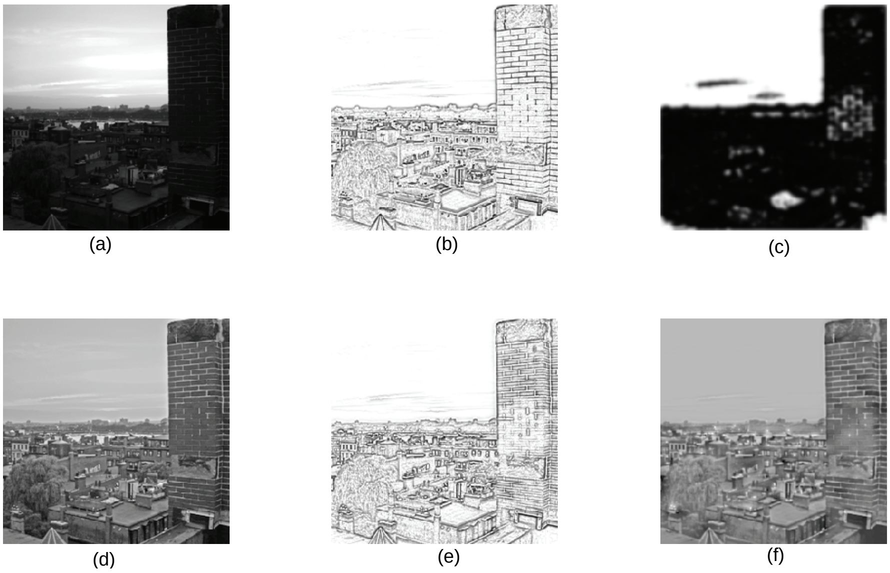


Figure 8. Lightness perception of a real-world image. (a) stimulus; (b) ON contour; (c) interaction zone; (d) retinal adaptation; (e) OFF contour; (f) perception result

reserving contrast, without fogging problem.

Finally, we test the performance of the whole model using a real-world image. Simulation results of different processing stages are shown in **Figure 8**. We can found that the proposed model could recover absolute lightness of natural images under varying illuminating conditions. Comparing with aFILM model, it can get smoother and more continuous contours than BCS, and explicitly weakens blurring, trapping and fogging problems to some degree. Moreover, the proposed model has the advantage of illuminating adaptation in comparison with Sepp's brightness perception model. It provides an alternative approach for lightness perception.

4. Conclusions

In this paper, a perceptual lightness anchoring model based on visual cognition is proposed. It can recover absolute lightness of real-world images using filling-in mechanism from single-scale boundaries. Further, it can weaken trapping, blurring and fogging to some extent. Reasonable perceptual results could be obtained for natural images under varying illumination conditions. The proposed model could be applied to image enhancement and image reconstruction in machine vision, and facilitate robust processing of higher levels such as object recognition. However, there are not normative objective measurements to evaluate the performance of visual cognition model. Most measurements are qualitative, and it makes no exception for this paper. Additionally, this model recovers absolute lightness from single scale channel. So its ability of noise suppression is weaker than multi-scale processing. We will study further in these aspects.

5. Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 60835005) and Scientific Research Plan Projects of NUDT (No. JC09-03-04). The authors are very grateful to the reviewers for their valuable remarks and constructive comments on this paper.

REFERENCES

[1] A. Gilchrist, C. Kossyfidis, F. Bonato and T. Agostini, "An Anchoring Theory of Lightness Perception," *Psychological Review*, Vol. 106, No. 4, 1999, pp. 795-834.

[2] M. Anton-Rodrigurez, F. J. Diaz-Pernaz, J. F. Diez-Higuera, M. Martinez-Zarzuela, D. Gonzalez-Ortega and D. Boto-Giralda, "Recognition of Coloured and Textured Images through a Multi-Scale Neural Architecture with Orientational Filtering and Chromatic Diffusion," *Neurocomputing*, Vol. 72, No. 16-18, 2009, pp. 3713-3725.

[3] A. Fazl, S. Grossberg and E. Mingolla, "View-Invariant Object Category Learning: How Spatial and Object Attention are Coordinated Using Surface-Based Attentional Shrouds," *Cognitive Psychology*, Vol. 58, No. 1, 2009, pp. 1-48.

[4] S. Grossberg and S. Hong, "A Neural Model of Surface Perception: Lightness, Anchoring, and Filling-in," *Spatial Vision*, Vol. 19, No. 2-4, 2006, pp. 263-321.

[5] W. Sepp and H. Neumann, "A Multi-Resolution Filling-in Model for Brightness Perception," 9*th International Conference on Artificial Neural Networks*, 1999, pp. 461-466.

[6] M. Vanleeuwen, I. Fahrenfort, T. Sjoerdsma, R. Numan and M. Kamermans, “Lateral Gain Control in the Outer Retina Leads to Potentiation of Center Responses of Retinal Neurons,” *The Journal of Neuroscience*, Vol. 29, No. 19, 2009, pp. 6358-6366.

[7] H. Jouhou, K. Yamamoto, M. Iwasaki and M. Yamada, “Acidification Decouples Gap Junctions but Enlarges the Receptive Field Size of Horizontal Cells in Carp Retina,” *Neuroscience Research*, Vol. 57, No. 2, 2007, pp. 203-209.

[8] M. S. Keil, G. Cristobal, T. Hansen and H. Neumann, “Recovering Real-World Images from Single-Scale Boundaries with a Novel Filling-In Architecure,” *Neural Networks*, Vol. 18, No. 10, 2005, pp. 1319-1331.

[9] R. D. S. Raizada and S. Grossberg, “Context-Sensitive Binding by the Laminar Circuits of V1 and V2: A Unified Model of Perceptual Grouping, Attention, and Orientation Contrast,” Boston University, 2000.

[10] M. S. Keil, “Local to Global Normalization Dynamic by Nonlinear Local Interaction,” *Physica D*, Vol. 237, No. 6, 2008, pp. 732-744.

J. Software Engineering & Applications, **2010, 3, 704-708**
doi:10.4236/jsea.2010.37080 Published Online July 2010 (http://www.SciRP.org/journal/jsea)

Design of Radial Basis Function Network Using Adaptive Particle Swarm Optimization and Orthogonal Least Squares

Majid Moradi Zirkohi, Mohammad Mehdi Fateh, Ali Akbarzade

Department of Electrical and Robotic Engineering, Shahrood University of Technology, Shahrood, Iran.
Email: m.moradi@ieee.org

Received May 12th, 2010; revised June 13th, 2010; accepted June 15th, 2010.

ABSTRACT

This paper presents a two-level learning method for designing an optimal Radial Basis Function Network (*RBFN*) *using Adaptive Velocity Update Relaxation Particle Swarm Optimization algorithm* (*AVURPSO*) *and Orthogonal Least Squares algorithm* (*OLS*) *called as OLS-AVURPSO method. The novelty is to develop an AVURPSO algorithm to form the hybrid OLS-AVURPSO method for designing an optimal RBFN. The proposed method at the upper level finds the global optimum of the spread factor parameter using AVURPSO while at the lower level automatically constructs the RBFN using OLS algorithm. Simulation results confirm that the RBFN is superior to Multilayered Perceptron Network* (*MLPN*) *in terms of network size and computing time. To demonstrate the effectiveness of proposed OLS-AVURPSO in the design of RBFN, the Mackey-Glass Chaotic Time-Series as an example is modeled by both MLPN and RBFN.*

Keywords: *Radial Basis Function Network, Orthogonal Least Squares Algorithm, Particle Swarm Optimization, Mackey-Glass Chaotic Time-Series*

1. Introduction

The radial basis function network (RBFN) as an alternative to the multilayered perceptron neural network (MLPN) has been studied intensively [1]. The RBFN has the universal approximation ability; therefore, the RBF neural network can be used for the interpolation problem. A Gaussian radial basis function is highly nonlinear, and provides some good characteristics for incremental learning with many well-defined mathematical features [2]. It is a powerful scheme for learning, identification, equalization, and control of nonlinear dynamic systems.

The training of feed forward ANN is based on nonlinear optimization technique; however, it may get trapped at a local minimum during the learning procedure using the gradient descent algorithm. The RBFN is an alternative method for aforementioned method.

The performance of RBFN critically depends upon the chosen RBF centers [3]. A new distance measure, which is superior to the Euclidean distance, was applied for selecting the centers from highly correlated input vector [4]. Another approach was proposed to determine the centers of RBF networks based on sensitivity analysis [5]. However, this approach has not considered the role of spread factor that is a significant factor to increase the accuracy of results. In contrast, Orthogonal Least Squares (OLS) algorithm [6] selects required number of RBF centers depending on the value of spread factor. The OLS employs the forward regression procedure to reduce the size of RBFN resulting in an adequate and parsimonious RBFN.

The OLS algorithm has solved a crucial problem of how to select RBFN centers very well; however, it doesn’t give a method for selecting the spread factor of an RBF [6].

The PSO was first introduced by Kennedy and Eberhart in 1995 [7]. Through the simulation of a simplified social system, the behavior of PSO can be treated as an optimization process. As compared with other optimization algorithms, the PSO requires less computational time. Therefore, it has successfully been applied to solve many problems [8-10].

This paper proposes a novel adaptive version of a two-level learning method for constructing a RBFN using a Velocity Update Relaxation Particle Swarm Optimization (VURPSO) presented in [11]. The novelty is to find the global optimum of spread factor parameter at the upper level using adaptive velocity update PSO namely AVURPSO that has more convergence speed and accurate response than VURPSO. At the lower level, it constructs a parsimonious RBFN using the OLS algorithm.

This paper is organized as follows: Section 2 describes the RBFN. Section 3 formulates the PSO algorithm and develops the AVURPSO algorithm. Section 4 introduces Mackey-Glass chaotic time-series. Section 5 presents simulation results and finally Section 6 concludes the paper.

2. RBFN

An RBFN has a feed forward structure consisting of a single hidden layer of locally tuned units which are fully interconnected to an output layer of linear units, as shown in **Figure 1**. All hidden units simultaneously receive the p-dimensional real valued input vector. The input vector to the network is passed to the hidden layer nodes via unit connection weights. The hidden layer consists of a set of radial basis functions. The hidden layer node calculates the Euclidean distance between the center and the Network input vector and then passes the result to the radial basis function. All the radial basis functions are, usually, of the same type. Thus the hidden layer performs a fixed nonlinear transformation and it maps the input space onto a new space. The output layer, then, implements a linear combiner on this new space and the only adjustable parameters are the weights of this linear combiner. These parameters can be determined using the linear least Squares method, which is an important advantage of this method. An RBFN is designed to perform a nonlinear mapping from the input space to the hidden space, followed by a linear mapping from the hidden space to the output space. Thus, the network represents a map from the p-dimensional input space to m-dimensional output space, according to:

$$d_i = w_{0i} + \sum_{j=1}^{h} w_{ji}\phi(\|x - c_j\|, \beta_j) \tag{1}$$

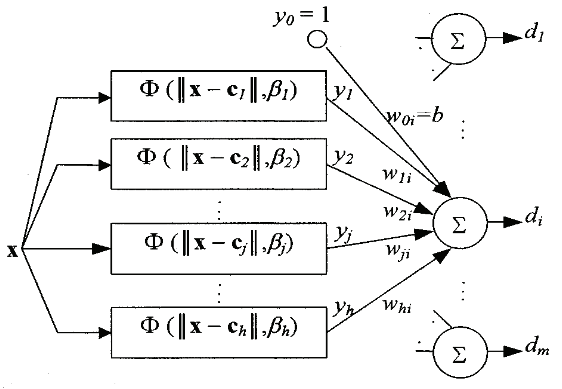


Figure 1. RBF Structure

where for $i = 1, 2, ..., m$, $j = 1, 2, ..., h$, x is an input vector. n , m and h are the number of input nodes, output nodes, and hidden units respectively. c_j is the j-th center node in the hidden layer, $\|x - c_j\|$ denotes Euclidean distance, $\phi(.)$ is a nonlinear transfer function called as RBF, w_{ji} is the weighting value between the i-th center and the j-th output node, β is the real constant known as spread factor. Equation (1) reveals that the output of network is computed as a weighted sum of the hidden layer outputs. The nonlinear output of the hidden layer is radically symmetrical. In this paper, the most widely used Gaussian function for the j-th hidden unit is chosen as follows:

$$\phi(\|x - c_j\|, \beta_i) = \exp\left(\frac{-\|x - c_j\|^2}{2\beta_j^2}\right) \tag{2}$$

The accuracy is controlled by three parameters: the number of radial basis functions or hidden units, centers of the hidden units, and the spread factor.

A common learning strategy for an RBF network is to randomly select some input data sets as the RBF centers in the hidden layer. The weights between hidden and output layer can then be estimated by using the stochastic gradient approach. The main disadvantage of this method is that it is very difficult to quantify how many numbers of center should be adequate to cover the input vector space. Furthermore, the training algorithm is possibly getting stuck into local minimum. To overcome these shortages, this paper develops the OLS-AVUURPSO to construct the RBFN.

3. AVURPSO Algorithm

The PSO algorithm is performed as follows: the unknown parameters are called the particles. Starting with a randomly initialization, the particles will move in a searching space to minimize an objective function. The parameters are estimated through minimizing the objective function. The fitness of each particle is evaluated according to the objective function for updating the best position of particle and the best position among all particles as two goals in each step of computing. Each article is directed to its previous best position and the global best position among particles. Consequently, the particles tend to fly towards the better searching areas over the searching space. The velocity of i-th particle v_i will be calculated as follows [7]:

$$\begin{aligned} v_i(k+1) = {} & w\, v_i(k) + c_1 r_1 (pbest_i(k) - x_i(k)) \\ & + c_2 r_2 (gbest(k) - x_i(k)) \end{aligned} \tag{3}$$

where in the k-th iteration, x_i is the position of particle, $pbest_i$ is the previous best position of particle, $gbest$ is the previous global best position of particles, w is the inertia weight, c_1 and c_2 are the acceleration coefficients namely the cognitive and social scaling parameters, r_1 and r_2 are two random numbers in the range of [0 1]. It is worthy to note that the inertia weight has not been in the first version of PSO [7]. If the inertia weight in (3) is set to 1, the first version of PSO is obtained.

The new position of *i*-th particle is then calculated as

$$x_i(k+1) = x_i(k) + v_i(k+1) \tag{4}$$

The PSO algorithm performs repeatedly until the goal is achieved. Number of iterations can be set to a specific value as a goal of optimization.

The first version of PSO has been improved in terms of convergence and accuracy, so far. To control the velocity, if the velocity v_i exceeds a maximum value of $v_{\max}$, then v_i is set to $v_{\max}$. In many applications, $v_{\max}$ has been set to $x_{\max}$. In addition, the inertia weight was proposed to control the velocity [12] as

$$w = (w_1 - w_2)\left(\frac{k_{\max} - k}{k_{\max}}\right) + w_2 \tag{5}$$

where w decreases from a higher value w_1 to a lower value w_2, and $k_{\max}$ is the maximum number of iteration. Moreover, the velocity was modified to improve the convergence [12] as

$$v_i(k) = \chi \begin{pmatrix} \omega v_i(k-1) + \\ c_1 r_1 (pbest_i(k) - x_i(k)) \\ + c_2 r_2 (gbest(k) - x_i(k)) \end{pmatrix} \tag{6}$$

where for $\varphi = c_1 + c_2$ and $\varphi \geq 2$, χ is given by

$$\chi = \frac{2}{\left|4 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \tag{7}$$

Adopting low values for c_1 and c_2 allows the particle to roam far from the target regions before being tugged back. On the other hand, adopting high values results in abrupt movement toward or passes the target regions. Therefore, c_1 and c_2 were introduced [12] as

$$\begin{aligned} c_1 &= (c_{1i} - c_{1f})\left(\frac{k_{\max} - k}{k_{\max}}\right) + c_{1f} \\ c_2 &= (c_{2i} - c_{2f})\left(\frac{k_{\max} - k}{k_{\max}}\right) + c_{2f} \end{aligned} \tag{8}$$

where c_{1i} and c_{2i} are the initial values of c_1 and c_2, and c_{1f} and c_{2f} are the final values of c_1 and c_2, respectively. Actually, the best solutions were determined over the full range of search for changing c_1 from 2.5 to 0.5 and c_2 from 0.5 to 2.5. With a large value of c_1 and a small value of c_2 at the beginning, particles are allowed to move around the search space instead of moving toward the $pbest_i$. A small value of c_1 and a large value of c_2 allow the particles converge to the $gbest_i$ in the latter part of the optimization.

In traditional PSO, the velocities of the particles are limited in the range of [$v_{\min}$; $v_{\max}$]. Usually $v_{\min}$ and $v_{\max}$ are set to $x_{\min}$ and $x_{\max}$, respectively. The positions of the particles are given in the range of [$x_{\min}$; $x_{\max}$]. Thus, evaluating the obtained results according to the limits for confining or rejecting the results takes extra computational burden. Velocity update relaxation particle swarm optimization (VURPSO) postulates the boundary velocity validity checking without checking the validity of positions in every iteration cycle.

In traditional PSO algorithm, the velocity is updated at every iteration cycle. In contrast, in velocity-updating relaxation [11], the velocity of each particle kept unchanged if its fitness at current iteration is better than one at preceding iteration; otherwise the particles' velocity is updated as stated by (3). As a result, the computational efficiency is enhanced. The new position of particle is then calculated as:

$$x_i^{k+1} = (1 - mf) \times x_i^k + (mf) \times v_i^{k+1} \tag{9}$$

where mf is called momentum factor given in the range of $0 < mf < 1$ because the new position vector is a point on the line between the former position vector, x_i^k, and the new velocity vector, v_i^{k+1}. In many applications, mf was given a constant. VURPSO exhibits to have strong global search ability at the beginning of the run and strong local search near the end of the run. The use of velocity update relaxation in traditional PSO helps to reduce the computational efforts.

In order to speed up the convergence speed, we propose a novel adaptive VURPSO strategy named AVURPSO. In this new strategy we change the momentum factor adaptively as follow:

$$mf = mf_1 + (mf_2 - mf_1)\left(\frac{k_{\max} - k}{k_{\max}}\right) \tag{10}$$

where mf decreases from a higher value mf_1 to a lower value mf_2. Moreover, we use (6) instead of (3).

4. Mackey-Glass Chaotic Time-Series

The Mackey-Glass Chaotic Time-Series [13] is stated as

$$\dot{x}(t) = -0.1x(t) + \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} \tag{11}$$

where we set $\tau = 17$ and $x(t-\tau) = 1.2$ for $0 \le t \le \tau$. The Mackey-Glass Chaotic Time-Series is modeled by a RBFN.

This time series is chaotic, and so there is no clearly defined period. The series is not converged or diverged and the trajectory is highly sensitive to initial condition. The input training data for RBF predictor is a four-dimensional vector in the following form of

$$w(t) = [x(t-18) \quad x(t-12) \quad x(t-6) \quad x(t)] \tag{12}$$

The output training data corresponds to the trajectory prediction.

$$y(t) = x(t+6) \tag{13}$$

A set of data with 1000 samples is obtained. We use the first 500 samples for training and the second 500 samples for validation (Test Data). The data is shown in **Figure 2**.

5. Simulation Results

To verify the performance of proposed method we present two comparisons. First of all, AVURPSO and VURPSO are compared in the Design of RBFN. Then, the RBFN and the MLPN are compared in Modeling of the Time Series.

5.1 Comparing AVURPSO and VURPSO in the Design of RBFN

As mentioned in previous section, the input of RBFN is the train data. For a given value to the spread factor, the OLS algorithm provides an optimum number of centers (NC) in RBFN from the training patterns. Next, it estimates the bias vector and weighting matrix using least square error technique for the prescribed sum of squared errors (SSE).

The RBFN is trained using OLS algorithm while we use the training patterns, the values of β given in **Table 1,** and $SSE = 0.01$. We should find the optimum value of spread factor to improve the results since β significantly affects the NC as confirmed by **Table 1**.

The Fitness function is defined in order to optimize the value of spread factor as follow:

$$Fitness = \frac{1}{Q}\sum_{i=1}^{Q}(y_{real} - y_{net})^2 + NC \tag{14}$$

where Q denotes the number of samples, while y_{real} is the real output, and y_{net} is the desired network output. In this approach, to escape from the local minima the fitness function is changed to

$$Fitness = \left[\frac{1}{Q}\sum_{i=1}^{Q}(y_{real} - y_{net})^2\right]^6 + NC \tag{15}$$

The number of particles and the maximum value of iterations are selected 12 and 50, respectively. And, mf is varied from 0.5 to 0.3. **Table 2** presents the obtained optimal value of spread factor, and the NC from these two methods.

The AVURPSO has obtained $\beta = 0.649$ resulting in a less NC and a less MSE in both sets of train data and test data. Moreover, the AVURPSO has a higher speed of convergence as shown in **Figure 3**.

Table 1. The role of spread factor in determining the RBFN centers and MSE

β	$\beta = 0.01$	$\beta = 0.1$	$\beta = 0.8$	$\beta = 1$
NC	490	162	42	499
MSE (Train)	0.007	0.0096	0.0092	0.0227
MSE (Test)	87.94	0.012	0.0086	0.0022

Table 2. Optimal value of spread factor and the NC obtained from the VURPSO and AVURPSO methods

method	β	**NC**	**MSE (Train)**	**MSE (Test)**
VURPSO	0.66	42	0.0088	0.0085
AVURPSO	0.649	36	0.0083	0.0079

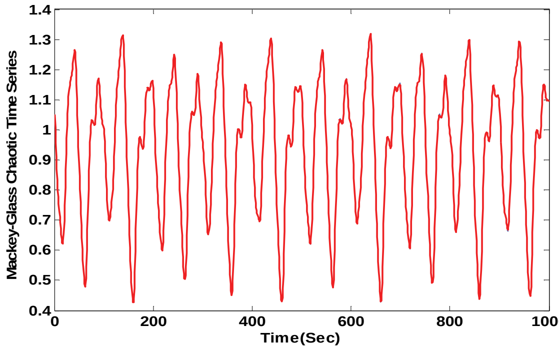


Figure 2. Chaotic time-series behavior

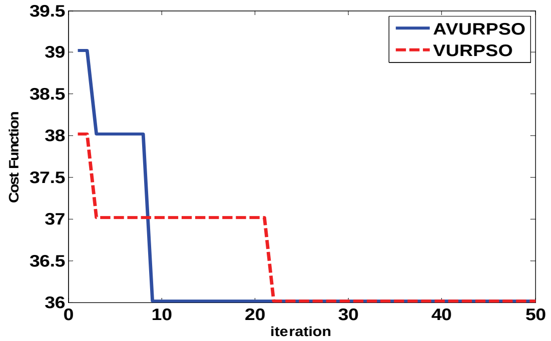


Figure 3. Comparing the convergence speed

Table 3. A comparison on computing time

algorithm	RBFN	MLPN
CPU Time	3.81	21.63

5.2 Comparing RBFN and MLPN in Modeling of the Time Series

First a MLPN with optimum topology is designed. Then, we compare the RBFN with the optimum MLPN in modeling of the time series algorithm. The architecture of MLPN consists of an input layer, one or more hidden layers, and an output layer. The MLPN is trained with Levenberg-Marquardt back propagation. To obtain the MLPN with optimal topology, different topologies are tested by AVRPSO algorithm as

1) One hidden layer with 7 neurons.

2) Two hidden layers with variant neurons between 1 to 20 at each.

The optimum topology is obtained (8, 5, 1) from option 2.

Now, we compare RBFN and the MLPN in modeling the time series on the computing time as shown in **Table 3**.

The RBFN possibility needs more neurons than MLPN; however, RBFN often can be designed in a fraction of time that it takes to train MLPN.

6. Conclusions

A two-level learning method has been presented for designing the RBFN using OLS-AVURPSO method. The proposed method at the upper level finds the global optimum of the spread factor parameter using the AVURPSO algorithm while at the lower level automatically constructs the RBFN using the OLS algorithm. To verify the performance of proposed method, two comparisons have been presented. First, the AVURPSO algorithm and the VURPSO algorithm are compared in the design of RBFN. Second, the RBFN and the MLPN are compared in the modeling of the time-series. The superiority of the AVURPSO algorithm to the VURPSO algorithm is verified due to obtaining a less NC, a less MSE and a higher speed of convergence. In the modeling of the Mackey-Glass time-series, simulation results confirm that the RBFN is superior to MLPN in terms of the network size and computing time.

REFERENCES

[1] S. Chen, S. A. Billings, C. F. N. Cowan and P. M. Grant, "Non-Linear Systems Identification Using Radial Basis Functions," *International Journal of Systems Science*, Vol. 21, No. 12, 1990, pp. 2513-2539.

[2] M. M. Gupta and L. Jin, "Static and Dynamic Neural Networks," John Wiley, 2003.

[3] R. Segal and M. L. Kothari, "Radial Basis Function (RBF) Network Adaptive Power System Stabilizer," *IEEE Transactions on Power Systems*, Vol. 15, No. 2, 2000, pp. 722-727.

[4] S. A. Billings and X. Hong. "Dual Orthogonal Radial Basis Function Networks for Nonlinear Time Series Prediction," *Neural Networks*, Vol. 11, No. 3, 1998, pp. 479-493.

[5] D. Shi, D. S. Yeung and J. Gao. "Sensitivity Analysis Applied to the Construction of Radial Basis Function Networks," *Neural Networks*, Vol. 18, No. 7, 2005, pp. 951-957.

[6] S. Chen, C. F. N. Cowan and P. M. Grant, "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks," *IEEE Transactions on Neural Networks*, Vol. 2, No. 2, March 1991, pp. 302-309.

[7] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Proceedings of IEEE International Conference on Neural Networks*, Vol. 4, 1995, pp. 1942-1948.

[8] S. Naka, T. Genji, T. Yura and Y. Fukuyama, "A Hybrid Particle Swarm Optimization for Distribution State Estimation," *IEEE Transactions on Power Systems*, Vol. 18, No. 1, 2003, pp. 60-68.

[9] M. Clerc, "The Swarm and the Queen: Towards the Deterministic and Adaptive Particle Swarm Optimization," *Proceedings of the Congress on Evolutionary Computation*, Washington, DC, 1999, pp. 1951-1957.

[10] A. Alfi and M. M. Fateh, "Parameter Identification Based on a Modified PSO Applied to Suspension System," *Journal of Software Engineering & Applications*, Vol. 3, 2010, pp. 221-229.

[11] A. Chatterjee, "Velocity Relaxed and Craziness-Based Swarm Optimized Intelligent PID and PSS Controlled AVR System," *Electrical Power and Energy Systems*, Vol. 31, No. 7-8, 2009, pp. 323-333.

[12] A. Ratnaweera and S. K. Halgamuge, "Self Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficient," *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, 2004, pp. 240-255.

[13] L. Wang, "A Course in Fuzzy Systems and Control," Prentice-Hall International, 1997.

J. Software Engineering & Applications, **2010, 3, 709-717**
doi:10.4236/jsea.2010.37081 Published Online July 2010 (http://www.SciRP.org/journal/jsea)

Secure Multi-Party Proof and its Applications

Chunming Tang[1,2], Shuhong Gao[3]

[1]School of Mathematics and Information Sciences, Guangzhou University, Guangzhou, China; [2]State Key Laboratory of Information Security, Chinese Academy of Science, Beijing, China; [3]Department of Mathematical Sciences, Clemson University, Clemson, USA.
Email: ctang@gzhu.edu.cn

Received April 30th, 2010; revised May 24th, 2010; accepted June 1st, 2010.

ABSTRACT

We define a new type cryptographical model called secure multi-party proof that allows any t players and a verifier to securely compute a function $f(x_1,...,x_t)$: each of the players learns nothing about other players' input and about the value of f, and the verifier obtains the value of f and it's validity but learns nothing about the input of any of the players. It is implemented by a protocol using oblivious transfer and Yao's scrambled circuit. We prove that our protocol is secure if the players and the verifier are semi-honest (i.e. they follow the protocol) and polynomial time bounded. The main applications of our protocol are for electronic voting and electronic bidding.

Keywords: *Multi-Party Proof, Multi-Party Computation, Electronic Voting, Electronic Bidding*

1. Introduction

1.1 Secure Multi-Party Computation and its Disadvantage

In a secure multi-party computation, a set of n parties with private inputs wish to jointly and securely compute a function that depends on the individual inputs of the parties. This computation should be such that each party receives its correct output (correctness), and none of the parties learn anything beyond their prescribed output (privacy). For example, in an election protocol, correctness ensures that no coalition of parties can influence the outcome of the election beyond just voting outcome for their preferred candidate, whereas privacy ensures that no parties learn anything about the individual votes of other parties. Secure multi-party computation can be viewed as the task of carrying out a distributed computation, while protecting honest parties from the malicious manipulation of dishonest (or corrupted) parties.

In all secure multi-party computation, only participants obtain information about the value of the function f computed. In some applications, some party say an arbiter, other than the participants, may want to know the function value and needs to be sure of its validity, yet the arbiter learns nothing about the secret inputs of the participants. Here's a possible scenario. Assume a company needs to appoint a new manager for a department. The administrators of the company hope that the manager is elected by the staff in this department only. The staff could elect a new manager by using an electronic voting protocol from a secure multi-party computation. However, these administrators are usually not in this department, hence they may not be convinced that the election result is valid.

Another main application of secure multi-party computation is the design of electronic bidding protocols. Usually, all participants jointly run a secure multi-party computation protocol and decide the winner; as a result, only all participants know who the winner is. However, the sponsor in any electronic bidding is not a participant; hence the sponsor may not be sure about the winner. Also, in some time the participants may not be allowed to know the winner, as the winner wants to be kept anonymous.

1.2 Our Contributions

We define a new type cryptographical model called secure multi-party proof that allows any t players and a verifier to securely compute a function $f(x_1,...,x_t)$. The model requires the following properties: each of the players learns nothing about other players' input and nor any information about the value of f, and the verifier obtains the value of f and it's validity but learns nothing about the input of any of the players. We implement this model by a protocol using oblivious transfer and

Yao's scrambled circuit. We prove that our protocol is secure if the players and the verifier are semi-honest (*i.e.* they follow the protocol) and polynomial time bounded. Based on our secure multi-party proof, our protocol can be used for electronic voting and electronic bidding.

1.3 Related Work

A great deal of work has been done about secure multi-party computation. Secure computation for two parties was first formulated by Yao [1] in 1982. In [2,3], Lindell and Pinkas gave a complete and explicit proof of Yao's protocol for secure two-party computation. Goldreich, Micali and Wigderson [4] showed how to securely compute any multivariate function (even if malicious adversaries are present); see [5] for complete proof of their results. Ben-Or, Goldwasser and Wigderson [6] (and, independently, Chaum, Crepeau and Damgard [7]) study secure multiparty computation in the secure channels setting. They show that: 1) If the adversary is eavesdropping then there exist $\left(\left\lceil \frac{n}{2}\right\rceil - 1\right)$-secure protocols for computing any function. 2) If the adversary is Byzantine, then any function can be $\left(\left\lceil \frac{n}{3}\right\rceil - 1\right)$-securely computed. Furthermore, they show that these bounds on the number of corruptions are tight. These protocols can be shown secure in the presence of non-adaptive adversaries. Adaptive security (*i.e.*, security in the presence of adaptive adversaries) is provable in certain variants of this setting.

Goldwasser and Levin [8] study the case of Byzantine adversaries where a majority of the parties may be corrupted. Chor and Kushilevitz [9] deal with secure multiparty computation with majority of the parties corrupted in the secure channels setting. Goldreich, Goldwasser and Linial [10] study secure multiparty computation in the presence of insecure channels and computationally unlimited adversaries. Ostrovsky and Yung [11] study secure multiparty computation in the presence of secure channels and mobile adversaries. Micali and Rogaway [12], and also Beaver [13], propose definitions for secure multiparty computation in the secure channels setting in the presence of adaptive adversaries. Other types of secure multi-party computation include adaptively secure multi-party computation [14], almost-everywhere secure computation [15], concurrent secure multi-party computation [16], and fair multi-party computation [17-19] and so on.

1.4 Organization

The paper is organized as follows. We start with some basic definitions and tools in Section 2. Section 3 gives a formal model of secure multi-party proof. Section 4 constructs a secure multi-party proof for any polynomial-time computable function if all participants are semi-honest. Section 5 provides a general method to construct a protocol that can be used for electronic voting and electronic bidding. Finally, Section 6 outlines concluding remarks and future directions.

2. Preliminary and Basic Tools

Let n denote a positive integer. We say that a function $\mu(n)$ is negligible in n (or just negligible) if, for every polynomial $p(n)$, we have $\mu(n) < 1/p(n)$ for all sufficiently large n. Let S be an infinite set and the size of an element $s \in S$ is denoted by $|s|$. Suppose $X = \{X_s\}_{s \in S}$ and $Y = \{Y_s\}_{s \in S}$ are two ensembles of random distributions (or random variables). We say that X and Y are computationally indistinguishable if, for every non-uniform polynomial-time distinguisher D, the absolute difference $|\Pr(D(X_s)=1) - \Pr(D(Y_s)=1)|$ is negligible in $|s|$ (for $s \in S$). For a probabilistic machine M, we denote by $M(x)$ the output of M when the input is x. The value $M(x)$ is a probabilistic distribution, as it depends on some random values from an internal random tape used by the machine.

Semi-honest Adversaries vs. Malicious Adversaries. Loosely speaking, the aim of a secure multi-party protocol is to protect honest parties against dishonest behaviors by some other parties. Usually, adversaries are divided into semi-honest and malicious adversaries.

A semi-honest adversary controls one of the parties and follows the protocol specification exactly. However, it may try to learn more information than allowed by the protocol via analyzing the transcript of messages received.

A malicious adversary may arbitrarily deviate from the specified protocol. When considering malicious adversaries, there are certain undesirable actions that cannot be prevented. Specifically, a party may refuse to participate in the protocol or substitute its local input (and use instead a different input). The adversary may also abort the protocol prematurely so that the adversary may obtain its output while the honest party does not.

In this paper, we assume that all parties or adversaries are semi-honest.

Special Symmetric Encryption. In [2], a special symmetric encryption scheme was constructed that has indistinguishable encryption for multiple messages. This means that for any two messages $\overline{x}$ and $\overline{y}$, no polynomial-time adversary can distinguish an encryption of $\overline{x}$ from that of $\overline{y}$.

Definition 1 Let (G, E, D) be a symmetric encryption scheme and let the range of a key k denoted by $R_n(k) = \{E_k(x) : x \in \{0,1\}^n\}$.

1) We say that (G,E,D) has an elusive range if, for every probabilistic polynomial-time machine M and for every polynomial $p(n)$, we have

$$\Pr(M(k) \in R_n(k)) < 1/p(n)$$

for sufficiently large n, where k is a random binary string of length n.

2) We say that (G,E,D) has an efficiently verifiable range if there exists a probabilistic polynomial time machine M such that $M(1^n,k,c)=1$ if and only if $c \in R_n(k)$.

By convention, for every $c \notin R_n(k)$, we let $D_k(c) = \perp$.

In [2], Y. Lindell and B. Pinkas give a simple construction of a special symmetric encryption scheme. Let $F=\{f_k\}$ be a family of pseudorandom functions [11], where $f_k:(0,1)^n \to \{0,1\}^{2n}$ for $k \in \{0,1\}^n$. Then define

$$E_k(x) = (r, f_k(r) \oplus x0^n)$$

where $x \in \{0,1\}^n$, $r \in_R \{0,1\}^n$ and $x0^n$ denotes the concatenation of x and 0^n.

Oblivious Transfer. We will briefly describe the oblivious transfer protocol of [20]. This protocol is secure in the presence of semi-honest adversaries. Our description will be for the case that $x_0, x_1 \in \{0,1\}$; when considering semi-honest adversaries, the general case can be obtained by running the single-bit protocol many times in parallel. It is assumed that there is a family of permutations with trapdoors, so the permutations are one-way functions if the trapdoors are not given. Furthermore, $B(x)$ is a hardcore bit of the one-way functions, so computing $B(x)$ is equivalent to inverting the one-way functions (without using the trapdoors).

Suppose P_1 has two bits $x_0, x_1 \in \{0,1\}$ and P_2 has $\sigma \in \{0,1\}$. The goal is for P_1 to transfer x_σ to P_2 but P_1 does not know which of the two bits was transferred. This is called a 1-out-of-2 oblivious transfer.

Oblivious Transfer Protocol

1) P_1 randomly chooses a permutation-trapdoor pair (E,t) from a family of trapdoor permutations. P_1 sends E (but not the trapdoor t) to P_2.

2) P_2 chooses a random ν_σ in the domain of E and computes $\omega_\sigma = f(\nu_\sigma)$. In addition, P_2 chooses a random $\omega_{1-\sigma}$ in the range of E. P_2 sends (ω_0, ω_1) to P_1.

3) P_1 uses the trapdoor t and computes $\nu_0 = E^{-1}(\omega_0)$ and $\nu_1 = E^{-1}(\omega_1)$. Then computes $b_0 = B(\nu_0) \oplus x_0$ and $b_1 = B(\nu_1) \oplus x_1$, where B is a hard-core bit of E. Finally, P_1 sends (b_0, b_1) to P_2.

4) P_2 computes $x_\sigma = B(\nu_\sigma) \oplus b_\sigma$, which is the bit transferred to P_2 by P_1.

It was proven in [21] that the above protocol is secure if both of P_1 and P_2 are semi-honest.

3. Definition of Secure Multi-Party Proof

Suppose there are t players $P_1, P_2, \cdots, P_t$ with secret inputs $x_1, x_2, \cdots, x_t$, respectively, and a verifier V. We assume that all the players and the verifier are computationally bounded. In addition, assume that $f(x_1, x_2, \cdots, x_t)$ is a polynomial-time computable function, so it has a polynomial size circuit.

Definition 2. A multi-party Proof consists of two sub-protocols:

Computation sub-protocol: It is an ordinary multi-party computation among the players $P_1, P_2, \cdots, P_t$. The goal is to compute a value for each player, say m_i for P_i for $1 \le i \le t$. Each value m_i depends on x_i and a random binary string r_i, both of them are kept secret by player P_i for $1 \le i \le t$. Each player does not gain any information about $f(x_1, x_2, \cdots, x_t)$ and any information on other players' inputs. Then each player P_i sends secretly m_i to V, $1 \le i \le t$.

Proof sub-protocol: In this sub-protocol, the verifier V computes the value $f(x_1, x_2, \cdots, x_t)$ from $m_1, m_2, \cdots,$ m_t and verifies its validity.

When defining security of multi-party proof, we have to consider the security of each of the two sub-protocols. The computation sub-protocol is an ordinary multi-party computation which allows a set of mutually distrusting parties to compute a function in a distributed way while guaranteeing (to the extent possible) the privacy of their local inputs and the correctness of the outputs. To be more exact, security is typically formulated by comparing a real execution of the protocol to an ideal execution where the parties just send their inputs to a trusted party and receive back their outputs. A real protocol is said to be secure if an adversary can do no more harm in a real execution than in an ideal execution (which is secure by definition). The main security properties that have been identified, and are implied by this formulation, are privacy (parties learn nothing more than their own output) and correctness (the outputs are correctly computed).

In the proof sub-protocol, the verifier V will learn nothing beyond the value of $f(x_1, x_2, \cdots x_t)$ and its validity, that is, V only obtains $f(x_1, x_2, \cdots x_t)$ and its validity. In another word, V does not gain any information about $x_1, x_2, \cdots, x_t$ yet is convinced that the values of $x_1, x_2, \cdots, x_t$ used in computing $f(x_1, x_2, \cdots, x_t)$ are the same as claimed, and furthermore, the verifier obtains the correct function value $x_1, x_2, \cdots, x_t$. This is defined more formally as follows.

Definition 3 (Security of Multi-Party Proof)

A multi-party proof is secure if it satisfies the following properties:

1) **Correctness**: Suppose each player P_1 (for $i = 1, 2, \cdots, t$) chooses random binary string r_i of length n. Let y_0 be the final value computed by V from $m_1, m_2, \cdots, m_t$. Then y_0 is equal to $f(x_1, x_2, \cdots, x_t)$ with high probability, that is, the probability Pr($y_0 \neq f(x_1, x_2, \cdots, x_t)$) is negligible as a function of n.

2) **Privacy**: For each player P_i, $1 \le i \le t$, let M_i be all the message that P_i obtains from all other players in the computation sub-protocol. The protocol is said to have privacy for all the players if, for each $1 \le i \le t$, there exists a bounded probabilistic polynomial time simulator S_i such that M_i is indistinguishable from the output $S_i(1^n)$ of the simulator S_i.

For the verifier V, let $M_v = (m_1, m_2, \cdots, m_t)$ denote all the messages received from the players. We say that the protocol has privacy for V if there exists a bounded probabilistic polynomial time simulator S_v so that M_v is indistinguishable with the output $S_v(1^n)$ of the simulator S_v.

Also, none of the players learn anything about the function value computed.

3) Validity: The validity includes the two following properties:

a) V can verify with high probability that the value m_i is correctly computed from the claimed value of x_i, $1 \le i \le t$, all of which are kept secret from V.

b) V can verify the correctness of y_0. Assume the function f is given as a boolean circuit C. Then V can verify every gate's computation from the input wires to the output wires.

4. Construction of Secure Multi-Party Proof

In this section, we will construct a secure multi-party proof for any polynomial-time computable function $f(x_1, x_2, \cdots x_t)$ if all players $P_1, \cdots, P_n$ are semi-honest.

4.1 Two-Party Computation Secure against Semi-Honest Adversaries

We firstly describe the construction of secure two-party computation (for semi-honest adversaries) due to Yao [1]. We follow the description by [2,3] where it is proven to be secure against semi-honest adversaries.

Let C be a Boolean circuit that receives two inputs $x, y \in \{0,1\}^n$ and outputs $C(x, y) \in \{0,1\}^n$ (for simplicity, we assume that the input length, output length and the security parameter are all n). We also assume that C has the property that every gate with a circuit-output wire has no other outgoing wires. We begin by describing the construction of a single garbled gate g in C. The circuit C is Boolean, and therefore any gate is represented by a function $g : \{0,1\} \times \{0,1\} \to \{0,1\}$. Let the two input wires to g be labeled ω_1 and ω_2. g may have several outgoing wires, but all of them are labeled by the same symbol ω_3. Furthermore, let $k_1^0, k_1^1, k_2^0, k_2^1, k_3^0, k_3^1$ be six random keys obtained by independently invoking the keygeneration algorithm $G(1^n)$; for simplicity, assume that these keys are also of length n. Intuitively, we wish to be able to compute $k_3^{g(\alpha,\beta)}$ from k_1^α and k_2^β, without revealing any of the other three values, $k_3^{g(1-\alpha,\beta)}$, $k_3^{g(\alpha,1-\beta)}$, and $k_3^{g(1-\alpha,1-\beta)}$. The garbled gate g is defined by the following four values

$$c_{0,0} = E_{k_1^0}\left(E_{k_2^0}\left(g_3^{g(0,0)}\right)\right)$$

$$c_{0,1} = E_{k_1^0}\left(E_{k_2^1}\left(g_3^{g(0,1)}\right)\right)$$

$$c_{1,0} = E_{k_1^1}\left(E_{k_2^0}\left(g_3^{g(1,0)}\right)\right)$$

$$c_{1,1} = E_{k_1^1}\left(E_{k_2^1}\left(g_3^{g(1,1)}\right)\right)$$

where E is from a private key encryption scheme (G, E, D) that has indistinguishable encryptions for multiple messages, and has an elusive efficiently verifiable range [2,3]. The actual gate is defined by a random permutation of the above values, denoted as c_0, c_1, c_2, c_3; from here on we call them the garbled of gate g. Notice that given k_1^α and k_2^β, and these values c_0, c_1, c_2, c_3, it is possible to compute the output of the gate $k_3^{g(\alpha,\beta)}$ as follows. For every i, computes $D_{k_2^\beta}\left(D_{k_1^\alpha}(c_i)\right)$. If there are more than one decryption then returns a non-$\perp$ value, then output abort. Otherwise, define k_3^γ to be the only non-$\perp$ value that is obtained. (Notice that if only a single non-$\perp$ value is obtained, then this will be $k_3^{g(\alpha,\beta)}$ because it is encrypted under the given keys k_1^{α} and k_2^{β}. Later we will show that except with negligible probability, only one non-$\perp$ value is indeed obtained.)

We are now ready to show how to construct the entire garbled circuit. Let m be the number of wires in the circuit C, and let $\omega_1,\cdots,\omega_m$ be labels of these wires. These labels are all chosen uniquely with the following exception: if ω_i and ω_j are both output wires from the same gate g, then $\omega_i=\omega_j$ (this occurs if the fan-out of g is greater than one). Likewise, if an input bit enters more than one gate, then all circuit-input wires associated with this bit will have the same label. Next, for every label ω_i, choose two independent keys $k_i^0,k_i^1 \leftarrow G\left(1^n\right)$; we stress that all of these keys are chosen independently of the others. Now, given these keys, the four garbled values of each gate are computed as described above and the results are permuted randomly. Finally, the output or decryption tables of the garbled circuit are computed. These tables simply consist of the values $\left(0,k_i^0\right)$ and $\left(1,k_i^1\right)$ where ω_i is a circuit-output wire. (Alternatively, output gates can just compute 0 or 1 directly. That is, in an output gate, one can define $c_{\alpha,\beta}=E_{k_1^{\alpha}}$ $\left(E_{k_2^{\beta}}\left(g\left(\alpha,\beta\right)\right)\right)$ for every $\alpha,\beta\in\left\{0,1\right\}$).

The entire garbled circuit of C, denoted $G\left(C\right)$, consists of the garbled table for each gate and the output tables. We note that the structure of C is given, and the garbled version of C is simply defined by specifying the output tables and the garbled table that belongs to each gate. This completes the description of the garbled circuit.

Let $x=x_1,\cdots,x_n$ and $y=y_1,\cdots,y_n$ be two n-bit inputs for C. Furthermore, let $\omega_1,\cdots,\omega_n$ be the input labels corresponding to x, and let $\omega_{n+1},\cdots,\omega_{2n}$ be the input labels corresponding to y. It is shown in [2] that given the garbled circuit $G\left(C\right)$ and the strings $k_1^{x_1},\cdots,k_n^{x_n},k_{n+1}^{y_1},\cdots,k_{2n}^{y_n}$, it is possible to compute $C\left(x,y\right)$, except with negligible probability. The complete protocol is as followed.

Protocol 1 (Yao's two-party protocol):

1) **Inputs**: P_1 has $x\in\left\{0,1\right\}^n$ and P_2 has $y\in\left\{0,1\right\}^n$.

2) **Auxiliary input**: A boolean circuit C such that for every $x,y\in\left\{0,1\right\}^n$ it holds that $C\left(x,y\right)=f\left(x,y\right)$, where $f:\{0,1\}^n\times\{0,1\}^n\to\{0,1\}^n$. We require that C is such that every gate with a circuit-output wire has no other outgoing wires.

3) The protocol

a) P_1 constructs the garbled circuit $G\left(C\right)$ as described in above, and sends it to P_2.

b) Let $\omega_1,\cdots,\omega_n$ be the circuit-input wires corresponding to x, and let $\omega_{n+1},\cdots,\omega_{2n}$ be the circuit-input wires corresponding to y. Then,

i. P_1 sends P_2 the strings $k_1^{x_1},\cdots,k_n^{x_n}$.

ii. For every i,P_i and P_2 execute a 1-out-of-2 oblivious transfer protocol in which P_1 input equals $\left(k_{n+i}^0,k_{n+i}^1\right)$ and P_2's input equals y_i.

The above oblivious transfers can all be run in parallel.

c) Following the above, P_2 has obtained the garbled circuit and $2n$ keys corresponding to the $2n$ input wires to C. Party P_2 then computes using the garbled circuit, as described above, obtaining $f\left(x\right)$. P_2 then sends the value $f\left(x\right)$ to P_1, and they both output this value.

Assume that the oblivious transfer protocol is secure in the presence of static semi-honest adversaries, and that the encryption scheme has indistinguishable encryptions for multiple messages, and has an elusive and efficiently verifiable range. Then it is proved in [2] that Protocol 1 securely computes f in the presence of static semi-honest adversaries.

4.2 Secure Multi-Party Proof against Semi-Honest Adversaries

In this section, we will construct multi-party proof secure against semi-honest adversaries for any polynomial time computable function. We firstly construct a secure multi-party proof between two parties, then generalize it to a secure multi-party proof for more than two participants.

4.2.1 Secure Two-Party Proof against Semi-Honest Adversaries

Assume $f\left(x\right)$ is a polynomial computable function, so there exists a polynomial size boolean circuit C such that for every $x,y\in\{0,1\}^n$ it holds that $C\left(x,y\right)=f\left(x,y\right)$. Both $f\left(x,y\right)$ and C are public. We claim that the following protocol is a secure two-party proof for f.

Protocol 2:

1) **Input**: P_1 has $x\in\{0,1\}^n$ and P_2 has $y\in\{0,1\}^n$.

2) **Set-up**:

a) Let $\omega_1,\cdots,\omega_n$ be the circuit-input wires corresponding to x, and let $\omega_{n+1},\cdots,\omega_{2n}$ be the circuit-input wires corresponding to y. Let s be the number of gates in C excluding its circuit-input gates and circuit-output gates

b) By running key generating algorithm G for $4n+2s$ times, P_1 generates $4n+2s$ independent strings

$$k_1^0,k_1^1,\cdots,k_{2n}^0,k_{2n}^1,k_{2n+1}^0,k_{2n+1}^1,\cdots,k_{2n+s}^0,k_{2n+s}^1,$$

then constructs a garbled computation table for every gate by using the special symmetric encryption described in Section 2, and obtains a garbled circuit $G(C)$ which consists of the garbled table for each gate and the output tables. Finally, P_1 publicizes $G(C)$, and keeps k_1^0, $k_1^1,\cdots,k_{2n+s}^0,k_{2n+s}^1$ secret.

3) Computation Sub-protocol:

a) For every $1\le i\le n$, P_1 and P_2 execute a 1-out-of-2 oblivious transfer protocol in which P_1's input equals $\left(k_{n+i}^0,k_{n+i}^1\right)$ and P_2's input equals y_i.

b) The above oblivious transfers can be run by using the oblivious transfer protocol in Section 2 and can be done in parallel.

c) P_1 sends V the strings $k_1^{x_1},\cdots,k_n^{x_n}$.

d) P_2 sends V the strings $k_{n+1}^{y_1},\cdots,k_{2n}^{y_n}$.

4) Proof Sub-protocol:

V has obtained the garbled circuit $G(C)$ and $2n$ keys corresponding to the $2n$ input wires to C. V then computes and obtains $f(x,y)$ via $G(C)$.

Proof: **Correctness**. It is correct by the design of the garbled circuit.

Privacy. Assume that P_1 constructs the garbled circuit $G(C)$ for boolean circuit C. According to the oblivious transfer protocol in Section 2, all messages P_1 obtains are $\left((\omega_{11},\omega_{12}),(\omega_{21},\omega_{22}),\cdots,(\omega_{n1},\omega_{n2})\right)$ from n 1-out-of-2 oblivious transfer protocols between P_1 and P_2. In the transfer protocol, every ω_{ij} is chosen at random. The simulator M for P_1 is just a pseudorandom generator, see for example [12,16]. Let $M(1^n)$ denotes the output of M which has length $2nk\left(k=\left|\omega_{ij}\right|\right)$. Then $((\omega_{11},\omega_{12}),(\omega_{21},\omega_{22}),\cdots,(\omega_{n1},\omega_{n2}))$ and $M(1^n)$ is computational indistinguishable. That is, privacy for P_1 is satisfied.

Now we consider privacy for P_2. All messages P_2 obtains are $((E_1,b_{11},b_{12}),(E_2,b_{21},b_{22}),\cdots,(E_n,b_{n1},b_{n2}))$, where E_i is a permutation-trapdoor and pair (b_{i1},b_{i2}) is random for $i=1,2,\cdots,n$. Hence, privacy for P_2 is satisfied by using a pseudorandom generator as a simulator.

The privacy for both P_1 and P_2 implies that P_1 and P_2 do not obtain any information on the other player's input.

Next we consider privacy for V in computation sub-protocol. All messages V obtains are garbled circuit $G(C),k_1^{x_1},k_2^{x_2},\cdots,k_n^{x_n}$ from P_1, and $k_{n+1}^{y_1},k_{n+2}^{y_2},\cdots,k_{2n}^{y_n}$ from P_2. The symmetric encryption algorithm used in $G(C)$ is the special symmetric encryption scheme in Section 2. Because the output of the symmetric encryption algorithm is random and both $k_1^{x_1},k_2^{x_2},\cdots,k_n^{x_n}$ and $k_{n+1}^{y_1},k_{n+2}^{y_2},\cdots,k_{2n}^{y_n}$ are random, privacy for V is satisfied by using a pseudorandom generator as a simulator.

Finally, each player learns nothing about the function value computed by V, since the player know nothing about what other players send to the verifier.

Validity. Because all participants are semi-honest, the validity of all the value m_i computed by P_i from x_i holds automatically. To see the correctness of the value $f(x,y)$, note that V obtains $G(C)$ and $k_1^{x_1},k_2^{x_2},\cdots,k_n^{x_n}$ from P_1, and $k_{n+1}^{y_1},k_{n+2}^{y_2},\cdots,k_{2n}^{y_n}$ from P_2. Hence, according to properties of special symmetric encryption, he can compute $f(x,y)$ and verifies its validity by verifying every gate's computation from the circuit-input wires to circuit-output wires.

4.2.2 Secure Multi-Party Proof against Semi-Honest Adversaries

In this section, we will generalize two-party proof to multi-party proof. Assume $f(x_1,\cdots,x_t)$ is a polynomial time computable function so there exists a boolean circuit C of polynomial size such that, for every $x_1,\cdots,x_t\in\{0,1\}^n$, it holds that $C(x_1,\cdots,x_t)=f(x_1,\cdots,x_t)$. Both $f(x_1,\cdots,x_t)$ and C are made public.

Protocol 3:

1) **Inputs**: P_i has $x_i\in\{0,1\}^n$ for $i=1,2,\cdots,t.$

2) **Set-up**:

a) Let $\omega_{i1},\omega_{i2},\cdots,\omega_{in}$ be the circuit-input wires corresponding to x_i for $i=1,2,\cdots,t.$ Let s denote the number of gates in the circuit C excluding its circuit-input gates and circuit-output gates.

b) By running key generating algorithm G for $2tn+2s$ times, P_1 generates $2tn+2s$ independent strings $k_1^0,k_1^1,\cdots,k_{tn}^0,k_{tn}^1,\cdots,k_{tn+s}^0,k_{tn+s}^1$ constructs a garbled computation table for every gate by using special symmetric encryption in Section 2, and obtains a garbled circuit $G(C)$ which consists of the garbled table for each gate and the output tables. Finally, P_1 publicizes $G(C)$, and keeps $k_1^0,k_1^1,\cdots,k_{tn}^0,k_{tn}^1,\ \cdots,k_{tn+s}^0,k_{tn+s}^1$ secret.

3) **Computation Sub-protocol**:

a) For every $i(=2,\cdots,t)$, P_1 and P_i execute a 1-out-of-2 oblivious transfer protocol in which P_1's input equals $\left(k_{ij}^0,k_{ij}^1\right)$ and P_i's input equals x_{ij}, for $j=1,2,\cdots,n.$

b) The above oblivious transfers can be run by using the oblivious transfer protocol in Section 2 and can be done in parallel.

c) P_i sends V the strings $k_{i1}^{x_{i1}},\cdots,k_{in}^{x_{in}}$ for $i=1,2,\cdots,t$.

4) **Proof Sub-protocol**:

V has obtained the garbled circuit $G(C)$ and tn keys corresponding to the tn input wires to C. V computes and obtains $f(x_1.\cdots,x_t)$ via $G(C)$.

Proof: **Correctness**. It follows from the design of the protocol.

Privacy. Assume that P_1 constructs the garbled circuit $G(C)$ for boolean circuit C. Then, according to the oblivious transfer protocol in Section 2, all the messages P_1 sees are $(\left(\omega_{11}^i,\omega_{12}^i\right),\left(\omega_{21}^i,\omega_{22}^i\right),\ \left(\omega_{n1}^i,\omega_{n2}^i\right))$ from n 1-out-of-2 oblivious transfer protocols between P_1 and $P_i(i=2,\cdots,t)$. By the design of the oblivious transfer protocol, every $\omega_{jk}^i\ (i=2,\cdots,t,\ j=1,2,\cdots,n)$ is random. The simulator M for P_1 is just a pseudorandom generator, say the one constructed in [12,16]. Let $M(1^n)$ denote the output of M, whose output length is $2(t-1)nk\left(k=\left|\omega_{ij}^i\right|\right)$.

Then $\left\{\left(\left(\omega_{11}^2,\omega_{12}^2\right),\left(\omega_{21}^2,\omega_{22}^2\right),\cdots,\left(\omega_{n1}^2,\omega_{n2}^2\right)\right),\ \cdots,\ \left(\left(\omega_{11}^t,\omega_{12}^t\right),\left(\omega_{21}^t,\omega_{22}^t\right),\cdots,\left(\omega_{n1}^t,\omega_{n2}^t\right)\right)\right\}$ and $M(1^n)$ is computational indistinguishable. That is, privacy for P_1 is satisfied.

We consider privacy for every $P_i(i=2,\cdots,t)$. All messages P_i obtains are $\left(\left(E_1^i,b_{11}^i,b_{12}^i\right),\ \left(E_2^i,b_{21}^i,b_{22}^i\right),\cdots,\ \left(E_n^i,b_{n1}^i,b_{n2}^i\right)\right)$, where each E_j^i is a permutation-trapdoor and the pair $\left(b_{j1}^i,b_{j2}^i\right)$ is random for $j=1,2,\cdots,n$. Hence, privacy for P_i is satisfied by using a pseudorandom generator as a simulator.

The privacy for all $P_1,P_2,\cdots,P_t$ implies that every P_i obtains no information on other players input during the computation sub-protocol.

Now we consider privacy for V in computation sub-protocol. All messages V obtains are garbled circuit $G(C)$ and $k_{11}^{x_{11}},k_{12}^{x_{12}},\cdots,k_{1n}^{x_{1n}}$ from P_1, and $k_{i1}^{x_{i1}},k_{i2}^{x_{i2}},\cdots,k_{in}^{x_{in}}$ from P_i, $i=2,\cdots,t$. The symmetric encryption algorithm used in $G(C)$ is the special symmetric encryption scheme in Section 2. Because output of the symmetric encryption algorithm is random and every $k_{ij}^{x_{ij}}\ (i=1,2,\cdots,t,\ j=1,2,\cdots,n)$ is random, privacy for V is satisfied by using a pseudorandom generator as a simulator.

Finally, each player learns nothing about the function value computed by V, since the player know nothing about what other players send to the verifier.

Validity. Because all participants are semi-honest, part (a) of the validity holds automatically. For part (b) of the validity, V obtains $G(C)$ and $k_{11}^{x_{11}},k_{12}^{x_{12}},\cdots,k_{1n}^{x_{1n}}$ from P_1, and gains $k_{i1}^{x_{i1}},k_{i2}^{x_{i2}},\cdots,k_{in}^{x_{in}}$ from P_i, $i=2,\cdots,t$. Hence, according to the properties of the special symmetric encryption, V can compute $f(x_1,x_2,\cdots,x_t)$ and verifies its validity by verifying every gate's computation from the circuit-input wires to circuit-output wires.

5. Electronic Protocols Based on Secure Multi-Party Proof

Protocol 3 can be used in many applications where the verifier has no influence on the functions to be computed, yet he/she wants to learn the function values, however, Protocols 1 and 2 cannot be used because there are only two participants. The arbitrator is assured of the correctness and validity of the function value computed. Two important applications we have in mind are electronic voting and electronic bidding.

Assume $f(x_1,\cdots,x_t)$ is a polynomial time computable function used in an application.

For electronic voting, the function is a sum: $f(x_1,x_2,\cdots,x_t)=x_1+x_2+\cdots+x_t$ the total count for votes of a candidate. For electronic bidding, we may have $f\left(x_1, x_2, \cdots, x_t\right) = \max\{x_1, x_2, \cdots, x_t\}$ where x_i is the bid from P_i, $1 \le i \le t$. For these functions there is polynomial sized circuit C for computing them. Both the function $f\left(x_1, \cdots, x_t\right)$ and its circuit C are public information. Then our Protocol 3 above can be applied to both of these cases, and it is secure if the participants and the verifier are semi-honest.

6. Conclusions

We defined a new type cryptographical model called secure multi-party proof that allows any t players and a verifier to securely compute a function with t variables. We presented a protocol that is secure when all the participants and the verifier are semi-honest. Our protocol can be used for electronic voting and electronic bidding.

The main difference between multi-party computation and multi-party proof is that, in the former case only participants know the output or partial output, while in the later case only a designated verifier learns the final output. The latter is more practicable in some important situations, for example, in an electronic bidding, where an arbiter is only a verifier but not a participant.

It should be noted, however, that our protocol is not secure against malicious adversaries.

Based on non-interactive zero-knowledge proof, it is possible to construct secure multi-party proof against malicious adversaries. However, these protocols are inefficient because of complexity of zero-knowledge proof. In future work, we intend to construct secure multi-party proof for any polynomial-time computable function $f\left(x_1, x_2, \cdots, x_t\right)$ from tools other than zero-knowledge proof.

7. Acknowledgements

This work was supported by Foundation of National Natural Science (China) (10871222) and Opening Foundation of Key Lab of Cryptological Technology and Information Security Ministry of Education in Shandong University.

REFERENCES

[1] A. Yao, “Protocols for Secure Computation,” *Proceedings of the* 23*rd Annual Symposium on Foundations of Computer Science*, Chicago, November 1982, pp. 160-164.

[2] Y. Lindell and B. Pinkas, “A Proof of Yao’s Protocol for Secure Two-Party Computation,” *Journal of Cryptology*, Vol. 22, No. 2, April 2009, pp. 161-188.

[3] Y. Lindell and B. Pinkas, “An Efficient Protocol for Secure Two-Party Computation in the Presence of Malicious Adversaries,” *Advances in Cryptology-Eurocrypt* 2007, LNCS 4515, Barcelona, May 2007, pp. 52-78.

[4] O. Goldreich, S. Micali and A. Wigderson, “How to Play Any Mental Game—A Completeness Theorem for Protocols with Honest Majority,” *Proceedings of the* 19*th Annual ACM Symposium on Theory of Computing*, New York, 1987, pp. 218-229.

[5] O. Goldreich, “Foundations of Cryptography: Volumne 2—Basic Applications,” Cambridge University Press, 2004.

[6] M. Ben-Or, S. Goldwasser and A. Wigderson, “Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation,” *Proceedings of the* 20*th Annual ACM Symposium on Theory of Computing*, Chicago, 1988, pp. 1-10.

[7] D. Chaum, C. Crepeau and I. Damgard, “Multiparty Unconditionally Secure Protocols,” *Proceedings of the* 20*th Annual ACM Symposium on Theory of Computing*, Chicago, 1988, pp. 11-19.

[8] S. Goldwasser and L. Levin, “Fair Computation of General Functions in Presence of Immoral Majority,” *Advances of Cryptology-Crypto*’90, LNCS 537, Santa Barbara, California, August 1990, pp. 77-93.

[9] B. Chor and E. Kushilevitz, “A Zero-One Law for Boolean Privacy,” *SIAM Journal on Discrete Mathematics*, Vol. 4, No. 1, February 1991, pp. 36-47.

[10] O. Goldreich, S. Goldwasser and N. Linial, “Fault-Tolerant Computation in the Full Information Model,” *SIAM Journal on Computing*, Vol. 27, No. 2, 1991, pp. 506-544.

[11] R. Ostrovsky and M. Yung, “How to Withstand Mobile Virus Attacks,” *Proceedings of the* 10*th Annual ACM Symposium on Principles of Distributed Computing*, Montreal, August 1991, pp. 51-59.

[12] S. Micali and P. Rogaway, “Secure Computation,” *Advances of Cryptology-Crypto*’91, LNCS 576, Santa Barbara, California, August 1991, pp. 392-404.

[13] D. Beaver, “Foundations of Secure Interactive Computing,” *Advances of Cryptology-Crypto*’91, LNCS 576, Santa Barbara, California, August 1991, pp. 377-391.

[14] R. Canetti, U. Feige, O. Goldreich and M. Naor, “Adaptively Secure Multi-Party Computation,” *Proceedings of the* 28*th Annual ACM Symposium on Theory of Computing*, Philadelphia, 1996, pp. 639-648.

[15] J. A. Garay and R. Ostrovsky, “Almost-Everywhere Secure Computation,” *Advances of Cryptology-Eurocrypt* 2008, LNCS 4965, Istanbul, April 2008, pp. 307-323.

[16] R. Pass, “Bounded-Concurrent Secure Multi-Party Computation with a Dishonest Majority,” *Proceedings of the* 36*th Annual ACM Symposium on Theory of Computing*, Chicago, 2004, pp. 232-241.

[17] C. Cachin and J. Camenisch, “Optimistic Fair Secure Computation,” *Advances of Cryptology-Crypto*’00, LNCS 1880, Santa Barbara, California, August 2000, pp. 93-111.

[18] S. D. Gordon, C. Hazay, J. Katz and Y. Lindell, “Complete Fairness in Secure Two-Party Computation,” *Proceedings of the* 40*th Annual ACM Symposium on Theory*

of Computing, Victoria, 2008, pp. 413-422.

[19] B. Pinkas, "Fair Secure Two-Party Computation," *Advance in Cryptology-Eurocrypt* 2003, LNCS 2656, Warsaw, May 2003, pp. 87-106.

[20] S. Even, O. Goldreich and A. Lempel, "A Randomized Protocol for Signing Contracts," *Communications of the ACM*, Vol. 28, No. 6, 1985, pp. 637-647.

[21] O. Goldreich, S. Goldwasser and S. Micali, "How to Construct Random Functions," *Journal of the ACM*, Vol. 33, No. 4, 1986, pp. 792-807.

[22] O. Goldreich, H. Krawcyzk and M. Luby, "On the Existence of Pseudorandom Generators," *SIAM Journal on Computing*, Vol. 22, No. 6, 1993, pp. 1163-1175.

[23] J. Hastad, R. Impagliazzo, L. A. Levin and M. Luby, "Construction of a Psedorandom Generator from Any One-Way Function," *SIAM Journal on Computing*, Vol. 28, No. 4, 1999, pp. 1364-1396.

J. Software Engineering & Applications, **2010, 3, 718-722**
doi:10.4236/jsea.2010.37082 Published Online July 2010 (http://www.SciRP.org/journal/jsea)

CRAB—CombinatoRial Auction Body Software System

Petr Fiala[1], Jana Kalčevová[1], Jan Vraný[2]

[1]Department of Econometrics, Faculty of Informatics and Statistics, University of Economics, Prague, Czech Republic; [2]Software Engineering Group, Faculty of Information Technology, Czech Technical University, Prague, Czech Republic.
Email: pfiala@vse.cz, kalcevov@vse.cz, jan.vrany@fit.cvut.cz

Received February 4th, 2010; revised June 17th, 2010; accepted June 19th, 2010.

ABSTRACT

Auctions are important market mechanisms for the allocation of goods and services. Combinatorial auctions are those auctions in which buyers can place bids on combinations of items. Combinatorial auctions have many applications. The paper presents the CRAB software system. CRAB is a non-commercial software system for generating, solving, and testing of combinatorial auction problems. The system solves problems by Balas' method or by the primal-dual algorithm. CRAB is implemented in Ruby and it is distributed as the file crab.rb. The system is freely available on web pages for all interested users.

Keywords: *Combinatorial Auction, Complexity, Software System, Generating, Solving, Testing*

1. Introduction

Auctions are important market mechanisms for the allocation of goods and services. Auction theory has caught tremendous interest from both the economic side as well as the Internet industry. Design of auctions is a multidisciplinary effort made of contributions from economics, operations research, software sciences, and other disciplines. The popularity of auctions and the requirements of e-business have led to growing interest in the development of complex trading models [1]. Combinatorial auctions are those auctions in which buyers can place bids on combinations of items, called bundles. This is particular important when items are complements.

CRAB system is suitable for:

- fast generating of standard combinatorial restrictions,
- adding specific restrictions,
- generating of bundle evaluations with complementarity properties,
- solving combinatorial auction problems by Balas' method or by the primal-dual algorithm,
- experimenting, and testing.

The user has to install Ruby interpreter in order to run CRAB. Needed information is at web pages http://users.fit.cvut.cz/~vranyj1/software/crab. There is also a possibility to download CRAB (more precisely the file crab.rb).

This design gives a possibility to study and extend the CRAB system, especially about the implemented models and methods, by the user. The system is available on web pages for all interested users.

2. Combinatorial Auctions

Combinatorial auctions are those auctions in which buyers can place bids on combinations of items, so called bundles. The advantage of combinatorial auctions is that the buyer can more precisely express his preferences. This is particular important when items are complements. The auction designer also derives value from combinatorial auctions. Allowing buyers more fully to express preferences often leads to improved economic efficiency and greater auction revenues. However, alongside their advantages, combinatorial auctions raise a host of questions and challenges [2]. Many types of combinatorial auctions can be formulated as mathematical programming problems.

The problem, called the winner determination problem, has received considerable attention in the literature. The problem is formulated as: Given a set of bids in a combinatorial auction, find an allocation of items to buyers that maximize the seller's revenue. Let us suppose that one seller offers a set M of m items, j = 1, 2,…, m, to n potential buyers. Items are available in single units. A bid made by buyer i, i = 1, 2, …, n, is defined as

$$B_i = \{S, v_i(S)\},$$

$S \subseteq M$, is a combination of items, so called bundle.

$v_i(S)$, is the valuation or offered price by buyer i for the bundle S.

The objective is to maximize the revenue of the seller given the bids made by buyers. Constraints establish that no single item is allocated to more than one buyer and that no buyer obtains more than one combination. This constraint is not necessary but there is a possibility to generate it in the CRAB system. Bivalent variables are introduced for model formulation:

$x_i(S)$ is a bivalent variable specifying if the combination S is assigned to buyer i ($x_i(S) = 1$).

The winner determination problem can be formulated as follows

$$\sum_{i=1}^{n} \sum_{S \subseteq M} v_i(S)\, x_i(S) \to \max$$

$$\text{subject to } \sum_{S \subseteq M} x_i(S) \le 1,\ \forall\ i = 1, 2, \dots, n,$$

$$\sum_{i=1}^{n} \sum_{S \subseteq M} x_i(S) \le 1,\ \forall\ j \in M, \tag{1}$$

$$x_i(S) \ge 0, \forall\ S \subseteq M,\ \forall\ i = 1, 2, \dots, n.$$

The objective function expresses the revenue. The first constraint ensures that no buyer receives more than one combination of items. The second constraint ensures that overlapping sets of items are never assigned. In the CRAB system a generalization is applied. The set of items M can be divided in p subsets P_k, k = 1, 2, …, p, called packages. All combinations of items are generated in each package. By this way all bundles are generated.

The algorithms proposed for solving combinatorial auction problems fall into two classes: exact algorithms, and approximate algorithms. The CRAB system uses Balas' method for finding of optimal solutions.

Alternatively, the primal-dual algorithm can be taken as a decentralized and dynamic method to determine the pricing equilibrium. A primal-dual algorithm usually maintains a feasible dual solution and tries to compute a primal solution that is both feasible and satisfies the complementary slackness conditions. If such a solution is found, the algorithm terminates. Otherwise the dual solution is updated towards optimality and the algorithm continues with the next iteration. The fundamental work [3] demonstrates a strong interrelationship between the iterative auctions and the primal-dual linear programming algorithms. A primal-dual linear programming algorithm can be interpreted as an auction where the dual variables represent item prices. The algorithm maintains a feasible allocation and a price set, and it terminates as the efficient allocation and competitive equilibrium prices are found.

For the winner determination problem we will formulate the LP relaxation and its dual. Consider the LP relaxation of the winner determination problem (1):

$$\sum_{i=1}^{n} \sum_{S \subseteq M} v_i(S)\, x_i(S) \to \max$$

$$\text{subject to } \sum_{S \subseteq M} x_i(S) \le 1,\ \forall\ i = 1, 2, \dots, n,$$

$$\sum_{i=1}^{n} \sum_{S \subseteq M} x_i(S) \le 1,\ \forall\ j \in M, \tag{2}$$

$$x_i(S) \in \{0,1\}, \forall\ S \subseteq M,\ \forall\ i = 1, 2, \dots, n.$$

The corresponding dual to problem (2)

$$\sum_{i=1}^{n} p(i) + \sum_{j \in S} p(j) \to \min$$

subject to

$$p(i) + \sum_{j \in S} p(j) \ge v_i(S), \forall\, i, S,$$

$$p(i), p(j) \ge 0, \forall\, i, j. \tag{3}$$

The dual variables $p(j)$ can be interpreted as anonymous linear prices of items, the term

$$\sum_{j \in S} p(j)$$

is then the price of the bundle S and

$$p(i) = \max_{S} \left[v_i(S) - \sum_{j \in S} p(j) \right]$$

is the maximal utility for the bidder i at the prices $p(j)$.

3. CRAB Tool

During the research on combinatorial auctions a need for input problem generator arose. There is a publicly available tool CATS [4], developed at Stanford University, however it has several drawbacks that make it, at least, hardly usable for our purpose. To fulfill our needs we have developed our own tool: CRAB. This tool has several advantages over the CATS, namely:

- fast problem generation,
- combinations are generated in a more predictable way,
- combinations are generated only in given subset of all items,
- CSV as the primary data format,
- fine-grained control over generated problem,
- built-in linear problem solver,
- multiple output formats.

This tool is implemented in Ruby. Although obvious programming language of choice for such kind of tools is C or C++ for performance reasons, we choose Ruby mainly for its dynamic, agile nature which enables us to quickly experiment with different approaches.

3.1 Overview

The combinatorial auction problem is given by the number of buyers and the number of all feasible combinations of goods—bundles. For each buyer also prices of bundles, bids and budget are needed. Number of goods is read in the vector form where the number of vector components (comma separated) is equal to the number of packages. Each vector component corresponds to the number of goods in the package.

In the first phase there are generated all combinations of goods in each package (except empty set). This step is done for every package. By this way all bundles are generated. The list of these bundles is saved in the file (*.csv)—one bundle on row—and there are prepared one column for each buyer. The first row contains column label and the second row is given for buyer's budget.

The user can load the file in CSV into a text editor or in a spreadsheet and fill in the bids (*i.e*. the price offered by the buyer for particular bundle) and budgets for each buyer. If the user uses CRAB only for tests he/she can use automatically generated prices and budget. In both cases the final file has to be saved also in CSV format.

In the second phase the file is transformed into the binary programming problem. The bundles correspond to variables and bids correspond to prices of objective function that is maximized. The problem consists of automatic constraints for each good (each good can be sold only once) and each buyer (buyer cannot get over his budget). The user is free to change automatically generated constraints and remove or add additional (for example not-typical) constraints. All data have to be saved in CVS format again.

Finally, the problem might be passes to the built-in binary programming solver to find out the optimal solution for given combinatorial auction. If so, the problem is transformed into form with minimizing objecttive function with non-negative prices and all constraints in form "less or equal". More detailed description of normalization process is in Sub-section 3.6. Afterwards the transformed model is passed to the Balas' algorithm [5].

3.2 Practical Usage

The CRAB tool as well as the source code could be downloaded from the above mentioned web page and it can operate in two modes: 1) interactive mode and 2) command line mode.

Interactive mode

CRAB could be run in interactive mode which means that the user is prompted for the data interactively. To start CRAB in interactive mode, type[1]

```
ruby crab.rb
```

in terminal emulator (or *command line* window on Windows) from within the directory where the program is installed. Please note that not all features are available when running in interactive mode, if you need them, you have to run it in *command line mode* described below.

Command line mode

As opposite to interactive mode, the CRAB could operate in *command line mode*, which means that all input data as well as all options have to be supplied on the command line. This mode allows one to automate tasks using scripts (and run CRAB non-interactively in night batch jobs, for instance). To get list of all available options, type

```
ruby crab.rb --help
```

All examples in following sections will be given in the command line mode.

3.3 Generating Bundles

As we said before, the CRAB can generate combinatorial problem as specified in Section 2. The principle command is:

```
ruby crab.rb --output <outfile> generate --buyers <nb> --bundles <bundlespec>
```

where <nb> is number of buyers (non-negative integer value) and <bundlespec> is vector specifying number of items in each bundle. Number of bundles is determined by dimension of the vector. Vector should be entered as comma-separated sequence of positive integer values with no spaces in between. CRAB can also generate random prices for each package as well as budget for each buyer.

Following command will generate combinatorial auction with four buyers and two packages first with 5 goods, second one with 3 goods and generate random prices and budgets. Output will be saved to file bids.csv:

```
ruby crab.rb --output bids.csv generate --buyers 4 --goods 5,3 --randomize
```

Every single good in generated combinatorial auction get assigned a unique integer id, starting at one. In previous example, the first good of first package gets id 1, second good in first package gets 2. The first good of second package gets 6, second one gets 7 and so on.

3.4 Output Format

The generated files are ordinary CSV[2] files and thus editable by almost every spreadsheet application. First row contains only column labels and contains no data. Second row contains budget of each buyer. Rest of rows specifies bids of bundles given by each buyer.

First two columns contain only labels and no data. First column denotes package, second one denotes particular good combination within the bundle. Bundle is denoted by id of first and last good in the bundle, the combination is

[1] Assuming the directory with ruby program is listed in the PATH environment variable. If not, fully path to the Ruby interpreter must be given, such as C:\Ruby\bin\ruby.

[2] Comma Separated Values, RFC 4180

denoted by minus-separated list of good ids. Following columns contains the bids given by buyers.

For output from the previous command see **Figure 1**.

User is free to modify the generated file to meet his/her needs; however the format of the file as described above must be preserved.

3.5 Transforming Combinatorial Auction to Binary Programming Problem

Once the combinatorial auction is generated, it can be transformed to the form of binary programming problem and passed to binary programming solver afterwards. The principle command for combinatorial auction transformation is:

```
./ruby crab.rb --output <output file> transform --bids <input file>
```

where <input file> is CSV file specifying the combinatorial auction. The form of input file must be the same as output of generate command. Optionally, you may use the --format option to specify output format. CRAB currently supports two output formats: 1) csv (which is the default) and 2) xa. The first one is the one used by built-in solver; the latter one could be directly passed to the XA integer solver [6].

Following command will transform file bids.csv into binary programming problem, saving the output to file named problem.cvs using the csv format.

```
./ruby crab.rb --output problem.cvs transform --bids bids.csv
```

Following command will create binary programming problem specification file as used by the XA solver:

```
./ruby crab.rb --output problem.lp transform --format xa --bids bids.csv
```

```
   Group,Bid,Offer  of  b1,Offer  of
b2,Offer of b3,Offer of b4
     "",Budget,8570,5879,6500,9065
     1-5,1,9,5,6,2
     1-5,2,7,4,8,3
     1-5,3,9,6,7,3
     1-5,4,6,4,4,7
     1-5,5,9,6,4,8
     1-5,1-2,8,7,11,13
     1-5,1-3,12,9,9,7
     1-5,1-4,9,14,10,12
     ...
     ...
     ...
     6-8,8,8,2,3,5
     6-8,6-7,10,12,10,8
     6-8,6-8,13,7,12,10
     6-8,7-8,11,10,8,9
     6-8,6-7-8,17,19,19,15
```

Figure 1. Example of output

Output format

As we mention above, the transform command can generate output files in two formats. In-depth description of XA format is beyond the scope of this paper. The CRAB format is ordinary CSV file. The first row and first column contains only column resp. row labels and no meaningful data. Remaining but last rows specifies constraints, last column is constraint's right hand side, pre-last column denotes relation. The last row contains objective function.

3.6 Solving

CRAB contains built-in binary programming solver based on Balas' method. The principle command for solving combinatorial auction is:

```
ruby carb.rb solve --problem <input file>
```

where <input file> is CSV file specifying the binary programming problem. The form of input file must be the same as output of transform command using the csv format.

The CRAB tool also provides few options to control the Balas' algorithm. The first option controls overall strategy to walk through the state space. Two strategies are available: depth-first (specified by --depth-first option) and breadth-first (which is the default, specified by --breadth-first option).

Second option deals with branching logic. If --one-first is specified then the one-filled branch is tried first, if --zero-first is specified, zero-filled branch is taken first. *One-first* strategy is the default one. Based on few experiments breadth-first strategy combined with one-first strategy gives the best results (by means of number of iterations required to solve particular problem).

Problem Normalization

As was written above, a binary programming problem with zero-one variables can be solved by Balas' method. Before using Balas' algorithm the problem has to have the correct form. In the CRAB system the problem has to satisfy following conditions:

- all constraints have to be in the form "less or equal",
- objective function has to be minimized and
- prices in the objective function have to be non-negative.

In the case that constraint is in the equation form CRAB automatically transforms it into two constraints—one in the form "less or equal" and the other "greater or equal". In the case that constraint is in the form "grater or equal" CRAB multiplies it by minus one. After such transformation all constraints are in the asked form.

If objective function is maximized it is multiplied by minus one and extreme is changed from max to min.

The last problem can be negative prices. As the problem is binomial (all variables are only one or zero) it is possible to use binomial substitution where the new variable equals 1—old variable. It is clear that the new variable is also binomial. After such substitution in whole model the price for new variable is positive. So it is necessary to use such substitution for all variables with negative prices in objective function.

Obviously the problem normalization is implemented in CRAB. After solving the results is automatically transformed by backward-substitution and user obtains the solution of original problem.

Output format

For output of the built-in solver see **Figure 2.**

As the solver is solving the problem, it prints some statistical information: total number partial solution in the queue, delta from last output and values of few other internal variables. After the solver finishes the computation, it prints out number of iterations done, the solution and the value of objective function.

```
   Iteration  1000:  47  solutions  in
queue (delta 46)
    z_f = 25930.0, z = 32702, z_max =
26433.0]
   Iteration  2000:  123  solutions  in
queue (delta 76)
    z_f = 25930.0, z = 32702, z_max =
26807.0]
   Iteration  3000:  269  solutions  in
queue (delta 146)
    z_f = 25991.0, z = 32702, z_max =
26812.0]
   Iteration  4000:  321  solutions  in
queue (delta 52)
    z_f = 27285.0, z = 32702, z_max =
28251.0]
   Iterations done: 4214
   Solution: Vector[1, 1, 1, 1, 1, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0], Z = 5203
```

Figure 2. Output of the solver

4. Conclusions

Combinatorial auctions are those auctions in which buyers can place bids on combinations of items, called bundles. This is particular important when items are complements. Combinatorial auctions have many real applications. The proposed CRAB system is a non-commercial software system for generating, solving, and testing of combinatorial auction problems. The tool has several advantages; as fast problem generations, combinations are generated only in given subset of all items, CSV as the primary data format, built-in problem solver, to name some of them.

5. Acknowledgements

The research project was supported by Grants No. 402/ 07/0166 and No. P402/10/0197 from the Grant Agency of Czech Republic.

REFERENCES

[1] M. Bellosta, I. Brigui, S. Kornman and D. Vanderpooten, “A Multi-Criteria Model for Electronic Auctions,” *ACM Symposium on Applied Computing*, 2004, pp. 759-765.

[2] P. Cramton, Y. Shoham and R. Steinberg, (Eds.) “Combinatorial Auctions,” MIT Press, Cambridge, 2006.

[3] S. Bikhchandani and J. M. Ostroy, “The Package Assignment Model,” *Journal of Economic Theory*, Vol. 107, No. 2, 2002, pp. 377-406.

[4] K. Leyton-Brown, M. Pearson and Y. Shoham, “Towards a Universal Test Suite for Combinatorial Auction Algorithms,” *Proceedings of ACM Conference on Electronic Commerce*, Minneapolis, 2000, pp. 448-457.

[5] E. Balas, “An Additive Algorithm for Solving Linear Programs with Zero-one Variables,” *Operations Research*, Vol. 13, No. 4, 1965, pp. 517-546.

[6] “XA Linear Optimizer System,” 2003. http://www.Sunsetsoft.com/

J. Software Engineering & Applications, **2010, 3, 723-727**
doi:10.4236/jsea.2010.37083 Published Online July 2010 (http://www.SciRP.org/journal/jsea)

Modeling and Analysis of Submerged Arc Weld Power Supply Based on Double Closed-Loop Control

Baoshan Shi[1], Kuanfang He[2], Xuejun Li[2], Dongmin Xiao[3]

[1]School of Mechanical and Vehicle Engineering, Beijing Institute of Technology, Zhuhai, China; [2]Hunan Provincial Key Laboratory of Health Maintenance for Mechanical Equipment, Xiantan, China; [3]College of Electromechanical Engineering, Xiantan, China.
Email: hkf791113@163.com

Received January 6th, 2010; revised May 9th, 2010; accepted May 11th, 2010.

ABSTRACT

According to the soft-switching pulsed SAW (Submerged arc weld) weld power supply based on the double closed-loop constant current control mode, a small signal mathematic model of main circuit of soft-switching SAW inverter was established by applying the method of three-terminal switching device modeling method, and the mathematic model of double closed-loop phase-shift control system circuit was established by applying the method of state-space averaging method. Dynamic performance of the inverter was analyzed on base of the established mathematic model, and the tested wave of dynamic performance was shown by experimentation. Research and experimentation show that relation between structure of the power source circuit and dynamic performance of the controlling system can be announced by the established mathematic model, which provides development of power supply and optimized design of controlling parameter with theoretical guidance.

Keywords**:** *SAW, Double Loop Control, Soft-Switching, Inverter, Mathematic Model*

1. Introduction

The full-bridge phase-shift zero-voltage soft-switching PWM inverter now is widely used in the weld field for its many excellent performances. Through establishing mathematic model and transfer function of soft-switching pulsed metal active gas welding power supply, the relation between structural parameters of circuit and dynamic performance of system is obtained, which is an effective method of designing and development of that power supply [1,2]. In the field of power electronics, problem of linear PWM DC-DC converter modeling was solved, there are many methods of modeling such as three-terminal switching device modeling method, data-sampling, symbol analysis and so on [3-6], and method of space state average applied to inverter modeling [7-10], which provide mathematic model of soft-switching pulsed metal active gas welding power supply with theoretical guidance.

This paper proposes a soft-switching SAW weld power supply based on the double closed-loop constant current control mode, which adopts structure of soft-switching full-bridge circuit and combines the conventional negative feedback of current or voltage and the peak current control mode. A small signal mathematic model of main circuit of soft-switching SAW inverter and the mathematic model of double loop control circuit are established by applying the method of three-terminal switching device modeling method and the method of space state average. According to mathematic model, dynamic performance of the inverter is analyzed, and tested wave of dynamic performance is shown to prove the rationality of the inverter by experimentation.

2. Principles

The sketch map of the double closed-loop feedback control system is shown in figure1. It uses hall sensor to sample current signal from primary transformer, and pouring into control loop after sophisticated high-speed rectifying. The control loop needs a reasonable slope compensation circuit to ensure the system to be stable and get appropriate open-loop frequency.

In the course of operation, the peak current signal $i_s(t)$ is sampled from the peak current of the VT, then plus a peak current slope compensation signal $i_a(t)R_{f1}$, which is a signal substituted traditional triangular wave signal in voltage mode control. The saw tooth signal $i_a(t)R_{f1}$ is synchronized with the signal of inverter cycle, which is mainly used to improve waveform of the peak current signal, reduce the noises from the power circuit, and advance system stability. Meanwhile, average output current of inductance $i_L(t)R_{f2}$ is detected, which is compared to the given signal to get error signal. The error signal is replaced by the given signal of voltage mode control after correcting or compensating, and incises the peak current $i_s(t)$ that adds saw tooth signal $i_a(t)R_{f1}$ to adjust duty cycle of VT, which realizes effective control of the output current.

The main advantages of inner loop control is to improve the overall dynamic response speed of system, protect power tube and realize correction of each current pulse, solve problem of magnetic bias of power transformer; the purpose of outer loop control is to improve control accuracy and technology of power.

3. Modeling and Analysis

3.1 Mathematical Model of Main Circuit

In this paper, mathematical model of main circuit of SAW soft switch inverter is established by the way of three-terminal switching device modeling method. The soft-switching circuit is full-bridge circuit in **Figure 1**; it is still a typical Buck Converter in essence [11,12].

The main circuit of soft-switching inverter equals to circuit According to three-terminal switching device modeling method in **Figure 2(a)**, dynamic low-frequency small signal circuit model in the pluralism domain is shown in **Figure 2(b)**.

According to **Figure 2(b)**, dynamic equations of AC small signal $\hat{i}_L(s)$ of inductance current in pluralism domain are expressed as Equations (1) and (2):

$$\hat{i}_L(s)=\frac{\left(\hat{u}_i(s)+\frac{U_i}{D}\hat{d}(s)\right)D}{Z_i(s)}=\frac{U_i}{Z_i(s)}\left(\frac{D}{U_i}\hat{u}_i(s)+\hat{d}(s)\right) \quad (1)$$

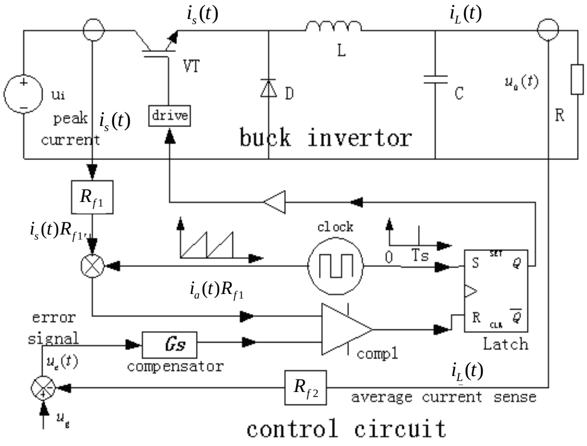
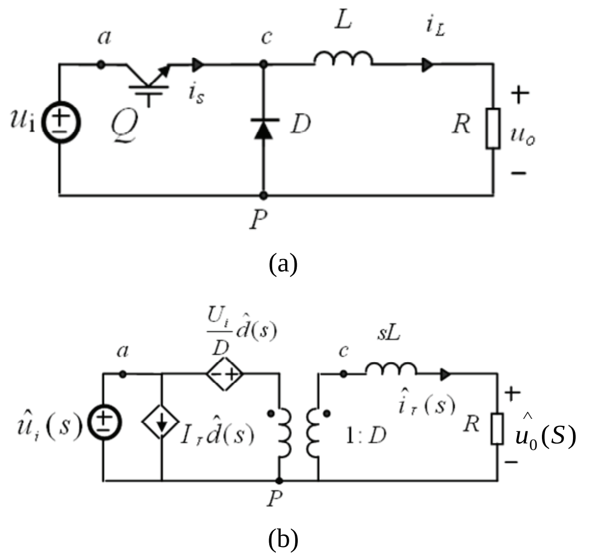


Figure 1. Block diagram of the control system



(a)

(b)

Figure 2. The AC signal model of main circuit

$$Z_i(s)=sL+R \quad (2)$$

Equations (3) and (4) are dynamic equations of small signal of output voltage in pluralism domain.

$$\hat{u}_o(s)=\hat{i}_L(s)Z_o(s) \quad (3)$$

$$Z_o(s)=R \quad (4)$$

From above equations, the transfer function for relation between output current of load and duty cycle can be denoted as Equation (5):

$$G_{id}(s)=\frac{\hat{i}_L(s)}{\hat{d}(s)}\Big|_{\hat{u}_i(s)=0}=\frac{U_i}{sL+R}=\frac{U_i}{R}\frac{1}{\frac{L}{R}s+1} \quad (5)$$

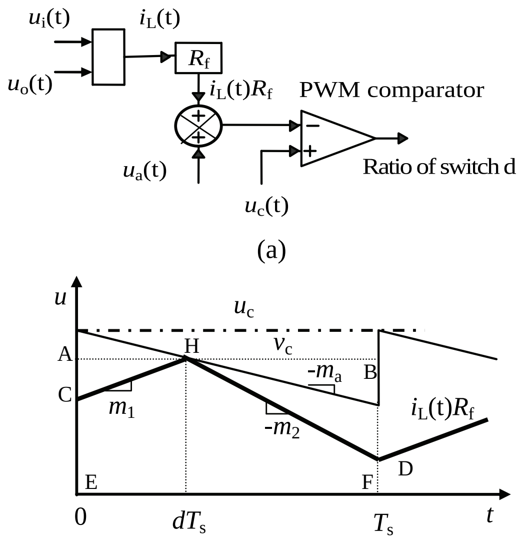
The transfer function for relation between output current of load and duty cycle can be denoted as Equation (6):

$$G_{ud}(s)=\frac{\hat{u}_O(s)}{\hat{d}(s)}\Big|_{\hat{u}_i(s)=0}=G_{id}(s)R=\frac{U_i}{\frac{L}{R}s+1} \quad (6)$$

In Equations (5) and (6), U_i is the equivalent DC input voltage; L is the output filter inductance; R is load resistance. Main circuits of inverter are composed of proportional part and inertia part in view of control structure.

3.2 Mathematical Modeling of Inner Loop Control System

The structure of control circuit of inner loop current is shown in **Figure 3**. The inductance current $i_L(t)$ is gained by input voltage $u_i(t)$ and output voltage $u_o(t)$, $i_L(t)$plus resistor R_f and then change into voltage signals $i_L(t)R_f$. $i_L(t)R_f$ plus the slope compensated voltage $u_a(t)$,which import to the negative terminal of PWM comparator. $u_c(t)$ is reference voltage of positive terminal of PWM comparator. Relations expression of duty cycle d and input



(a)

(b)

Figure 3. Model of current-injection controller

voltage $u_i(t)$, output voltage $u_o(t)$, slope compensation voltage $u_a(t)$, reference voltage $u_c(t)$ and inductance current $i_L(t)$ are derivatived by application of Space State Average method.

From **Figure 3(b)**, we can indicates expression of average voltage of inductance sampling current in the opening of each cycle as Equation (7).

$$R_f i_{Lavg}(t) = \frac{1}{T_s}\int_0^{T_s} R_f i_L dt = \frac{1}{T_s}\left(S_{ABEF} - S_{\Delta ACH} - S_{\Delta BDH}\right) \tag{7}$$

In the Equation (7), SABEF、S ACH and S BDH are the area of rectangular ABEF, triangle ACH and the triangle BDH in **Figure 3(b)**, according to Equation (7), we have Equation (8):

$$R_f i_{Lavg}(t) = u_c(t) - m_a dT_s - \frac{1}{2}m_1 d^2 T_s - \frac{1}{2}m_2\left(1-d\right)^2 T_s \tag{8}$$

In Equation (8), we can obtain Equation (10) after adding disturbing variable and taking into account that the system state has no relation to the slope compensation voltage shown in Equation (9).

$$m_a(t) = M_a \tag{9}$$

$$R_f\left(I_L + \hat{i}_L(t)\right) = \left(U_c + \hat{u}_c(t)\right) - M_a\left(D + \hat{d}(t)\right)T_s - \frac{1}{2}\left(M_1 + \hat{m}_1(t)\right)\left(D + \hat{d}(t)\right)^2 T_s - \frac{1}{2}\left(M_2 + \hat{m}_2(t)\right)\left[1 - \left(D + \hat{d}(t)\right)\right]^2 T_s \tag{10}$$

In Equation (10), $\hat{i}_L(t)$、$\hat{u}_c(t)$、$\hat{d}(t)$、$\hat{m}_1(t)$、and $\hat{m}_2(t)$ are corresponding disturbing variable .We can obtain Equation (11) after linear treatment of Equation (10)

$$R_f\hat{i}_L(t) = \hat{u}_c(t) - M_aT_s\hat{d}(t) - \frac{1}{2}D^2T_s\hat{m}_1(t) - \frac{1}{2}\left(1-D\right)^2 T_s\hat{m}_2(t) \tag{11}$$

In view of equation of $\hat{m}_1 = R_f\dfrac{\hat{u}_i - \hat{u}_o}{L}$ and equation of $\hat{m}_2 = R_f\dfrac{\hat{u}_o}{L}$ in Buck Converter, the transfer function of inner peak current control circuit shown in Equation (12) can be obtained from Equation (11).

$$\hat{d}(s) = \frac{1}{M_aT_s}\left[\hat{u}_c(s) - R_f\hat{i}_L(s) - \frac{R_fD^2T_s}{2L}\hat{u}_i(s) - \frac{R_f(1-2D)T_s}{2L}\hat{u}_o(s)\right] \tag{12}$$

3.3 Mathematical Model of Double Closed-Loop Control System

In order to improve accuracy of control system, reduce steady error, the average output current or voltage are sampled in this control system, and compared to the inner given compensation signal $u_c(s)$. Primarily role of outer loop regulator is to improve and optimize system performance; PI regulator is used in this paper. According to Subsection 3.1 and Subsection 3.2, overall control system diagram based on model of double closed-loop current is shown in **Figure 4**, the DC signal $u_i(s)$ can be treated as system disturbance.

According to **Figure 4**, open-loop transfer function of the inner loop current is expressed as Equation (13).

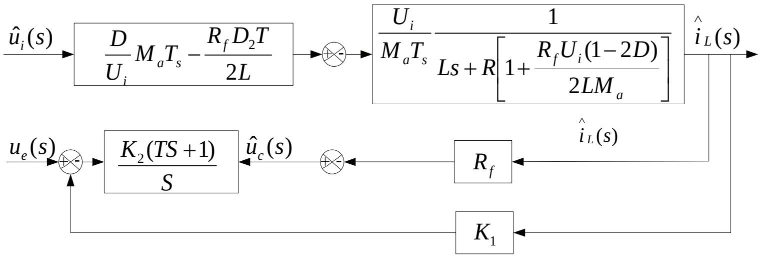

$\hat{u}_i(s)$
$\frac{D}{U_i}M_aT_s - \frac{R_fD_2T}{2L}$
$\frac{U_i}{M_aT_s}\frac{1}{Ls+R\left[1+\frac{R_fU_i(1-2D)}{2LM_a}\right]}$
$\hat{i}_L(s)$
$u_e(s)$
$\frac{K_2(TS+1)}{S}$
$\hat{u}_c(s)$
$\hat{i}_L(s)$
R_f
K_1

Figure 4. Block diagram of the double loop system

$$G_1(S) = \frac{R_f U_i}{M_a T_S} \frac{1}{LS + R[1+\frac{R_f U_i (1-2D)}{2LM_a}]} \tag{13}$$

Equation (14) is inner closed-loop transfer function.

$$W_1(s) = \frac{i_L(s)}{u_c(s)} =$$

$$\frac{U_i}{M_a T_s Ls + M_a RT_s + \frac{RR_f U_i (1+2D) T_S}{2L} + R_f U_i} \tag{14}$$

Equation (15) is transfer functions of entire open double closed-loop system.

$$G(s) = \frac{K_2(TS+1)}{S} \cdot W_1(S) \cdot K_1 =$$

$$K_1 K_2 U_i \frac{TS+1}{S} \cdot \frac{1}{M_a T_s LS + M_a RT_s + \frac{RR_f U_i (1-2D) T_s}{2L} + R_f U_i} \tag{15}$$

Equation (16) is transfer functions of closed-loop transfer function for the entire system.

$$W(S) =$$

$$\frac{K_2 U_i (TS+1)}{S(M_a T_S LS + M_a RT_s + \frac{RR_f U_i (1-2D) T_s}{2L} + R_f U_i) + K_1 K_2 U_i (TS+1)} \tag{16}$$

In Equation (16), D is duty cycle; U_i is inputting DC voltage; *TS* is the inverting cycle; R_f is sampling resistor of the inner current; M_a is the rising slope of compensation voltage; *L* is Output filter inductance; *R* is pulsed arc load; K_1 is feedback coefficient of outer loop current; K_2 is adjusting gain; *T* is time constants of regulator.

3.4 Analysis of Dynamic Characteristic

The cutoff frequency of open-loop that is an important characteristic index that is the embodiment of dynamic response of control system [13]. Dynamic characteristic of welding power are analyzed as mathematical model established in Subsection 3.3. Provided outer loop is a simple proportional control mode, and the open-loop system fc is frequency when open loop gain equals to1, according to Equation (15), and set $\left\| G\ (j2\pi f_c) \right\| = 1$, we have:

$$\left\| G\ (j2\pi f_c) \right\| =$$

$$\left\| \frac{K_1 K_2 U_i}{M_a T_s} \frac{1}{Lj2\pi f_c + R\left[1 + \frac{R_f V_i (1-2D)}{2LM_a} + \frac{R_f U_i}{R}\right]} \right\| = 1 \tag{17}$$

In this paper $\left\| Lj2\pi f_c \right\| >> \left\| R\left[1 + \frac{R_f V_i (1-2D)}{2LM_a} + \frac{R_f U_i}{R}\right] \right\|$, Equation (17) can be taken form as following:

$$f_c \approx \frac{K_1 K_2 U_i}{2\pi M_a T_s L} \tag{18}$$

In Equation (18), open-loop traversing frequency f_c is proportional of outer loop resistor, DC input voltage and gain of outer loop adjuster. But the traversing open-loop frequency is inversely proportional to the rising rate of compensation voltage, switching cycle and output inductance. Since outer loop resistor is limited by linear adjustment range of voltage of control circuit, opening traversing frequency of control system can be increased by the way of reducing rising ratio of compensation voltage and output inductance, and increasing frequency of inverter. From above analysis, dynamic response of double closed loop control system is improved greatly by inner loop current control.

4. Experimentation

Dynamic characteristics of arc welding inverter are always defined as the relationship between output current or output voltage and time when load instantaneous change, which is a major performance index of arc weld power source. Two sets of experiments of constant current outer characteristic of arc weld power source are done to prove the effect of theatrical analysis based on the double closed-loop constant current control mode. The experiments of arc welding are current response under condition of the instantaneous change of the given signal and current response under condition of instantaneous change of the given load.

The curve of **Figure 5** is current response when current instantaneous change from 0 A to 430 A under the simulated load of 0.1 Ω. In figure5, the current instantaneous change from 50 A to 320 A just needs time of 2 ms, which conclude that system has better dynamic performance through this experimentation.

The curve of **Figure 6** is current response curve measured by given current value of 100A and simulated load changing from 0.09 Ω to 0.03 Ω. In **Figure 6**, the given current instantaneous change from 130 A to 100 A just needs time of 4 ms, which conclude that system has better dynamic performance and a constant current outer characteristic.

5. Conclusions

Mathematical model of SAW weld soft-switching inverter based on a double closed-loop constant voltage and current control is established. Based on mathematical model, dynamic performance is analyzed, and dynamic characteristic curve of arc weld power source is tested in

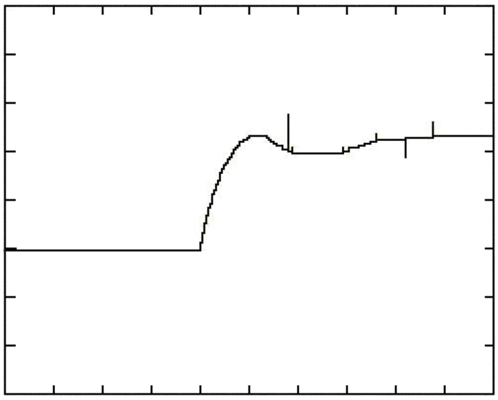


Figure 5. The current response curve while instantaneous change from 0 A to 430 A

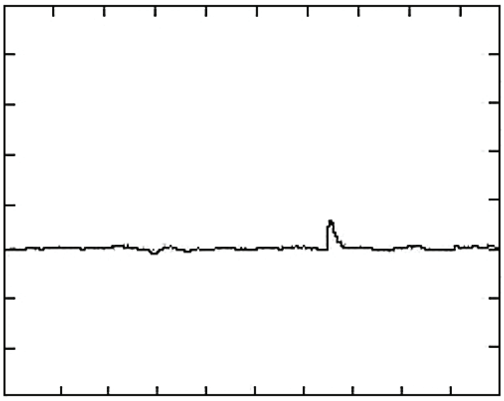


Figure 6. The current response while the load changing

this paper, it shows that double closed-loop control can improve dynamic characteristics of arc welding power source, which can meet request of SAW technology.

REFERENCES

[1] Y. B. Li, "Study on Soft-Switching Inverting High-Speed Double Wire Pulsed MAG Welding Equipment & its Digital Synchronic Control Technology," South China University of Technology, Guangzhou, 2004.

[2] Z. M. Wang, "Study on Novel Submerged Arc Welding Inverter and it's Intelligent Welding System," South China University of Technology, Guangzhou, 2002.

[3] R. Brown and R. D. Middlebrook, "Sampled-Data Modeling of Switching Regulators," Record of PESC, 1981, pp. 349-369.

[4] B. T. Lin and S. S. Qiu, "Symbolic Analysis of PWM Switching Power Converters," *Acta Electronica Sinica*, Vol. 24, No. 9, 1996, pp. 83-87.

[5] G. Chen and Y. X. Xie, "Modeling of PWM Switching Converters. Telecom Power Technologies," Vol. 23, No. 1, 2006, pp. 22-24.

[6] Y. Li and H. Z. Wang, "Averaged Modeling and Simulation of Unideal Buck Boost Converter in State of Continuous Conduction Mode," *The World of Power Supply*, Vol. 2006, No. 8, pp. 38-41.

[7] Q. M. Niu, P. Luo, Z. J. Li and B. Zhang, "Space State Average Model of PSM in Boost Converter," *Journal of Electronics & Information Technology*, Vol. 28, No. 10, 2006. pp. 1955-1958.

[8] G. X. Wang, Y. Kang and J. Chen, "Control Modeling of a Single-Phase Inverter Based on State-Space Average Method," *Power Electronics*, Vol. 38, No. 5, 2004, pp. 9-12.

[9] R. D. Middlebrook and S. Cuk, "A General Unified Approach to Modeling Switching Converter Power Stages," Record of PESC, 1976, pp. 18-34.

[10] "Vatche Vorperian. Simplified Analysis of PWM Converters Using Model of PWM Switch," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 26, No. 3, 1990, pp. 490-505.

[11] V. Vlatkovic, J. A. Sabate, R. B. Ridley, *et al.*, "Small-Signal Analysis of the Phase-Shifted PWM Converter," *IEEE Transactions on Power Electronics*, Vol. 7, No. 1, 1992, pp. 128-135.

[12] M. J. Schutten and D. A. Torrey, "Improved Small-Signal Analysis for the Phase-Shifted PWM Power Converter," *IEEET Transactions on Power Electronics*, Vol. 18, No. 2, 2003, pp. 659-669.

[13] S. S. Hu, "Principles of Automatic Control," Publishing Company of Science, Beijing, 2002.

J. Software Engineering & Applications, 2010, 3, 728-735
doi:10.4236/jsea.2010.37084 Published Online July 2010 (http://www.SciRP.org/journal/jsea)

Memetics of the Computer Universe Based on the Quran

Pallacken Abdul Wahid

Hira, J.T. Road, Thalassery, Kerala, India.
Email: pawahid@hotmail.com

Received April 26th, 2010; revised May 17th, 2010; accepted May 27th, 2010.

ABSTRACT

The emerging concept of computer universe has the potential to bring about a radical change in our perception of the world. Energy is word of God that carries His commands. It derives its properties in accordance with the divine instructions immanent in it. Memetics is the science of information carried by the entity called energy. Living and non-living systems represent two different languages in which information content of energy exists. These may be distinguished as bioprogram (biological information) and abioprogram (chemical information) respectively. While abioprogram can be explained in terms of a structure-code concept, the bioprogram is intangible to human beings and is stored on the chromosomes of the cell. It is non-physical in nature but requires a physical medium for its storage like the computer program. An organism is natural biocomputer or biorobot. All organisms except Homo sapiens are totally programmed unconscious systems like man-made robot. Man is the only conscious, freewilled robot (abd in Arabic) of God. Consciousness and freewill are the attributes of the unique processor of human being, the mind (qalb in Arabic). The computer model of the organism helps us to define and explain the phenomena of life and death. The phenomenon of life is the manifestation of the execution of the biosoftware while death is the result of deletion of the biosoftware. A dead body is like a computer without software. Man-made computer, robot, etc., which run on man-made software are forms of "artificial life". The basic change that the computer concept of the universe brings into our present knowledge of the universe and cosmology is that it is the divine information carried in energy that represents the underlying reality of the universe.

Keywords: *Computer Universe, Memetics, Abiomemetics, Biomemetics, Phenomena of Life and Death, Natural Biosoftware Engineering*

1. Introduction

The universe presents itself a self-propelled, self-regulated, self-sustained system. An organised reality exists in nature. The countless number of celestial bodies of colossal size tracing their own paths in the cosmos without collision in its 13.75-billion-year history is indicative of a perfectly programmed behaviour. The gravitational force responsible for this meticulous and amazing consistency in the peripheral motion and recurring relative positions of stars and planets do not operate in an arbitrary manner, but obey certain laws prescribed by the Creator. It is because of this, we are able to formulate principles which can reflect the natural order. The high degree of success achieved by man-made mathematical models in describing and/or predicting several natural phenomena adds strength to this reasoning. Every natural process is spontaneous phenomenon. Physical, chemical and biological processes are spontaneous as though the reacting species know what they should produce under different conditions. Their properties and mode of behaviour in diverse environments are fixed. A plant or an animal develops through execution of instructions in the sequence specified in the program stored in its starting cell (e.g., zygote, seed, etc.). All these are suggestive of a computerised setup. Although science covers many aspects of the universe from subatomic level to galaxies and beyond, there are certain fundamental questions like what is energy, how the various components of the universe acquired their characteristic properties, how the laws and rules governing them came into being, what is life, what is death, etc., that remain unanswered in science. These issues constitute some of the "fundamental unknowns" about the universe that can be unravelled by resorting to a computer model in conjunction with the Quran. This paper addresses these issues and presents a summary of my work in this line [1,2].

2. The Computer Universe

The universe is a computer designed, programmed and created by God. Konrad Zuse who built the first programmable computer was the first to suggest in 1967 that the entire universe was being computed on a computer, possibly a cellular automaton [3]. He referred to this as "Rechnender Raum" (Computing Cosmos or Computing Space), which in fact started the field of digital physics. Jurgen Schmidhuber of Dalle Molle Institute for Artificial Intelligence (IDSIA), Switzerland, proposes an algorithmic theory of everything. Schmidhuber assumes: "a long time ago, the Great Programmer wrote a program that runs all possible universes on His Big Computer….Each universe evolves on a discrete time scale….Any universe's state at a given time is describable by a finite number of bits" [4]. In 1998, I proposed a computer model of the universe in the light of the Quran in my book *The Divine Expert System* [5]. Both physical universe and biological organisms have been described as natural computer systems. Four years later, Seth Loyd of Massachusetts Institute of Technology, USA, published a paper suggesting that the universe is a quantum computer [6]. He also elaborated the concept in his subsequent book [7]. Another notable work in this field is that of Stephen Wolfram from USA [8]. According to him, all of reality might result from a kind of algorithm, like a computer program being enacted again and again on the underlying building blocks of space and matter. He argues that the whole universe can be viewed as one huge cellular automaton. Recently Denis Bray suggested every living cell is a computer [9].

3. Memetics of the Universe

Memetics is science of information based on the computer model of the universe. The universe is energy-filled space. Energy is information. No one knows the fundamental nature of the entity called energy. Energy lies in the realm of the unknown (*i.e.*, intangible to humans). For practical purposes we define energy in terms of its manifested characteristics. A well-known definition of energy is that given by Dave Watson: "Energy is a property or characteristic (or trait or aspect?) of matter that makes things happen, or, in the case of stored or potential energy, has the "potential" to make things happen" [10]. Energy exists in various forms such as matter, kinetic energy, potential energy, heat, magnetism, electricity, electromagnetic radiation, sound, etc. All these forms have been described the way we perceive.

The Quran informs us that when God wants to create a thing, He says "be to it and it comes into being" (Q. 2:117). Note that God says "be" to "the thing" He wants to create. This means that the thing to be created is already there in a virtual form (intangible to man). From the Quran, it is possible to infer that the virtual form is nothing but God's words. The word of God is instruction or command (Q. 11:44) and His words are infinite: "And if all the trees on earth were pens and the ocean (were ink), with seven oceans behind it to add to its (supply), yet would not the words of Allah be exhausted (in the writing): for Allah is exalted in power, full of wisdom." (Q. 31:27). God's words form the instructions (programs) in intangible form, which we call energy. Therefore, energy is God's word that represents the divine software (commands or instructions).

In the light of these revelations, the phenomenon of creation can be explained in terms of a *ghayb-shahadat* paradigm. The verse 31:27 indicates the existence of a large collection of God's words (intangible energy). These in fact constitute the divine programs for whatever thing God wants to create. The word "it" in the verse 2:117 given above refers to the divine program concerned. That is, when God wants to create a thing, He needs say only "be" to "it" (*i.e.*, the appropriate program of the thing to be created) and "it comes into being" (i.e., it becomes tangible to man). This implies that when God gives the command "be" to the intangible (*ghayb* in Arabic) program of a thing to be created, it transforms itself into the form tangible (*shahadat* in Arabic) to man; i.e., into the form which human mind can process and interpret. The mind deciphers it in accordance with the human biosoftware, which results in conscious perception. The universe is therefore what human mind constructs as stipulated in the biosoftware. The phenomenon of creation can therefore be conceived as the transformation of intangible energy into tangible form. This is much like the production of a hardcopy (*shahadat*) of an intangible (*ghayb*) document by a computer when a command is given to it.

Two categories of universal components namely, non-living and living, can be distinguished. The non-living systems may be thought of as being run on abioprogram (chemical information) and the living systems on bioprogram (biological information). These are the two basic forms in which information exist in the universe. The unit of information (*i.e.*, energy) may be represented as "meme". The term "meme" was introduced by Richard Dawkins to mean "replicator" [11]. However the term is used here not with the connotation of a "replicator" or with the other characteristics originally assigned to it. *Meme is defined here as a piece of information (energy) in the abiotic and biotic segments of the universe.* The meme based on abioprogram may be referred to as "abiomeme" and the meme based on bioprogram may be called "biomeme" [2].

3.1 Abiomemetics

Abiomemetics is science of chemical information. It can be best understood in terms of a structure-code concept, which is illustrated here taking the example of matter form of energy. The atom is considered here as the basic unit of matter for illustrating the concept (**Table 1**). The structure signifies a code "written" in a special language (abioprogram) like the symbolic language used in computer machines. The semantic content of the code is deciphered in accordance with the abioprogram and the structure derives its properties. The Quranic message that the universal components carry God's commands (Q. 41:12) can be explained in this way. The numerous substances found in the universe owe their vastly diverse properties to their structures, which, in turn, are decided by the composition and arrangement of atoms. Structure at the level of a molecule (substance) is the totality of nuclide composition and arrangement of the atoms. In the structure-code concept, nuclides form the alphabets and along with their arrangement, as in a word, through bonding, etc., the code is deciphered in terms of its properties (**Figure 1**). A set of alphabets can carry meaning only if it has affiliation with a language. The meaning of a word depends on its alphabetic composition as well as the order in which they are arranged. Two words may be different in their alphabetic composition or in their arrangements. For instance, English words "nest" and "sent" have the same alphabets but different arrangements whereas the words "take" and "buy" are different in their alphabetic composition. Likewise, different chemical structures are formed based on the composition and arrangement of the atoms of the elements. The structures of n-butane and iso-butane have the same elements and same number of atoms with the chemical formula of C_4H_{10}; but the arrangement of the atoms is different in the two substances. These two structures correspond to English words "nest" and "sent". The chemical structures of water (H_2O) and benzene (C_6H_6) are different in their elemental (alphabet) composition. They are comparable with English words "take" and "buy" (**Figure 1**). By this analogy, the phenomenon of how chemical structures (substances) acquire their properties based on the divine abioprogram can be explained. Periodicity in the properties of elements which provide the basis for their classification (Periodic Table) and also for the prediction of properties of a hitherto unknown element; specificity in the change of properties of a substance with a change in structure, etc., are clearly the clauses of the abioprogram operating at different levels of structural hierarchy. Recognition of at least some of these rules is now helping us in the search for new compounds with specific properties. For instance, computer-aided molecular modelling (CAMM) has become a powerful tool for studying virtually any chemical structure. The method works on the reverse logic of structure-property relationships. In this case, we specify the properties; the computer will give us the structure of the molecule in return. Use of this technique in the search for new drugs has enabled the researchers to cut short the long list of candidate molecules to a smaller number expected to have the required biological activity. In fact by studying the structure-property relationships, we are deciphering the abioprogram at various hierarchal levels of the universe. The chemical structure may be thus likened to a kind of algorithm conforming to the abioprogram. The universe is therefore nothing but information-laden system.

Table 1. Property acquisition by non-living matter (abiomeme) based on structure-code concept.

Building block	Meme	Software	Function
Alphabet	Word	English	Meaning
Element	Molecule	Abioprogram	Properties

Note: Atom is taken as the unit of matter for convenience in illustrating the principle.

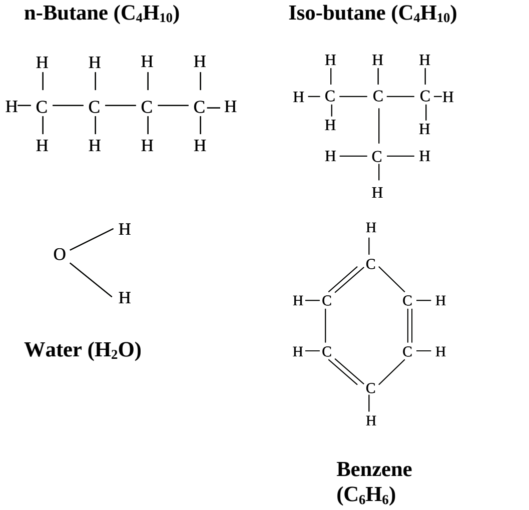


Figure 1. Representation of chemical structures as abiomemes.

3.2 Biomemetics

Biomemetics is science of biological information. Biosystems carry divine instructions in a different way. "Breathing of *rooh*" into a clay model to create man (Adam) mentioned in the Quran (Q. 15:26-29) and "breathing of life" mentioned in the Bible (Genesis 2:7) refer to one and the same event – installation of divine biosoftware in a clay model of man. Upon installation of the *rooh* (the term *nafs* is also used in the Quran in the context of man) in that non-living clay model, it sprang to life much like a lifeless computer springs to "life" when software is installed. Software is the invisible soul of a computer. Similarly, the invisible soul of an organism is its biosoftware. The Scriptural revelations make clear distinction between the way in which God's instructions (programs) are carried by the non-living and living components of the universe. The Quran further informs us that it is the removal (or in computer parlance, "deletion") of the *nafs* (biosoftware of human being) that causes death (Q. 6:93). In other words, a dead body is like a computer without software.

A system is said to be "living" if it carries software. Therefore the phenomenon of life can be defined as the manifestation of the execution of the program. Going by this definition, all the so-called non-living and living systems are in reality "living systems" in their own right as they do carry divine programs. The physical universe is in reality a "living system" as it operates on the abioprogram. We may distinguish the so-called "living" and "non-living" systems as two different forms of life as they are operated on different software. However, we shall retain the conventional terms "living" and "non-living" for convenience. Computer, robot, etc., are also living systems as they work on man-made programs. We may distinguish them as forms of "artificial life".

4. Organism—Natural Biocomputer or Biorobot

An organism is a biocomputer or biorobot depending on its configuration. The biological program (bioprogram) at the level of the species is termed microbioprogram. Microbioprogram at the level of the member of a species is termed biomemome. Cell is biochip, the organisational unit of the biosystem. The cell has both hardware and biosoftware. The hardware is produced in the cell as per the program appropriate to execute the program. The chemical structures (including DNA) in the cell constitute the hardware while biosoftware is stored on the chromosomes as invisible information. At the level of the organism, tissues and organs make the hardware.

A distinguishing feature of the bioprogram from abioprogram is that while abioprogram is encoded by the structure, bioprogram requires a physical medium for its storage as in our computer. The chromosomes in the body cells serve as the storage device (hard disk) of the biosystem. The arrangement of biomemes on the chromosomes, *i.e.*, biomemory organization may be viewed as the "biomemogram" of a species (**Figure 2**). Although we may get some broad idea of memetic allocation on the chromosomes (e.g., allelic loci) by studying the inheritance pattern of characters through breeding trials, it is not possible to map out their storage pattern on the biomemory. A memory sector on the chromosome may be storing one or more biomemes that constitute the program for a specific biological activity or function. It is the totality of the biomemetic package that makes the biomemome of the individual.

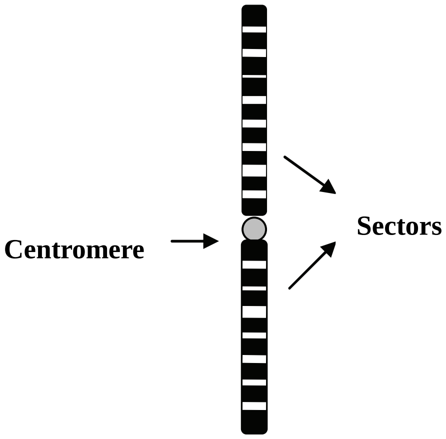


Figure 2. Biomemory organization on a chromosome. Note: The chromosome (biomemory) is divided into many sectors.

An organism may be unichipped, *i.e.*, unicellular (e.g., bacteria) or multichipped, *i.e.*, multicellular (e.g., plants, animals). Multicellular organisms like animals are functionally comparable with man-made robots. They can be categorised as biorobots as they are totally programmed unconscious biosystems. On the other hand man can be described as the conscious, intelligent, freewilled robot of God in the light of the Quranic revelations [1,2].

God created man to serve Him. Man is addressed by God as "*abd*" meaning servant (Q. 51:56). The Arabic word *abd* is synonymous with the English word "robot" (servant). The root of this word may be traced to the Czech play *Rossum's Universal Roboters* of the 1920s in which human workers were portrayed as "robots" (*robota* in Czechoslovakian means servitude). In this context it may be mentioned that an understanding of God's purpose of creation of man and the universe is very much necessary to comprehend the overall mission of God. This aspect has been discussed in detail elsewhere [2]. Robot is a programmed dedicated machine designed to perform certain desired tasks. It is essentially a computer with three additional features namely, sensors (which receives data from the environment), microprocessors (which transform data into information) and actuators (or muscles, which control the energy requirement). An animal biorobot with sensory organs and brain (the location of CPU) is comparable with man-made robot but God's human robot is far beyond that level of sophistication. Man is bestowed with an additional processor, *qalb* (mind), which is capable of conscious perception of the world and has the freedom to take decision and act. Several functions of human *qalb* are identified in the Quran.

All of them are cases of conscious perception. The *qalb* is the part that thinks (i.e., processing information) and learns (Q. 22:46), and understands (Q. 7:179). The Quran further reveals that the faculties of hearing, seeing, understanding and feeling are made only for human species. "Say: It is He (Allah) who created you and made for you the faculties of hearing, seeing, feeling and understanding; little thanks it is you give." (Q. 67:23). These faculties are associated with the *qalb* (human mind), which is responsible for the conscious perception of the world. The Quran also reveals that these faculties are absent in animals although they have eyes and ears (Q. 7:179; 25:44). Clearly animals are totally programmed unconscious biorobots like man-made robots.

The role of mind comes in the execution of the conscious activities; that is, activity decided and/or executed by the mind is a conscious activity. Similarly any signal received through input ports (**Figure 3**) and processed by mind results in conscious perception. Consider the visual perception. Eye is the input port for electromagnetic radiation. It receives the radiation signal from outside, does the preliminary processing and then transmits it to the brain location concerned to decode its information content. The human mind decodes it in accordance with the biosoftware and creates its translated version. For example, let us say we are looking at a red object. This means that the object in reality is emitting or reflecting electromagnetic radiation in the 650-700 nm wavelength range. This radiation enters our eyes and its information is deciphered by our mind in accordance with our biosoftware to perceive it as red. In other words the object has no colour; the colour of the object is generated inside our brain. If our biosoftware stipulates blue colour for that wavelength range, we would have seen that object as blue.

Our biosoftware also prescribes limits to our perceptional potentials. For example, our visual perception is restricted to within 400-700 nm wavelength range. Therefore we cannot see X-rays and gamma rays because their wavelengths are outside of the limits set by our biosoftware in spite of the fact they also belong to the same form of energy, electromagnetic radiation. Our auditory perception is within the sound wave frequency range of about 12-20000 Hz outside of which we cannot hear and so on. It also assigns thresholds and maxima for each kind of sensory perception. For example, for a sound wave to become audible to us it should have a minimum intensity. There is also a maximum level beyond which our perception of sound will not be enhanced. In other words, human mind is unable to create tangible image to any signal that falls outside of the limits prescribed by the biosoftware. This implies two things; one is that the world around us is inherently soundless, colourless, shapeless and tasteless. It is the human mind that creates these characteristics based on the biosoftware and imparts them to the outside world; and the other is that it is our biosoftware that determines what to perceive and how to perceive. Obviously, many forms of energy remain unperceived by us. This is the realm of the unknown (intangible), which the Quran describes as "*ghayb*" (in the computer jargon, virtual). The realm of the known or that perceived (tangible) by man is referred to in the Quran as "*shahadat*". God alone has the knowledge of both the intangible and the manifest (Q. 59:22). The dichotomy of *ghayb* and *shahadat* is relevant only in the human context. The fact that conscious perception of the world is determined by human biosoftware and that energy (information) does exist outside of the domain of human perception indicates the existence of the intangible.

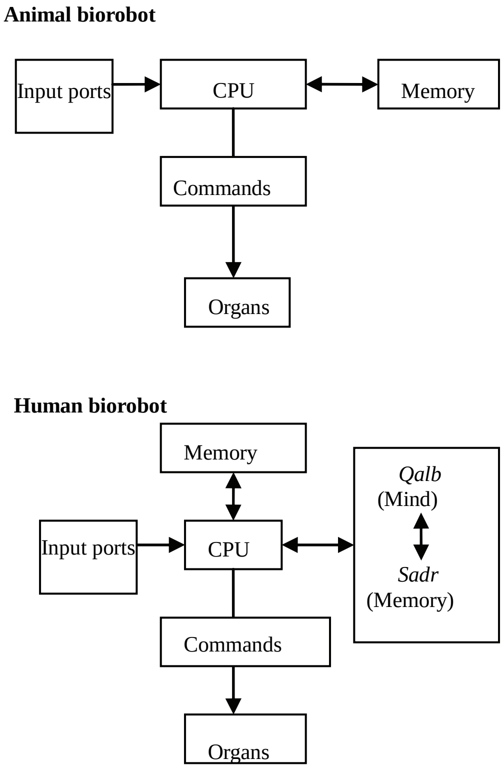


Figure 3. Contrast between animal and human biorobot systems.

As in animal biorobot, the CPU of human system is responsible for the house-keeping activities, i.e., internal biological activities that sustain the system. In addition to mind, human robot also has *sadr* (memory), where the conscious activities are recorded and stored. Cerebellum is thought to be a likely location of *sadr* in the light of certain Quranic verses [2]. The information stored in *sadr* in fact forms the dossier of conscious activities (including the acquired data and information) carried out by mind during the life of a person (for evaluation by God in the hereafter). The mechanism of storage could be identical to that of biosoftware storage on the chromosome.

Characterisation of biological information as non-particulate biomemes is opposed to the current particulate concept of biological information. Biologists do not distinguish biological information from chemical information but believe the DNA molecule (the gene) encodes biological information also. That is, the genome (the totality of the genes) structure besides having its physical and chemical properties also encodes the biological program of the organism. In effect, molecular biologists are superimposing biological information over chemical information literally making the molecule an entangled web of contrasting information. The computer model of the organism adopts Wilhelm Johannsen's non-physical gene concept. This concept agrees well with the Quranic and the Biblical revelation of intangible biosoftware. The computer model permits us to define the phenomena of life and death while in biology (the science of life), which is based on the particulate gene, it has not been possible to define these phenomena.

While proposing the concept of gene in 1909, Wilhelm Johannsen cautioned against two things. One was against considering the gene for a particular character, which implied that the genetic program should be viewed as an integrated program, and the other was against treating the gene as particulate [12]. Both these warnings have since proved correct. Today in the era of genomics, biologists are unable to define what the gene is and attribute genes for different characters. [13]. Particulate gene had never been the idea of early geneticists. The scientific community was, however, not comfortable with a hypothetical non-material entity having a "metaphysical" aura around it. Thus with the elucidation of the double helical structure of DNA and confirmation of its role in protein synthesis, the non-physical gene metamorphosed into physical gene. The molecular gene was born that way.

5. Natural Biosoftware Engineering Mechanisms

Natural biosoftware engineering mechanisms provide the tool to create diverse bimemomes. Since the biomemes are stored on the chromosome in sectors, qualitative and quantitative changes in bioinformation can be brought about by shifting, adding, deleting and shuffling the sectors. The role of transposable elements is crucial in many of these mechanisms. The bioinformation content of a cell as a whole can also be altered by increasing or decreasing the number of chromosomes. There are also mechanisms for multiplication of the cells as well as for eliminating the unwanted chromosomes or chromosome sectors. We find a variety of natural biosoftware engineering mechanisms in operation that can do all these and more. Some of these are: crossing over during meiosis (gamete formation), non-disjunction of sister chromatids during mitosis and meiosis leading to increase in chromosome number, the so-called chromosomal aberrations such as deletion, duplication inversion or insertion of chromosome sectors, etc. Although these phenomena are generally treated as errors or mistakes by biologists, they in fact form powerful molecular tools to produce radically different chromosome compositions and hence bioinformation content. All these processes viewed in the light of the computer concept are biosoftware-driven phenomena and not mistakes. It is these biosoftware engineering processes that produce biomemetic variability and hence phenotypic diversity in the offspring as programmed.

6. Developmental Biology

Another area where the computer concept of the organism throws light is developmental biology. It has not been possible to explain based on the molecular gene how anatomically and functionally different tissues develop. Until recently, it was thought that mitosis (a kind of cell division) taking place during development of an individual from zygote produces daughter cells with identical genomes. A recent finding that different tissues carry different genomes [14,15] called into question the current belief. The biomemetic concept adopted in the computer model however enables us to explain the phenomenon of ontogenetic development of an individual in conformity with this discovery [2]. Let us examine the development of a human individual from zygote based on the biomemetic concept

An important feature of the biomemome is that it produces a dynamic phenotype that is changing continuously from time zero (the time at which say a zygote starts developing) to its death. The biomemome is thus an integrated program deciding the phenome (phenotype) at every instant. The biomemome, in simple terms, is an integrated biological program of an individual. Development of an individual is just one phase in the continuous execution of the biomemome of the individual. Development of an individual presents the scenario of creating tissues (or group of cells) with different tasks assigned to them. This is achieved through mitotic cell division. Mitotic division is not mere copying or multiplication of cells as is believed now. Although the entire biomemome stored in the zygote (the biomemome of the individual) is copied into the daughter cells during mitosis, it can be assumed that the process also assigns to the resulting cells a set of biomemes to be in operation in each of them. In this way, differentiation of the operable biomemes progresses as dictated by the biomemome culminating in the formation of tissues. The set of operable biomemes in a given tissue may be designated as its "operamome". The hardware (including DNA) of the cells of a tissue is synthesized to suit the functions of the operamome. This is reflected in the differences in the cell structures among the tissues. For example, a muscle cell is structurally and functionally different from a nerve cell, neuron. They both carry identical biomemomes but different operamomes. Thus, even though all the cells carry the biomemome of the individual, the operamome varies with tissue.

An example of visible manifestation of operamomic transistion in the phenotype is the metamorphosis of larva into butterfly. The biomemome of the organism creates the butterfly phenotype through a selective switching on and switching off of biomemes. As a result, different sets of biomemes (operamomes) come into operation creating a butterfly from a totally different phenotype, larva.

7. The Abiomeme-Biomeme Interactions

Phenotype is the product of biosoftware-environment interaction. Although the importance of environment in moulding the phenotype is recognized, the actual relationship between the two is not well understood. This can be explained convincingly through biomemetic approach. Not all the biomemes of an operamome in a tissue are in operation at any given instant as in the case of instructions carried in our computer program. Only those biomemes required at that instant are in operation. The others are silent. A latent biomeme comes into operation at the time stipulated in the program (e.g., development of sexual characters at puberty) or when the situation (e.g., environmental stress condition) warrants. Thus we can say the biomemes in operation in summer are not the same as those operate in winter. If a person spends some time in a hot place and then enters an air-conditioned room, the operamome will also change accordingly as specified in the biomemome. The environmental condition thus acts as switch for the right biomemes (if available) to come into play. When an insecticide is sprayed against a pest in a crop field, and if the pest has the biomemetic package that can resist its harmful effect, the chemical will act as stimulus to turn on those biomemes which in turn will confer protection to the organism against that chemical. The consequence of this memetic operation is “resistance development” in the pest against that chemical. Although the biomemetic package has been present in the organism all the while, it has not been in operation as the situation warranting its role has not arisen until then. Consequently, the cell may not be having the necessary hardware or it may have to modify some available hardware for execution of such rarely executed programs. Nevertheless when the situation arises for the biomemetic package to come into operation, it requires the right hardware for its execution. Therefore the cell synthesizes the necessary hardware (any structure including DNA) or modifies the existing hardware according to the program to make way for the execution of the newly turned-on memes. It is such events that in fact biologist refer to as “cell-induced mutagenesis”. There are several reports relating to this phenomenon, which cannot be explained based on particulate gene concept but can be convincingly explained biomemetically. A couple of examples are given here.

Miroslav Radman, a molecular geneticist at the Universite Rene Descartes in Paris, discovered the phenomenon of cell-directed mutagenesis in 1970. He showed that bacteria harboured a genetic program to make mutations. At that time, no one believed this heretical proposal [16]. Many biologists were skeptical about this discovery because genetic mutation was considered as a random phenomenon. Although the discovery of a new group of DNA-synthesizing enzymes (polymerases) as the generator of mutations in times of stress [16] gave credence to Radman’s finding, it was the work of Cairns *et al*. that galvanized the critics. In 1988 molecular biologist John Cairns and his colleagues at the Harvard School of Public Health reported induced mutations of various elements of the lac operon changes in *Escherichia coli* bacteria [17]. Their results showed that bacteria could induce specific mutations depending on their environmental conditions. But unfortunately these discoveries were sidelined. A recent report of resistance of bacteria to antibiotics also provides evidence of cell-induced mutation. Commenting on the work of Kohanski *et al.* [18], Martin Enserink writes: “Traditionally, the development of antibiotic resistance—a big and growing problem in medicine—has been seen as a passive phenomenon. Haphazard mutations occur in bacterial genomes, and bacteria randomly swap genetic elements. Every now and then, a mutation or a bit of newly acquired DNA enables the microbes to detoxify antibiotics, pump them out of the cells, or render them harmless in another way. When these microbes are exposed to antibiotics, natural selection will allow them to outcompete the ones that aren’t resistant. But in the past 6 years, a different view has emerged, says microbiologist Jesús Blázquez of the Spanish National Research Council in Madrid. Researchers have discovered that mutation rates in bacteria sometimes go up in response to stress, in some cases promoting resistance. And studies by Blázquez and others have shown that the antibiotics themselves can cause this phenomenon, called hypermutability”[19].

Although against the particulate gene concept, the above reports are clear evidence of the existence of biological information in the cell itself to bring about necessary mutations at times of need. In the computer model of the organism, the changes in DNA are merely changes in the hardware like the change in any other cell structure. It is the biosoftware of the organism stored on the chromosomes, that brings about these changes or creates new structures (including DNA) so as to provide necessary hardware. In all these cases depending on the stimuli or signals received from the environment, specific biomemes are triggered into operation. Not all organisms will respond similarly to a given stress or environmental condition. An organism can react to an environmental condition only as directed by its biomemome. This would imply that all phenomic changes that occur in an organism are biomemome-directed phenomena *from within the cell* and *not externally induced* as is believed now. These may also be taken as instances of abioprogram-bioprogram interactions. The availability of biomemes to counter environmental stresses including the kind of resistance development is a natural evidence of God's designing the organism to meet the requirement in His scheme of things. These instances illustrate that heritable changes (mutations) that occur in an organism are biosoftware-induced and not by the action of any extraneous mutagen.

REFERENCES

[1] P. A. Wahid, "The Computer Universe: A Scientific Rendering of the Holy Quran," Adam Publishers and Distributors, New Delhi, 2006.

[2] P. A. Wahid, "An Introduction to Islamic Science," Adam Publishers and Distributors, New Delhi, 2007.

[3] K. Zuse, "Rechnender Raum," *Elektronische Datenverarbeitung*, Vol. 8, 1967, pp. 336-344.

[4] J. Schmidhuber, "A Computer Scientist's View of Life, the Universe, and Everything," In: C. Freksa, Ed., *Foundations of Computer Science*: *Potential-Theory-Cognition*, Lecture Notes in Computer Science, Springer, Berlin, 1997, pp. 201-208. http://www.idsia.ch/~juergen

[5] P. A. Wahid, "The Divine Expert System," Centre for Studies on Science, Aligarh, 1998.

[6] S. Lloyd, "Computational Capacity of the Universe," *Physical Review Letters*, Vol. 88, 2002, p. 237901.

[7] S. Lloyd, "Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos," Alfred A. Knopf, New York, 2006.

[8] S. Wolfram, "A New Kind of Science," Wolfram Media, Inc., Champaign, USA, 2002.

[9] D. Bray, "Wetware: A Computer in Every Cell," Yale University Press, 2009.

[10] Retrieved on 2 February 2007. http://www.ftexploring.com/energy/definition.html

[11] R. Dawkins, "The Selfish Gene," Oxford University Press, Oxford, 1976.

[12] W. Johannsen, "The Genotype Conception of Heredity," The American Naturalist, Vol. 45, 1911, pp. 129-159.

[13] R. Falk, "The Gene—A Concept in Tension," In: P. Beurton, R. Falk and H.-J. Rheinberger, Eds., *The Concept of the Gene in Development and Evolution. Historical and Epistemological Perspectives*, Cambridge University Press, Cambridge, 2000, pp. 317-348.

[14] "DNA not the Same in Every Cell of Body: Major Genetic Differences between Blood and Tissue Cells Revealed," *ScienceDaily*, 16 July 2009.

[15] B. Gottlieb, L. E. Chalifour, B. Mitmaker, N. Sheiner, D. Obrand, C. Abraham, M. Meilleur, T. Sugahara, G. Bkaily and M. Schweitzer, "BAK1 Gene Variation and Abdominal Aortic Aneurysms," *Human Mutation*, Vol. 30, No. 7, 2009, p. 1043.

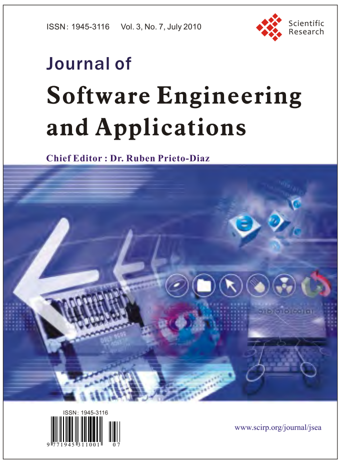
[16] M. Chicurel, "Can Organisms Speed their Own Evolution?" *Science*, Vol. 292, No. 5523, pp.1824-1827.

[17] J. Cairns, J. Overbaugh and S. Miller, "The Origin of Mutants," *Nature*, Vol. 335, 1988, pp. 142-145.

[18] M. A. Kohanski, M. A. DePristo and J. J. Collins, "Sublethal Antibiotic Treatment Leads to Multidrug Resistance via Radical-Induced Mutagenesis," *Molecular Cell*, Vol. 37, No. 3, 2010, pp. 311-320.

[19] M. Enserink, *ScienceNOW Daily News*, 11 February 2010.

Call for Papers


ISSN: 1945-3116 Vol. 3, No. 7, July 2010
Scientific Research
Journal of
Software Engineering and Applications
Chief Editor : Dr. Ruben Prieto-Diaz
ISSN: 1945-3116
www.scirp.org/journal/jsea

Journal of Software Engineering and Applications

ISSN 1945-3116 (print) ISSN 1945-3124 (online)

www.scirp.org/journal/jsea

Journal of Software Engineering and Applications(JSEA) publishes four categories of original technical articles: papers, communications, reviews, and discussions. Papers are well-documented final reports of research projects. Communications are shorter and contain noteworthy items of technical interest or ideas required rapid publication. Reviews are synoptic papers on a subject of general interest, with ample literature references, and written for readers with widely varying background. Discussions on published reports, with author rebuttals, form the fourth category of JSEA publications.

Editor-in-Chief

Dr. Ruben Prieto-Diaz, Universidad Carlos III de Madrid, Spain

Subject Coverage

- Applications and Case Studies
- Artificial Intelligence Approaches to Software Engineering
- Automated Software Design and Synthesis
- Automated Software Specification
- Component-Based Software Engineering
- Computer-Supported Cooperative Work
- Software Design Methods
- Human-Computer Interaction
- Internet and Information Systems Development
- Knowledge Acquisition
- Multimedia and Hypermedia in Software Engineering
- Object-Oriented Technology
- Patterns and Frameworks
- Process and Workflow Management
- Programming Languages and Software Engineering
- Program Understanding Issues
- Reflection and Metadata Approaches
- Reliability and Fault Tolerance
- Requirements Engineering
- Reverse Engineering
- Security and Privacy
- Software Architecture
- Software Domain Modeling and Meta-Modeling
- Software Engineering Decision Support
- Software Maintenance and Evolution
- Software Process Modeling
- Software Reuse
- Software Testing
- System Applications and Experience
- Tutoring, Help and Documentation Systems

Notes for Prospective Authors

Submitted papers should not have been previously published nor be currently under consideration for publication elsewhere. All papers are refereed through a peer review process. For more details about the submissions, please access the website.

Website and E-Mail

Website: http://www.scirp.org/journal/jsea E-Mail: jsea@scirp.org

J. Software Engineering and Applications, **2010, 3, 629-735.**
Published Online July 2010 in SciRes (http://www.SciRP.org/journal/jsea).

TABLE OF CONTENTS

Volume 3 Number 7 July 2010