ISSN: 1945-3116(Print), 1945-3124(Online) Vol. 3, No. 3, March 2010

Journal of Software Engineering and Applications

Chief Editor : Dr. Ruben Prieto-Diaz

www.scirp.org/journal/jsea

Scientific Research

Journal Editorial Board

http://www.scirp.org/journal/jsea

J. Software Engineer & Applications, **2010, 3, 191-302**
Published Online March 2010 in SciRes (http://www.SciRP.org/journal/jsea/).

TABLE OF CONTENTS

Volume 3 Number 3 March 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

PRODUCTION INFORMATION

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

J. Software Engineer & Applications, **2010, 3: 191-197**
doi:10.4236/jsea.2010.33024 Published Online March 2010 (http://www.SciRP.org/journal/jsea)

Linear Control Problems of the Fuzzy Maps

Andrej V. Plotnikov, Tatyana A. Komleva, Irina V. Molchanyuk

Department of Applied Mathematics, Odessa State Academy of Civil Engineering and Architecture, Odessa, Ukraine.
Email: a-plotnikov@ukr.net, t-komleva@ukr.net, i-molchanuyk@ukr.net

Received October 28th, 2009; revised November 16th, 2009; accepted November 20th, 2009.

ABSTRACT

In the present paper, we show the some properties of the fuzzy R-solution of the control linear fuzzy differential inclusions and research the optimal time problems for it.

Keywords*: *Fuzzy Differential Inclusions, Control Problems

1. Introduction

The first study of differential equations with multivalued right-hand sides was performed by A. Marchaud [1] and S. C. Zaremba [2]. In early sixties, T. Wazewski [3,4], A. F. Filippov [5] obtained fundamental results on existence and properties of the differential equations with multivalued right-hand sides (differential inclusions). One of the most important results of these articles was an establishment of the relation between differential inclusions and optimal control problems, that promoted to develop the differential inclusion theory [6–9].

Considering of the differential inclusions required to study properties of multivalued functions, *i.e.* an elaboration the whole tool of mathematical analysis for multivalued functions [6,10,11].

In works [12,13] annotate of an R-solution for differential inclusion is introduced as an absolutely continuous multivalued function. Various problems for the R-solution theory were regarded in [14–18]. The basic idea for a development of an equation for R-solutions (integral funnels) is contained in [19].

In the last years there has been forming new approach to control problems of dynamic systems, which foundation on analysis of trajectory bundle but not separate trajectories. The section of this bundle in any instant is some set and it is necessary to describe the evolution of this set. Obtaining and research dynamic equations of sets there is important problem in this case. The metric space of sets with the Hausdorff metric is natural space for description dynamic of sets. In theory of multivalued maps definitions on derivative as for single-valued maps is impossible because space of sets is nonlinear. This bound possibility description dynamic sets by differential equations. Therefore, the control differential equations with set of initial conditions [20–22] and the control differential inclusions [8,23–34] use for it.

In recent years, the fuzzy set theory introduced by Zadeh [35] has emerged as an interesting and fascinating branch of pure and applied sciences. The applications of fuzzy set theory can be found in many branches of regional, physical, mathematical, differential equations, and engineering sciences. Recently there have been new advances in the theory of fuzzy differential equations [36–47] and inclusions [43,48–52] as well as in the theory of control fuzzy differential equations [53–55] and inclusions [56,57].

In this article we consider the some properties of the fuzzy R-solution of the control linear fuzzy differential inclusions and research the optimal time problems for it.

2. The Fundamental Definitions and Designations

Let $comp\left(R^n\right)\left(conv\left(R^n\right)\right)$ be a set of all nonempty (convex) compact subsets from the space R^n,

$$h(A,B)=\min_{r\geq 0}\left\{S_r(A)\supset B,\ S_r(B)\supset A\right\}$$

be Hausdorff distance between sets A and B, $S_r(A)$ is r-neighborhood of set A.

Let E^n be the set of all u: $R^n\to[0,1]$ such that u satisfies the following conditions:

1) u is normal, that is, there exists an $x_0\in R^n$ such that $u(x_0)=1$;

2) u is fuzzy convex, that is,

$$u\left(\lambda x+(1-\lambda)y\right)\geq\min\left\{u(x),u(y)\right\};$$

3) For any $x, y\in R^n$ and $0\leq\lambda\leq 1$;

4) u is upper semicontinuous;

5) $[u]^0 = cl\{x \in R^n : u(x) > 0\}$ is compact.

If $u \in E^n$, then u is called a fuzzy number, and E^n is said to be a fuzzy number space. For $0 < \alpha \le 1$, denote

$$[u]^\alpha = \{x \in R^n : u(x) \ge \alpha\}.$$

Then from 1)-4), it follows that the α-level set $[u]^\alpha \in conv(R^n)$ for all $0 \le \alpha \le 1$.

Theorem 1. (Negoita and Ralescu [58]). If $u \in E^n$, then

1) $[u]^\alpha \in conv(R^n)$ for all $\alpha \in [0,1]$;

2) $[u]^\alpha \subset [u]^\beta$ for $0 \le \alpha < \beta \le 1$;

3) If $\{\alpha_k\} \subset [0,1]$ is a decreasing sequence converging to $\alpha > 0$ then

$$[u]^\alpha = \bigcap_{k \ge 1} [u]^{\alpha_k}$$

Conversely, if $\{A^\alpha : 0 \le \alpha \le 1\}$ is a family of convex compact subsets of R^n satisfying 1)-3), then $[u]^\alpha = A^\alpha$ for $0 < \alpha \le 1$ and

$$[u]^0 = \overline{\bigcap_{0<\alpha\le 1} A^\alpha} \subset A^0.$$

If $g : R^n \times R^n \to R^n$ is a function, then using Zadeh's extension principle we can extend $\tilde{g}$ to $E^n \times E^n \to E^n$ by the equation

$$\tilde{g}(u,v)(z) = \sup_{z=g(x,y)} \min\{u(x), v(y)\}.$$

It is well known that

$$[\tilde{g}(u,v)]^\alpha = g([u]^\alpha, [v]^\alpha)$$

for all $u, v \in E^n$, $0 \le \alpha \le 1$ and continuous function g. Further, we have

$$[u+v]^\alpha = [u]^\alpha + [v]^\alpha, \quad [ku]^\alpha = k[u]^\alpha,$$

where $k \in R$.

Define $D : E^n \times E^n \to [0,\infty)$ by the relation

$$D(u,v) = \sup_{0\le\alpha\le 1} h([u]^\alpha, [v]^\alpha),$$

where h is the Hausdorff metric defined in $comp(R^n)$. Then D is a metric in E^n.

Further we know that [59]

1) (E^n, D) is a complete metric space;

2) $D(u+w, v+w) = D(u,v)$ for all $u, v, w \in E^n$;

3) $D(\lambda u, \lambda v) = |\lambda| D(u,v)$ for all $u, v \in E^n$ and $\lambda \in R$.

It can be proved that

$$D(u+v, w+z) \le D(u,w) + D(v,z)$$

for $u, v, w, z \in E^n$.

Definition 1. A mapping $F : [0,T] \to E^n$ is strongly measurable if for all $\alpha \in [0,1]$ the set-valued map $F_\alpha : [0,T] \to conv(R^n)$ defined by $F_\alpha(t) = [F(t)]^\alpha$ is Lebesgue measurable.

Definition 2. A mapping $F : [0,T] \to E^n$ is said to be integrably bounded if there is an integrable function $h(t)$ such that $\|x(t)\| \le h(t)$ for every $x(t) \in F_0(t)$.

Definition 3. The integral of a fuzzy mapping $F : [0,T] \to E^n$ is defined levelwise by $\left[\int_0^T F(t)dt\right]^\alpha = \int_0^T F_\alpha(t)dt$ The set of all $\int_0^T f(t)dt$ such that $f : [0,T] \to R^n$ is a measurable selection for F_α for all $\alpha \in [0,1]$.

Definition 4. A strongly measurable and integrably bounded mapping $F : [0,T] \to E^n$ is said to be integrable over $[0,T]$ if $\int_0^T F(t)dt \in E^n$.

Note that if $F : [0,T] \to E^n$ is strongly measurable and integrably bounded, then F is integrable. Further if $F : [0,T] \to E^n$ is continuous, then it is integrable.

Theorem 2. [36]. Let $F, G : [0,T] \to E^n$ be integrable and $c \in [0,T]$, $\lambda \in R$. Then

1) $\int_0^T F(t)dt = \int_0^c F(t)dt + \int_c^T F(t)dt$;

2) $\int_0^T F(t) + G(t)dt = \int_0^T F(t)dt + \int_0^T G(t)dt$;

3) $\int_0^T \lambda F(t)dt = \lambda \int_0^T F(t)dt$;

4) $D(F,G)$ is integrable;

5) $D\left(\int_0^T F(t)dt, \int_0^T G(t)dt\right) \le \int_0^T D(F(t), G(t))dt$

Consider the following control linear fuzzy differential inclusions

$$\dot{x} \in A(t)x + G(t,w), \ x(t_0) = x_0, \qquad (1)$$

and the following nonlinear fuzzy differential inclusions

$$\dot{x} \in F(t,x,w), \quad x(t_0) = x_0, \qquad (2)$$

where $\dot{x}$ means dx/dt; $t \in R_+$ is the time; $x \in R^n$ is the state; $w \in R^m$ is the control; $A(t)$ is $(n \times n)$-dimensional matrix-valued function; $G : R_+ \times R^m \to E^n$, $F : R_+ \times R^n \times R^m \to E^n$ are the set-valued functions.

Let

$$W : R_+ \to conv(R^m) \tag{3}$$

be the measurable multivalued map.

Definition 5. Set LW of all single-valued branches of the multivalued map $W(\cdot)$ is the set of the possible controls.

Obviously, the control fuzzy differential Inclusion (2) turns into the ordinary fuzzy differential inclusion

$$\dot{x} \in \Phi(t, x), \quad x(t_0) = x_0, \tag{4}$$

if the control $\tilde{w}(\cdot) \in LW$ is fixed and $\Phi(t,x) \equiv F(t, x, \tilde{w}(t))$.

The fuzzy differential Inclusions (3) has the fuzzy R-solution, if right-hand side of the fuzzy differential Inclusion (3) satisfies some conditions [52].

Let $X(t)$ denotes the fuzzy R-solution of the differential Inclusion (3), then $X(t, w)$ denotes the fuzzy R-solution of the control differential Inclusion (2) for the fixed $w(\cdot) \in LW$.

Definition 6. The set

$$Y(T) = \{X(T, w): w(\cdot) \in LW\}$$

be called the attainable set of the fuzzy System (2).

3. The Some Properties of the R-Solution

In this section, we consider the some properties of the R-solution of the control fuzzy differential Inclusion (1).

Let the following condition is true.

Condition A:

A1. $A(\cdot)$ is measurable on $[t_0, T]$;

A2. The norm $\|A(t)\|$ of the matrix $A(t)$ is integrable on $[t_0, T]$;

A3. The multivalued map $W : [t_0, T] \to conv(R^m)$ is measurable on $[t_0, T]$;

A4. The fuzzy map $G : R_+ \times R^m \to E^n$ satisfies the conditions

1) measurable in *t*;

2) continuous in *w*;

A5. There exist $v(\cdot) \in L_2[t_0, T]$ and $l(\cdot) \in L_2[t_0, T]$ such that

$$|W(t)| \le v(t), \quad |G(t, w)| \le l(t)$$

almost everywhere on $[t_0, T]$;

A6. The set $Q(t) = \{G(t, w(t)): w(\cdot) \in LW\}$ is compact and convex for almost every $[t_0, T]$, *i.e.* $Q(t) \in conv(E^n)$.

Theorem 3. Let the condition A is true.

Then for every $w(\cdot) \in LW$ there exists the fuzzy R-solution $X(\cdot, w)$ such that

1) the fuzzy map $X(\cdot, w)$ has form

$$X(t, w) = \Phi(t) x_0 + \Phi(t) \int_{t_0}^{t} \Phi^{-1}(s) G(s, w(s)) ds,$$

where $t \in [t_0, T]$; $\Phi(t)$ is Cauchy matrix of the differential equation $\dot{x} = A(t)x$;

2) $X(t, w) \in E^n$ for every $t \in [t_0, T]$;

3) the fuzzy map $X(\cdot, w)$ is the absolutely continuous fuzzy map on $[t_0, T]$.

Proof. The proof is easy consequence of the [31,34,52,54] and Theorem 1.

Theorem 4. Let the condition A is true.

Then the attainable set $Y(T)$ is compact and convex.

Proof. The proof is easy consequence of the [31,34,52,54] and Theorem 1.

We obtained the basic properties of the fuzzy R-solution of System (1). Now, we consider the some control fuzzy problems.

4. The Optimal Time Problems

Consider the control linear fuzzy differential Inclusion (1), when

$$G(t, w) = B(t)w + F(t), \tag{4}$$

where

B1. $B(\cdot)$ is measurable on $[t_0, T]$;

B2. The norm $\|B(t)\|$ of the matrix $B(t)$ is integrable on $[t_0, T]$;

B3. The fuzzy map F: $[t_0,T] \to E^n$ is measurable on $[t_0,T]$;

B4. There exists $f(\cdot) \in L_2[t_0, T]$ such that

$$|F(t)| \le f(t)$$

almost everywhere on $[t_0,T]$.

Consider the following optimal control problem: it is necessary to find the minimal time T and the control $w^*(\cdot) \in LW$ such that the fuzzy R-solution of Systems (1),(4) satisfies one of the conditions:

$$X(T, w^*) \cap S_k \ne \varnothing, \tag{5}$$

$$X(T, w^*) \subset S_k, \tag{6}$$

$$X(T, w^*) \supset S_k, \tag{7}$$

where $S_k \in E^n$ is the terminal set.

Clearly, these time optimal problems are different from the ordinary time optimal problem by that here control object has the volume.

Definition 6. We shall say that the pair $(w^*(\cdot), X(\cdot, w^*))$ satisfies the maximum principle on $[t_0, T]$, if there exists the vector-function $\psi(\cdot)$, which is the solution of the

system

$$\dot{\psi} \in -A^T(t)\psi,\ \psi(T) \in S_1(0)$$

and the following conditions are true

1) the maximum condition

$$C\left(B(t)w^*(t),\psi(t)\right) = \max_{w \in W(t)} C\left(B(t)w,\psi(t)\right)$$

almost everywhere on $[t_0,T]$;

2) the transversal condition:

a) in the case (5):

$$C\left(\left[X(T,w^*)\right]^1,\psi(T)\right) = -C\left(\left[S_k\right]^1,-\psi(T)\right);$$

b) in the case (6): for all $\alpha \in [0,1]$

$$C\left(\left[X(T,w^*)\right]^\alpha,\psi(T)\right) \le C\left(\left[S_k\right]^\alpha,\psi(T)\right)$$

and there exists $\beta \in [0,1]$ such that

$$C\left(\left[X(T,w^*)\right]^\beta,\psi(T)\right) = C\left(\left[S_k\right]^\beta,\psi(T)\right);$$

c) in the case (6): for all $\beta \in [0,1]$

$$C\left(\left[X(T,w^*)\right]^\alpha,-\psi(T)\right) \le C\left(\left[S_k\right]^\alpha,-\psi(T)\right)$$

and there exists $\beta \in [0,1]$ such that

$$C\left(\left[X(T,w^*)\right]^\beta,-\psi(T)\right) = C\left(\left[S_k\right]^\beta,-\psi(T)\right).$$

Clearly, that there cases of the transversal condition of the maximum principle correspond to the three cases of the time optimal problems.

Theorem 5. (necessary optimal condition). Let the condition A are true and the pair $(T,w^*(\cdot))$ is optimality.

Then the pair $(w^*(\cdot),X(\cdot,w^*))$ satisfies the maximum principle on $[t_0,T]$.

Proof. Let $w^*(\cdot)$ is the optimal control and $X(\cdot,w^*)$ is the optimal R-solution of the Systems (1),(4), *i.e.*

1) $X(T,w^*) \in Y(T)$;

2) $X(T,w^*) \cap S_k = \varnothing$.

From 1) and 2) we have

$$\max_{X \in [Y(T)]^1} C(X,\psi) \ge C\left(\left[S_k\right]^1,-\psi\right)$$

for all $\psi \in S_1(0)$.

Consequently

$$p = \max_{X \in [Y(T)]^1} \min_{\psi \in S_1(0)} C(X,\psi) + C\left(\left[S_k\right]^1,-\psi\right) \ge 0.$$

From $\left[X(T,w^*)\right]^1 \cap \left[S_k\right]^1 \ne \varnothing$ we have

$$q(T,\psi) = C\left(\left[X(T,w^*)\right]^1,\psi\right) + C\left(\left[S_k\right]^1,-\psi\right) \ge 0$$

for all $\psi \in S_1(0)$.

From Theorem 1 we have that the function $q(T,\psi)$ is continuous on $R_+ \times S_1(0)$.

If $q(T,\psi) > 0$ for all $\psi \in S_1(0)$ then we have $q^0(T) = \min_{\psi \in S_1(0)} q(T,\psi) \ge \gamma > 0$. Hence there exists $\tau < T$ such that $q^0(\tau) \ge 0$. Consequently we have

$$C\left(\left[X(\tau,w^*)\right]^1,\psi\right) + C\left(\left[S_k\right]^1,-\psi\right) \ge 0$$

for all $\psi \in S_1(0)$, *i.e.* $\left[X(\tau,w^*)\right]^1 \cap \left[S_k\right]^1 \ne \varnothing$.

It contradicts that T is optimal time.

If $p > 0$,

$$\max_{X \in [Y(T)]^1} \min_{\psi \in S_1(0)} C(X,\psi) + C\left(\left[S_k\right]^1,-\psi\right)$$
$$= C\left(\tilde{X},\tilde{\psi}\right) + C\left(\left[S_k\right]^1,-\tilde{\psi}\right)$$

and $\left[X(T,w^*)\right]^1 \ne \tilde{X}$, than we have a contradiction. Hence there exist $\tilde{\psi} \in S_1(0)$ such that

$$C\left(\left[X(T,w^*)\right]^1,\tilde{\psi}\right) = \max_{X \in [Y(T)]^1} C(X,\tilde{\psi}),$$
$$C\left(\left[X(T,w^*)\right]^1,\tilde{\psi}\right) = -C\left(\left[S_k\right]^1,-\tilde{\psi}\right).$$

Consequently

$$\left(\int_0^T \Phi(T)\Phi^{-1}(s)B(s)w^*(s)ds,\tilde{\psi}\right)$$
$$= \max_{w(\cdot) \in LW}\left(\int_0^T \Phi(T)\Phi^{-1}(s)B(s)w(s)ds,\tilde{\psi}\right)$$

Then we have

$$\left(\Phi(T)\Phi^{-1}(s)B(s)w^*(s),\tilde{\psi}\right)$$
$$= \max_{w(\cdot) \in LW}\left(\Phi(T)\Phi^{-1}(s)B(s)w(s),\tilde{\psi}\right)$$

for almost everywhere $s \in [t_0,T]$. If

$$\psi(t) = \left(\Phi(T)\Phi^{-1}(t)\right)^T \tilde{\psi} \Big/ \left\|\left(\Phi(T)\Phi^{-1}(t)\right)^T \tilde{\psi}\right\|,$$

than the theorem is proved.

Example. Consider the following control linear fuzzy differential inclusions

$$\dot{x} \in \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x + w + F,\ x(0) = 0,$$

where $x = (x_1,x_2)^T$ is the state; $w = (w_1,w_2)^T \in W = S_1(0)$ is the control; $F \in E^2$ is the fuzzy set, where

$$\upsilon(f)=\begin{cases}1-4f_1^2-9f_2^2, & 4f_1^2+9f_2^2\le 1\\ 0, & 4f_1^2+9f_2^2>1\end{cases}.$$

Consider the following optimal control problem: it is necessary to find the minimal time T and the control $w^*(\cdot)\in LW$ such that the fuzzy R-solution of system satisfies of the conditions:

$$X(T,w^*)\cap S_k\neq\varnothing$$

where $S_k\in E^2$ is the terminal set such, that

$$\sigma(x)=\begin{cases}\sqrt{1-(x_1-2\pi)^2-(x_2-1)^2}, & x\in Q,\ x_2\ge 1\\ \sqrt{1-(x_1-2\pi)^2} & x\in Q,\ -1<x_2<1\\ \sqrt{1-(x_1-2\pi)^2-(x_2+1)^2} & x\in Q,\ x_2\le -1\\ 0 & x\notin Q\end{cases}$$

$$Q=\left\{\begin{pmatrix}x_1\\ x_2\end{pmatrix}\in R^2:\begin{matrix}2\pi-1\le x_1\le 2\pi+1\\ \sqrt{1-(x_1-2\pi)^2}-1\le x_2\\ \le\sqrt{1-(x_1-2\pi)^2}+1\end{matrix}\right\}.$$

Obviously, the optimal pair $T=2\pi$ and $w^*(t)=(\cos(t),-\sin(t))$ satisfy of the conditions of the Theorem 5:

1) $(w^*(t),\psi(t))=C(W,\psi(t))$ for a.e. $t\in[0,2\pi]$;

2) $C\left(\left[X(T,w^*)\right]^1,\psi(T)\right)=-C\left(\left[S_k\right]^1,-\psi(T)\right)$,

where $\psi(t)=(\cos(t),-\sin(t))^T$ for a.e. $t\in[0,2\pi]$,

$$\left[X(T,w^*)\right]^1=(T\cos(T),-T\sin(T))^T=(2\pi,0)^T,$$

$$\left[S_k\right]^1=\left\{(x_1,x_2)^T:x_1=2\pi,\ -1\le x_2\le 1\right\}.$$

5. Conclusions

In the last decades, a number of works devoted to problems of optimal control of multiple-valued trajectories (fuzzy trajectories, trajectory bundles or an ensemble of trajectories) appeared; these works fall into a subdivision of the optimal control theory, namely, the theory of process control under uncertainty and fuzzy conditions. This is conditioned by the fact that, in actual problems arising in economy and engineering in the course of construction of a mathematical model, it is practically impossible to exactly describe the behavior of an object. This is explained by the following fact. First, for some parameters of the object, it impossible to specify exact values and laws of their change, but it is possible to determine the domain of these changes. Second, for the sake of simplicity of the mathematical model being constructed, the equations that describe the behavior of the object are simplified and one should estimate the consequences of such a simplification. Therefore, if is possible to divide the articles devoted to this direction into two types characterized by the following distinctive features:

1) There exists an incomplete or fuzzy information on the initial data;

2) The equations describing the behavior of the object to be controlled are assumed to be inexact, for example, they can contain some parameters whose exact values and laws of variation are unknown but the domain of their values is fuzzy.

In the second case, fuzzy differential inclusions are frequently used to describe behavior of objects. The reason is that, first this approach is most obvious and, second, theory of fuzzy and ordinary differential inclusions is well found and is rapidly developed at the present time.

In the present paper, the necessary conditions of optimal of control for a system of the latter form of equations with the fuzzy R-solutions are formulated and proved.

REFERENCES

[1] A. Marchaud, "Sur les champs de demicones et equations differentielles du premier order," Bulletin of Mathematical Society, France, No. 62, pp. 1–38, 1934.

[2] S. C. Zaremba, "Sur une extension de la notion d'equation differentielle," Comptes Rendus l'Académie des Sciences, Paris, No. 199, pp. 1278–1280, 1934.

[3] T. Wazewski, "Systemes de commande et equations au contingent," Bulletin L'Académie Polonaise des Science, SSMAP, No. 9, pp. 151–155, 1961.

[4] T. Wazewski, "Sur une condition equivalente e l'equation au contingent," Bulletin L'Académie Polonaise des Science, SSMAP, No. 9, pp. 865–867, 1961.

[5] A. F. Filippov, "Classical solutions of differential equations with multi-valued right-hand side," SIAM Journal of Control, No. 5, pp. 609–621, 1967.

[6] J.-P. Aubin and A. Cellina, "Differential inclusions. Set-valued maps and viability theory," Springer-Verlag, Berlin-Heidelberg-New York-Tokyo, 1984.

[7] N. Kikuchi, "On contingent equations," Japan-United States Seminar on Ordinary Differential and Functional Equations, Lecture Notes in Mathematics, Springer, Berlin, Vol. 243, pp. 169–181, 1971.

[8] V. A. Plotnikov, A. V. Plotnikov, and A. N. Vityuk, "Differential equations with multivalued right-hand sides," Asymptotics Methods, AstroPrint, Odessa, 1999.

[9] G. V. Smirnov, "Introduction to the theory of differential inclusions," Graduate Studies in Mathematics, American Mathematical Society, Providence, Rhode Island, Vol. 41, 2002.

[10] J.-P. Aubin and H. Frankovska, "Set-valued analysis," Birkhauser, Systems and Control: Fundations and Applications, 1990.

[11] F. S. de Blasi and F. IerVolino, "Equazioni differential-icon soluzioni a valore compatto convesso," Bollettino

della Unione Matematica Italiana, Vol. 2, No. 4–5, pp. 491–501, 1969.

[12] A. I. Panasyuk, "Dynamics of sets defined by differential inclusions," Siberian Mathematical Journal, Vol. 27, No. 5, pp. 155–165, 1986.

[13] A. I. Panasyuk, "On the equation of an integral funnel and its applications," Differential Equations, Vol. 24, No. 11, pp. 1263–1271, 1988.

[14] A. I. Panasyuk, "Equations of attainable set dynamics, part 1: Integral funnel equations," Journal of Optimization Theory and Applications, Vol. 64, No. 2, pp. 349–366, 1990. "Equations of attainable set dynamics part 2: Partial differential equations," Journal of Optimization Theory and Applications, Vol. 64, No. 2, pp. 367–377, 1990.

[15] A. I. Panasyuk and V. I. Panasyuk, "Asymptotic optimization of nonlinear control systems," Izdatel Belorussia Gosudarstvo University, Minsk, 1977.

[16] A. I. Panasjuk and V. I. Panasjuk, "An equation generated by a differential inclusion," Matematicheskie Zametki, Vol. 27, No. 3, pp. 429–437, 1980.

[17] A. I. Panasyuk and V. I. Panasyuk, "Asymptotic turnpike optimization of control systems," Nauka i Tekhnika, Minsk, 1986.

[18] A. A. Tolstonogov, "On an equation of an integral funnel of a differential inclusion," Matematicheskie Zametki, Vol. 32, No. 6, pp. 841–852, 1982.

[19] A. I. Panasyuk, "Quasidifferential equations in a metric space," Differentsial'nye Uravneniya, Vol. 21, No. 8, pp. 1344–1353, 1985.

[20] D. A. Ovsyannikov, "Mathematical methods for the control of beams," Leningrad University, Leningrad, 1980.

[21] V. I. Zubov, "Dynamics of controlled systems," Vyssh. Shkola, Moscow, 1982.

[22] V. I. Zubov, "Stability of motion: Lyapunov methods and their application," Vyssh. Shkola, Moscow, 1984.

[23] S. Otakulov, "A minimax control problem for differential inclusions," Soviet Doklady Mathematics, Vol. 36, No. 2, pp. 382–387, 1988.

[24] S. Otakulov, "Approximation of the optimal-time problem for controlled differential inclusions," Cybernetics Systems Analysis, Vol. 30, No. 3, pp. 458–462, 1994.

[25] A. V. Plotnikov, "Linear control systems with multivalued trajectories," Kibernetika, Kiev, No. 4, pp. 130–131, 1987.

[26] A. V. Plotnikov, "Compactness of the attainability set of a nonlinear differential inclusion that contains a control," Kibernetika, Kiev, No. 6, pp. 116–118, 1990.

[27] A. V. Plotnikov, "A problem on the control of pencils of trajectories," Siberian Mathematical Journal, Vol. 33, No. 2, pp. 351–354, 1992.

[28] A. V. Plotnikov, "Two control problems under uncertainty conditions," Cybernet Systems Analysis, Vol. 29, No. 4, pp. 567–573, 1993.

[29] A. V. Plotnikov, "Controlled quasi-differential equations and some of their properties," Differential Equations, Vol. 34, No. 10, pp. 1332–1336, 1998.

[30] A. V. Plotnikov, "Necessary optimality conditions for a nonlinear problems of control of trajectory bundles," Cybernetics and System Analysis, Vol. 36, No. 5, pp. 729–733, 2000.

[31] A. V. Plotnikov, "Linear problems of optimal control of multiple-valued trajectories," Cybernetics and System Analysis, Vol. 38, No. 5, pp. 772–782, 2002.

[32] A. V. Plotnikov and T. A. Komleva, "Some properties of trajectory bunches of controlled bilinear inclusion," Ukrainian Mathematical Journal, Vol. 56, No. 4, pp. 586–600, 2004.

[33] A. V. Plotnikov and L. I. Plotnikova, "Two problems of encounter under conditions of uncertainty," Journal of Applied Mathematics and Mechanics, Vol. 55, No. 5, pp. 618–625, 1991.

[34] V. A. Plotnikov and A. V. Plotnikov, "Multivalued differential equations and optimal control," Applications of Mathematics in Engineering and Economics, Heron Press, Sofia, pp. 60–67, 2001.

[35] L. A. Zadeh, "Fuzzy sets," Information and Control, No. 8, pp. 338–353, 1965.

[36] O. Kaleva, "Fuzzy differential equations," Fuzzy Sets and Systems, Vol. 24, No. 3, pp. 301–317, 1987.

[37] O. Kaleva, "The Cauchy problem for fuzzy differential equations," Fuzzy Sets and Systems, Vol. 35, No. 3, pp. 389–396, 1990.

[38] O. Kaleva, "The Peano theorem for fuzzy differential equations revisited," Fuzzy Sets and Systems, Vol. 98, No. 1, pp. 147–148, 1998.

[39] O. Kaleva, "A note on fuzzy differential equations," Nonlinear Analysis, Vol. 64, No. 5, pp. 895–900, 2006.

[40] T. A. Komleva, L. I. Plotnikova, and A. V. Plotnikov, "Averaging of the fuzzy differential equations," Work of the Odessa Polytechnical University, Vol. 27, No. 1, pp. 185–190, 2007.

[41] T. A. Komleva, A. V. Plotnikov, and N. V. Skripnik, "Differential equations with set-valued solutions," Ukrainian Mathematical Journal, Springer, New York, Vol. 60, No. 10, pp. 1540–1556, 2008.

[42] V. Lakshmikantham, T. G. Bhaskar, and D. J. Vasundhara, "Theory of set differential equations in metric spaces," Cambridge Scientific Publishers, Cambridge, 2006.

[43] V. Lakshmikantham and R. N. Mohapatra, "Theory of fuzzy differential equations and inclusions," Series in Mathematical Analysis and Applications, Taylor & Francis Ltd., London, Vol. 6, 2003.

[44] J. Y. Park and H. K. Han, "Existence and uniqueness theorem for a solution of fuzzy differential equations," International Journal of Mathematics and Mathematical Sciences, Vol. 22, No. 2, pp. 271–279, 1999.

[45] J. Y. Park and H. K. Han, "Fuzzy differential equations," Fuzzy Sets and Systems, Vol. 110, No. 1, pp. 69–77, 2000.

[46] S. Seikkala, "On the fuzzy initial value problem," Fuzzy Sets and Systems, Vol. 24, No. 3, pp. 319–330, 1987.

[47] D. Vorobiev and S. Seikkala, "Towards the theory of fuzzy differential equations," Fuzzy Sets and Systems,

Vol. 125, No. 2, pp. 231–237, 2002.

[48] J.-P. Aubin, "Mutational equations in metric spaces," Set-Valued Analysis, Vol. 1, No. 1, pp. 3–46, 1993.

[49] J.-P. Aubin, "Fuzzy differential inclusions," Problems of Control and Information Theory, Vol. 19, No. 1, pp. 55–67, 1990.

[50] V. A. Baidosov, "Differential inclusions with fuzzy right-hand side," Soviet Mathematics, Vol. 40, No. 3, pp. 567–569, 1990.

[51] V. A. Baidosov, "Fuzzy differential inclusions," Journal of Applied Mathematics and Mechanics, Vol. 54, No. 1, pp. 8–13, 1990.

[52] E. Hullermeier, "An approach to modeling and simulation of uncertain dynamical systems," International Journal of Uncertainty, Fuzziness Knowledge-Based Systems, Vol. 5, No. 2, pp. 117–137, 1997.

[53] N. D. Phu and T. T. Tung, "Some properties of sheaf-solutions of sheaf fuzzy control problems," Electronic Journal of Differential Equations, No. 108, pp. 1–8, 2006. http://www.ejde.math.txstate.edu.

[54] N. D. Phu and T. T. Tung, "Some results on sheaf-solutions of sheaf set control problems," Nonlinear Analysis, Vol. 67, No. 5, pp. 1309–1315, 2007.

[55] N. D. Phu and T. T. Tung, "Existence of solutions of fuzzy control differential equations," Journal of Sci-Tech Development, Vol. 10, No. 5, pp. 5–12, 2007.

[56] I. V. Molchanyuk and A. V. Plotnikov, "Linear control systems with a fuzzy parameter," Nonlinear Oscillator, Vol. 9, No. 1, pp. 59–64, 2006.

[57] V. S. Vasil'kovskaya and A. V. Plotnikov, "Integro-differential systems with fuzzy noise," Ukrainian Mathematical Journal, Vol. 59, No. 10, pp. 1482–1492, 2007.

[58] C. V. Negoito and D. A. Ralescu, "Applications of fuzzy sets to systems analysis," A Halsted Press Book, John Wiley & Sons, New York-Toronto, Ont., 1975.

[59] M. L. Puri and D. A. Ralescu, "Fuzzy random variables," Journal of Mathematical Analysis and Applications, No. 114, pp. 409–422, 1986.

J. Software Engineering & Applications, 2010, 3: 198-207
doi:10.4236/jsea.2010.33025 Published Online March 2010 (http://www.SciRP.org/journal/jsea)

Scientific Research

Incremental Computation of Success Patterns of Logic Programs

Lunjin Lu

Department of Computer Science and Engineering, Oakland University, Rochester, USA.
Email: lunjin@acm.org

Received October 28th, 2009; revised November 26th, 2009; accepted November 30th, 2009.

ABSTRACT

A method is presented for incrementally computing success patterns of logic programs. The set of success patterns of a logic program with respect to an abstraction is formulated as the success set of an equational logic program modulo an equality theory that is induced by the abstraction. The method is exemplified via depth and stump abstractions. Also presented are algorithms for computing most general unifiers modulo equality theories induced by depth and stump abstractions.

***Keywords*:** *Incremental Analysis, Success Patterns, Abstract Interpretation, Depth Abstract, Stump Abstraction, Logic Programs*

1. Introduction

In abstract interpretation, program analyses are viewed as program execution over non-standard data domains. Cousot and Cousot first laid solid logical foundations for abstract interpretations [1,2]. Their idea is to define a collecting semantics for a program which associates each program point with the set of all storage states that are possibly obtained when the execution reaches the point. In practice, an abstraction of the collecting semantics is calculated by simulating over a non-standard data domain the computation of the collecting semantics over the standard data domain. The standard data domain and the non-standard domain are called the concrete domain and the abstract domain respectively.

Abstract interpretation has been used to perform various analyses of logic programs such as occur check analysis [3], mode analysis [4–6], sharing analysis [7,8] and type analysis [6,9,10]. Further more, a number of abstract interpretation frameworks for logic programs have been brought about [5], Jones *et al.* [11], Bruynooghe [9] and Marriott *et al.* [12]. With an abstract interpretation framework, the design of a particular analysis reduces to the design of an abstract domain and a number of abstract operations on the abstract domain. The safeness of the analysis is verified by formalizing a correspondence between the concrete domain and the abstract domain and proving that the abstract operations safely simulate the concrete operations with respect to the correspondence. The correspondence between the abstract domain and the concrete domain can be formalized either as an abstraction (function) from the concrete domain to the abstract domain, or as a concretization (function) from the abstract domain to the concrete domain, or as a joined pair of abstraction and concretization, or as a relation between the concrete domain and the abstract domain. We assume that the correspondence is given as a surjective abstraction from the domain of concrete terms into a domain of abstract terms[1].

A program analysis is currently performed with respect to a fixed abstraction; and different analyses corresponding to different abstractions are performed separately even when there is a strong relationship between them. Take depth abstractions for example, a depth 3 analysis will be performed separately from a depth 2 analysis even if the result of the depth 2 analysis can be used to perform the depth 3 analysis, as we will show later in this paper. This paper is concerned with refining program analyses whereby the result of a coarser analysis corresponding to a stronger abstraction is used to obtain a finer analysis corresponding to a weaker abstraction. In particular, we are concerned with obtaining finer success patterns of a logic program from coarser success patterns of the same program. We introduce an ordering relation on abstractions of terms. We then argue, for a class of

[1]In case an abstraction is not surjective, we can always construct a new of abstract terms by eliminating those abstract terms that are not images of any concrete term under the abstraction. The abstraction is a surjective abstraction from the domain of concrete terms to the new domain of abstract terms.

abstractions, that the set of success patterns of a logic program P with respect to an abstraction α is tantamount to the success set of the equational logic program $P \cup E_\alpha$ where E_α is an equality theory induced by α. Therefore, either the fixpoint semantics or the procedural semantics defined for equational logic programs can be used to compute success patterns of logic programs. From this observation, the success patterns of a logic program P can be computed by incremental refinement. A set of coarser success patterns of P relative to a stronger abstraction α_1 can be obtained by computing the fixpoint semantics of the equational logic program $P \cup E_{\alpha_1}$. If the success patterns are not fine enough for the application at hand, candidates for finer success patterns relative to a weaker abstraction α_2 can be generated from the coarser success patterns and verified by using either the procedural or the fixpoint semantics of equational logic program $P \cup E_{\alpha_2}$. This refinement process is repeated until success patterns are fine enough for the application.

The remainder of this paper is organized as follows. Section 2 presents a fixed-point and a procedural abstract semantics of logic programs for a class of abstractions and lays a foundation for incremental refinement of success patterns of logic programs with respect to that class of abstractions. Sections 3 and 4 devote to incremental refinement of success patterns of logic programs for depth abstractions and stump abstractions respectively. In Section 5, we conclude this paper with a summary of the paper and some points to future work in analysis refinement.

2. A Foundation for Incremental Refinement

Let $\Sigma, \Pi, Vars$ be respectively a set of function symbols, a set of predicate symbols and a denumerable set of variables. $Term(\Sigma, V)$ denotes the set of terms constructible from Σ and V, and $Atom(\Pi, S)$ denotes the set of atoms constructible from Π and S where S is a set of terms. The Herbrand universe $\mathcal{HU}$ are theHerbrand base $\mathcal{HB}$ of a logic program P are

$$\mathcal{HU} = Term(\Sigma, \varnothing)$$

and

$$\mathcal{HB} = Atom(\Pi, \mathcal{HU})$$

respectively. Let $Term = Term(\Sigma, Vars)$. Let $Term^\alpha$ be a set of abstract terms, and α be an abstraction from $Term$ to $Term^\alpha$. α induces an equivalence relation $\approx_\alpha$ on $Term$, $(t_1 \approx_\alpha t_2) = (\alpha(t_1) = \alpha(t_2))$. So, abstract terms in $Term^\alpha$ is identified with equivalence classes of $\approx_\alpha$. That is, $Term^\alpha = Term_{/\approx_\alpha}$. α is called stable if $\forall t, s \in Term. \forall \theta \in Sub. ((t \approx_\alpha s) \rightarrow (t\theta \approx_\alpha s\theta))$ where Sub is the set of substitutions. Let $E_\alpha = \{\approx_\alpha\}$. E_α is an equality theory on $Term$. We extend α to an abstraction from $Atom(\Pi, Term)$ to $Atom(\Pi, Term^\alpha)$ as follows: $\alpha(p(t_1, \cdots, t_n)) = p(\alpha(t_1), \cdots, \alpha(t_n))$. $\approx_\alpha$ and E_α are extended accordingly.

Let $t, s \in Term(or\ Atom)$ and $\sigma, \theta \in Sub$. σ is an E_α-unifier of t and s if $t\sigma \approx_\alpha s\sigma$. t and s are E_α-unifiable if they have one or more E_α-unifiers. σ is more general than θ with respect to E_α, denoted as $\sigma \leq_{E_\alpha} \theta$, iff there is an $\eta \in Sub$ such that $X\sigma\eta \approx_\alpha X\theta$ for all $X \in Vars$. An E_α-unifier σ of t and s is a maximally general E_α-unifier (E_α-mgu) of t and s iff, for any other E_α-unifier θ of t and s, $\theta \not\leq_{E_\alpha} \sigma$.

2.1 Fixpoint and Procedural Abstract Semantics

This section presents a fixpoint and a procedural abstract semantics of a definite logic program P with respect to a stable abstraction α. It is well known that the success set of P is tantamount to the least fixpoint of the following function $\mathbf{T}: \wp(\mathcal{HB}) \mapsto \wp(\mathcal{HB})$ by van Emden and Kowalski [13].

$$\mathbf{T}(I) = \{H\sigma : \exists \sigma. \exists H \leftarrow B_1, \cdots, B_m \in P. \\ B_1\sigma \in I \wedge \cdots \wedge B_m\sigma \in I\} \quad (1)$$

For any logic program P and any abstraction α, $P \cup E_\alpha$ is an equational logic program. The fixpoint semantics of $P \cup E_\alpha$ given by Jaffar *et al.* [14] is

$$\mathbf{T}^\alpha \uparrow \omega \quad \text{with} \quad \mathbf{T}^\alpha : \wp(\mathcal{HB}_{/\approx_\alpha}) \mapsto \wp(\mathcal{HB}_{/\approx_\alpha})$$

being defined as follows.

$$\mathbf{T}^\alpha(I^\#) = \{[H\sigma]_{\approx_\alpha} : \exists \sigma. \exists H \leftarrow B_1, \cdots, B_m \in P. \\ [B_1\sigma]_{\approx_\alpha} \in I^\# \wedge \cdots \wedge [B_m\sigma]_{\approx_\alpha} \in I^\#\} \quad (2)$$

According to Jaffar *et al.* [14],

$$(P \cup E_\alpha \models A) \leftrightarrow ([A]_{\approx_\alpha} \in \mathbf{T}^\alpha \uparrow \omega)$$

for any $A \in \mathcal{HB}$. We adopt $\mathbf{T}^\alpha \uparrow \omega$ as the fixpoint abstract semantics of P relative to α. The following lemma states the $\mathbf{T}^\alpha \uparrow \omega$ is a safe approximation of $\mathbf{T} \uparrow \omega$ with respect to α.

Lemma 1 If α is a stable abstraction then $\forall A \in \mathcal{HB}. (A \in \mathbf{T} \uparrow \omega \rightarrow [A]_{\approx_\alpha} \in \mathbf{T}^\alpha \uparrow \omega)$.

The procedural semantics of an equational logic program $P \cup E_\alpha$ is the equational *SLD* resolution with

respect to the equality theory E_α, denoted as SLD_α. SLD_α plays same role for $P \cup E_\alpha$ as *SLD* for P. SLD_α differs from *SLD* in the sense that, in SLD_α, E_α-unification plays the role of normal unification in *SLD*. In the following, we adapt SLD_α so that it works on equivalence classes of $\approx_\alpha$ on *Atom*. Define $[t]_{\approx_\alpha}\theta = [t\theta]_{\approx_\alpha} = \alpha(t\theta)$. Notice that equivalence classes of terms (resp. atoms) are identified with abstract terms (resp. abstract atoms). The application of a substitution θ to an equivalence class $[t]_{\approx_\alpha}$ can accomplished by applying θ to any term t' in $[t]_{\approx_\alpha}$ taking $[t'\theta]_{\approx_\alpha}$ as the result because of the stability of α which also allows us to define an E_α-*mgu* of $[t]_{\approx_\alpha}$ and $[s]_{\approx_\alpha}$ as an E_α-*mgu* of t and s. The basic step in SLD_α can now be defined as follows.

Definition 1 Let $G^\# \equiv \leftarrow A_1^\#, \cdots, A_j^\#, \cdots, A_p^\#$ and $C \equiv H \leftarrow B_1, \cdots, B_q$ be a variant of a clause of P. $W^\#$ is called E_α-derived from $G^\#$ and C using E_α-mgu σ if (1) σ is an E_α-mgu of $A_j^\#$ and $\alpha(H)$; and (2) $W^\# = \leftarrow A_1^\#\sigma, \cdots, A_{j-1}^\#\sigma, \alpha(B_1)\sigma, \cdots, \alpha(B_q)\sigma, A_{j+1}^\#\sigma, \cdots, A_p^\#\sigma$.

It is proven by Jaffar *et al.* [14] that

$$(P \cup E_\alpha \vdash A) \leftrightarrow (P \rightarrow_{SLD_\alpha} [A]_{\approx_\alpha})$$

where $P \rightarrow_{SLD_\alpha} [A]_{\approx_\alpha}$ denotes that $[A]_{\approx_\alpha}$ is provable from P using SLD_α. This implies that $\rightarrow_{SLD_\alpha}$ can be used to verify whether an abstract atom $[A]_{\approx_\alpha}$ is a success pattern of P with respect to α according to lemma 1. In summary,

$$(P \cup E_\alpha \vdash A) \leftrightarrow ([A]_{\approx_\alpha} \in \mathbf{T}^\alpha \uparrow \omega) \leftrightarrow (P \rightarrow_{SLD_\alpha} [A]_{\approx_\alpha}) \quad (3)$$

2.2 Foundation for Incremental Refinement

Let α_1 and α_2 be two abstractions. Define $\alpha_1 \subseteq \alpha_2$ iff $t \approx_{\alpha_1} s \rightarrow t \approx_{\alpha_2} s$ for all $t, s \in Term$. When $\alpha_1 \subseteq \alpha_2$, we say that α_1 is weaker or finer than α_2 and that α_2 is stronger or coarser than α_1. Note that if $\alpha_1 \subseteq \alpha_2$ then $[t]_{\approx_{\alpha_1}} \subseteq [t]_{\approx_{\alpha_2}}$ for any $t \in Term$. In other words, $\approx_{\alpha_1}$ is a finer partition on *Term* (and *Atom*) than $\approx_{\alpha_2}$. If $\alpha_1 \subseteq \alpha_2$ then $E_{\alpha_1} \vdash E_{\alpha_2}$. Therefore, we have

$$(\alpha_1 \subseteq \alpha_2) \rightarrow ((P \cup E_{\alpha_1}) \vdash (P \cup E_{\alpha_2})) \quad (4)$$

By Equations (3) and (4),

$$(\alpha_1 \subseteq \alpha_2) \rightarrow \forall A \in \mathsf{HB}.(([A]_{\approx_{\alpha_1}} \in \mathbf{T}^{\alpha_1} \uparrow \omega) \rightarrow ([A]_{\approx_{\alpha_2}} \in \mathbf{T}^{\alpha_2} \uparrow \omega)) \quad (5)$$

Equation (5) lays a foundation for incremental refinement of success patterns of logic programs. An initial set of the success patterns of a logic program P can be obtained by computing $\mathbf{T}^\alpha \uparrow \omega$ which is a safe approximation of $\mathbf{T} \uparrow \omega$ relative to α. If the success patterns in $\mathbf{T}^\alpha \uparrow \omega$ are not finer enough for the application at hand then finer success patterns can be computed by a generate-and-test approach as follows. Firstly, a weaker abstraction α' is formed and candidates elements for $\mathbf{T}^{\alpha'} \uparrow \omega$ are generated from $\mathbf{T}^\alpha \uparrow \omega$. The formation of α' and generation of candidates elements for $\mathbf{T}^{\alpha'} \uparrow \omega$ can be done by splitting one or more equivalence classes of $\approx_\alpha$. Secondly, $SLD_{\alpha'}$ is used to verify if a particular candidate element is in $\mathbf{T}^{\alpha'} \uparrow \omega$. This process of refinement is repeated until success patterns are fine enough.

If $\alpha' \subseteq \alpha$, $[A]_{\approx_{\alpha'}} \subseteq [A]_{\approx_\alpha}$ for any $A \in \mathcal{HB}$, *i.e.*, the $\approx_{\alpha'}$ equivalence class including A is contained in the $\approx_\alpha$ equivalence class including A. Let $R_{\alpha,\alpha'}$ be a refinement operator that splits an $\approx_\alpha$ equivalence class C into the set of $\approx_{\alpha'}$ equivalence classes contained in C.

$$R_{\alpha,\alpha'}(C) = \{[A]_{\approx_{\alpha'}} : A \in \mathsf{HB} \wedge [A]_{\approx_\alpha} = C\}$$

Then candidates elements for $\mathbf{T}^{\alpha'} \uparrow \omega$ can be generated from $\mathbf{T}^\alpha \uparrow \omega$ by applying $R^*_{\alpha,\alpha'}$ to $\mathbf{T}^\alpha \uparrow \omega$ where $R^*_{\alpha,\alpha'}$ is defined $R^*_{\alpha,\alpha'}(S) = \bigcup_{c \in s} R_{\alpha,\alpha'}(C)$.

For a given set S of $\approx_\alpha$ equivalence classes, $R^*_{\alpha,\alpha'}$ returns the union of the sets of $\approx_{\alpha'}$ equivalence classes resulting from applying $R_{\alpha,\alpha'}$ to $\approx_\alpha$ equivalence classes in S.

2.3 An Example of Incremental Refinement

We illustrate the idea of incremental refinement of success patterns of logic programs by means of depth abstractions proposed by Sato *et al.* [15]. A depth abstraction partitions *Term* into a finite number of equivalent classes. Two terms belong to the same class iff their term trees are identical to a certain depth *n*, called the depth of abstraction. For example, $h(f(a), g(b))$ is equivalent to $h(f(b), g(a))$ to depth 2. Let d_n denote depth n abstraction. $d_n(t)$ replaces each sub-term of t at depth n with a _ that denotes any

term. Letting $p \in \Pi$, we have

$$d_1(p(f(a),g(b)) = d_1(p(f(b),g(a)) = p(f(\_),g(\_)).$$

Deferring a formal presentation of depth abstractions until Section 3, we now show how SLD_α can be used when it is necessary to increase the depth of abstraction.

Example 1 Let $\alpha = d_1$ and P={$a(f(c))$. $b(f(h(c)))$. $p(x) \leftarrow a(x), b(x)$. }. We have $\mathbf{T}^{d_1} \uparrow 0 = \varnothing$,

$$\mathbf{T}^{d_1} \uparrow 1 = \{a(f(\_)), b(f(\_))\},$$

$$\mathbf{T}^{d_1} \uparrow 2 = \{a(f(\_)), b(f(\_)), p(f(\_))\},$$

$$\mathbf{T}^{d_1} \uparrow 3 = \{a(f(\_)), b(f(\_)), p(f(\_))\},$$

and

$$\mathbf{T}^{d_1} \uparrow \omega = \mathbf{T}^{d_1} \uparrow 3 = \{a(f(\_)), b(f(\_)), p(f(\_))\}.$$

Suppose now we want to be more precise and decide to compute $\mathbf{T}^{d_2} \uparrow \omega$. Note that the set of ground atoms that $\mathbf{T}^{d_2} \uparrow \omega$ approximates is a subset of the set of ground atoms that $\mathbf{T}^{d_1} \uparrow \omega$ approximates. Instead of computing the least fixpoint of $\mathbf{T}^{d_2}$, we compute $\mathbf{T}^{d_2} \uparrow \omega$ by a generate-and-test approach. We first generate a set of candidate elements for $\mathbf{T}^{d_2} \uparrow \omega$ and then use SLD_{d_2} resolution to eliminate false candidates. The generation of candidates is accomplished by applying the refinement operator R_{d_1,d_2} defined in Section 3 to elements in $\mathbf{T}^{d_1} \uparrow \omega$. For each element in $\mathbf{T}^{d_1} \uparrow \omega$, R_{d_1,d_2} generates a set of candidates by substituting each occurrence of _ with every element from $\mathcal{HU}_{/\approx_{d_1}} = \{c, f(\_), h(\_)\}$. Thus, the set of candidates is

$$\{a(f(c)), a(f(f(\_))), a(f(h(\_))), b(f(c)), b(f(f(\_))),$$
$$b(f(h(\_))), p(f(c)), p(f(f(\_))), p(f(h(\_)))\}.$$

After eliminating candidates that are not provable from P using SLD_{d_2},we have

$$\mathbf{T}^{d_2} \uparrow \omega = \{a(f(c)), b(f(h(\_)))\}.$$

$p(f(c))$ has been eliminated as follows. First, $\leftarrow p(f(c))$ is resolved with the clause $p(x) \leftarrow a(x), b(x)$ resulting in $\leftarrow a(f(c)), b(f(c))$.Then goal $\leftarrow a(f(c))$ is resolved with the unit clause $a(f(c))$. However, $\leftarrow b(f(c))$ cannot be resolved with b(f(h(c))) because $d_2(b(f(c)))$= $d_2(b(f(c)))$ while $d_2(b(f(h(c)))) = b(f(h(\_)))$.

The following two sections demonstrate incremental refinement of success patterns of logic programs by considering two families of abstractions, namely depth abstractions and stump abstractions.

3. Depth Abstractions

The idea of enumerating success patterns of logic programs to a certain depth is due to Sato and Tamaki [15]. Depth abstraction has been used to ensure termination of an analysis, e.g. [10,16,17]. All terms (resp. atoms) identical to a certain depth are considered equivalent. For example, both $f(a,g(h(0),1),b)$ and $f(a,g(2,h(h(0))),b)$ have main functor $f/3$ and the first and the third of their arguments are same. Both of their second arguments have $g/2$ as main functor. If this information is enough, then we can use either $f(a,g(h(0),1),b)$ or $f(a,g(2,h(h(0))),b)$ as a representative of them. Since we are not interested in the arguments of $g/2$ we shall replace each argument of $g/2$ with a special symbol _, denoting any term, that is, we use $f(a,g(\_,\_),b)$ to represent both $f(a,g(h(0),1),b)$ and $f(a,g(2,h(h(0))),b)$. $f(a,g(\_,\_),b)$ actually represents an infinite number of terms.

This section defines depth abstractions, constructs a refinement operator and an equational unification algorithm for such abstractions, and exemplifies incremental refinement of success patterns with respect to depth abstractions.

3.1 Depth Abstractions

Let $t = f(t_1,\cdots,t_m)$ be a term. Then t is a depth 0 sub-term of t, and a term s is a depth k sub-term of t if s is a depth $(k-1)$ sub-term of t_i for some $1 \le i \le m$.

Definition 2 Let t be a term. The depth k abstraction of t, denoted by $d_k(t)$, is obtained by replacing each depth k sub-term of t with an _.

$$\begin{aligned} d_k(t) &= \_ \quad k = 0 \\ d_k(f(t_1,\cdots,t_m)) &= f(d_{k-1}(t_1),\cdots,d_{k-1}(t_m)) \quad k > 0 \end{aligned}$$

For instance, the depth 2 abstraction of $f(g(X,Y), g(h(Z)))$ is $f(g(\_,\_),g(\_))$, and its depth 3 abstraction is $f(g(X,Y),g(h(\_)))$.

Lemma 2 For any $k \ge 0$, d_k is stable.

3.2 A Refinement Operator for Depth Abstractions

Let $t^\#$ be an abstract term denoting an $\approx_{d_{k-1}}$ equivalence class.

$$\tilde{d} : Term(\Sigma \cup \{\_\}, \varnothing) \mapsto \wp(Term(\Sigma \cup \{\_\}, \varnothing))$$

defined below splits $t^\#$ by replacing each _ in $t^\#$ with an abstract term from $\mathcal{HU}_{/\approx_{d_1}}$ in every possible way.

$$\tilde{d}(\_) = \{f(\_,\cdots,\_) \mid f \in \Sigma\}$$

$$\tilde{d}(g(t_1,\cdots,t_m)) = \{g(s_1,\cdots,s_m) \mid \forall 1 \le j \le m.s_j \in d(t_j)\}$$

Its extension yields a refinement operator

$$\tilde{d} : Atom(\Pi, Term(\Sigma \cup \{\_\}, \varnothing)))$$

$$\mapsto \wp(Atom(\Pi, Term(\Sigma \cup \{\_\}, \varnothing)).$$

$$\tilde{d}(p(t_1,\cdots,t_n)) = \{p(s_1,\cdots,s_n) \mid \forall 1 \le j \le n.s_j \in \tilde{d}(t_j)\}$$

$$\tilde{d}^* : \wp(Term(\Sigma \cup \{\_\}, \varnothing)) \mapsto \wp(Term(\Sigma \cup \{\_\}, \varnothing))$$

is the extension of $\tilde{d}$ to sets of abstract atoms.

$$\tilde{d}^*(S) = \bigcup_{A^{\#} \in S} \tilde{d}(A^{\#})$$

Lemma 3 If $\Sigma \neq \varnothing$ then $R_{d_k, d_{k+1}} = \tilde{d}$ and $R^*_{d_k, d_{k+1}} = \tilde{d}^*$ for any $k \ge 0$.

3.3 An E_{d_k}-Unification Algorithm

Now we present an E_{d_k}-unification algorithm and prove its correctness. The following algorithm for E_{d_k}-unification results from modifying Robinson's unification algorithm [18]. Function $occur(k, X, t)$ is true iff X occurs in t at any depth $j < k$.

Algorithm 1 *This algorithm decides if* t_1 *and* t_2 *are* E_{d_k}*-unifiable and, if* E_{d_k}*-unifiable, returns an* E_{d_k}*-mgu of* t_1 *and* t_2.

```
01  function Dunify(k, t1, t2) ⇒ (unifiable, σ)
02  begin
03  if k = 0 then (unifiable, σ) ← (true, ∅)
04      else if t1 or t2 is a variablethen
05      begin let X be the variable and t the other term
06          if X = t then (unifiable, σ) ← (true, ∅)
07          else if occur(k, X, t) then
(unifiable, σ) ←Dunify(k, X, t{X ↦ t})
08          else (unifiable, σ) ← (true, {X ↦ d_k(t)})
09      end else
10      begin let t1 = f(x1,…,xn) and t2 = g(x1,…,xm)
11          if f ≠ g or m ≠ n then unifiable← false
                else
12          begin j ← 0, (unifiable, σ_0) ← (true, ∅)
13              while j < m and unifiable do
14              begin j ← j +1
15                  (unifuable,τ_j)←Dunify (k-1,x_jσ_{j-1},y_jσ_{j-1})
16                  if unifiable then σ_j ← σ_{j-1}τ_j
17              end
18          σ ← σ_m
19      end
20      end
21      return (unifiable, σ)
22  end
```

The line 07 in algorithm 1 deals with E_{d_k}-unification of X and t where X occurs in t at some depth $j < k$. This does not necessarily mean failure of the E_{d_k}-unification of X and t. For instance, $\theta = \{X \mapsto f(Y)\}$ is a E_{d_1}-*mgu* of X and $f(X)$. Algorithm 1 reduces the problem of E_{d_k}-unification of X and t into the problem of E_{d_k}-unification of X and $t\{X \mapsto t\}$.

Lemma 4 If two terms t_1 and t_2 are E_{d_k}-unifiable, then algorithm 1 terminates and gives a unique (module renaming) E_{d_k}-mgu of t_1 and t_2. Otherwise, the algorithm terminates and reports the fact.

3.4 Refinement of Success Patterns for Depth Abstractions

All depth abstractions are comparable with respect to $\subseteq$. Abstractions corresponding to bigger depths are finer than those corresponding to smaller depths. Formally,

Lemma 5 For any $0 \le j \le k$, $d_k \subseteq d_j$.

Lemma 5 implies that, for any $A \in \mathcal{HB}$, if $[A]_{\approx_{d_k}} \in \mathbf{T}^{d_k} \uparrow \omega$ then $[A]_{\approx_{d_{k-1}}} \in \mathbf{T}^{d_{k-1}} \uparrow \omega$. This enables us to refine success patterns of P by increasing abstraction depth. Suppose that success patterns in $\mathbf{T}^{d_{k-1}} \uparrow \omega$ are not fine enough and it is necessary to compute $\mathbf{T}^{d_k} \uparrow \omega$. Rather than throwing away $\mathbf{T}^{d_{k-1}} \uparrow \omega$ and computing $\mathbf{T}^{d_k} \uparrow \omega$ from scratch, we compute $\mathbf{T}^{d_k} \uparrow \omega$ by

1) applying $\tilde{d}^*$ to $\mathbf{T}^{d_{k-1}} \uparrow \omega$ resulting in a set of candidate elements for $\mathbf{T}^{d_k} \uparrow \omega$ since $\tilde{d}^*(\mathbf{T}^{d_{k-1}} \uparrow \omega) \supseteq \mathbf{T}^{d_k} \uparrow \omega$;

2) applying SLD_{d_k} to eliminate those candidate elements that are not provable from P using SLD_{d_k}.

The following two examples illustrate incremental refinement of success patterns of logic programs with respect to depth abstractions.

Example 2 Let

$$P = \{p(a,b), p(X,Y) \leftarrow q(X,Y), q(a,b), q(r(X), s(Y)) \leftarrow q(X,Y)\}$$

We have

$\mathbf{T}^{d_1} \uparrow 0 = \varnothing$

$\mathbf{T}^{d_1} \uparrow 1 = \{p(a,b), q(a,b)\}$

$\mathbf{T}^{d_1} \uparrow 2 = \{p(a,b), q(a,b), q(r(\_), s(\_))\}$

$\mathbf{T}^{d_1} \uparrow 3 = \{p(a,b), q(a,b), q(r(\_), s(\_)), p(r(\_), s(\_))\}$

$\mathbf{T}^{d_1} \uparrow 4 = \{p(a,b), q(a,b), q(r(\_), s(\_)), p(r(\_), s(\_))\}$

So,

$\mathbf{T}^{d_1} \uparrow \omega = \mathbf{T}^{d_1} \uparrow 4 = \{p(a,b), q(a,b), q(r(\_), s(\_)), p(r(\_), s(\_))\}$

Example 3 Let P be the same as example 2 and suppose that success patterns in $\mathbf{T}^{d_1} \uparrow \omega$ are not fine enough. We compute $\mathbf{T}^{d_2} \uparrow \omega$ as follows. We first apply $\tilde{d}^*$ to $\mathbf{T}^{d_1} \uparrow \omega$ resulting in the following candidate elements for $\mathbf{T}^{d_2} \uparrow \omega$.

$$p(a,b), q(a,b), q(r(a), s(a)), q(r(a), s(b)), q(r(a), s(r(\_))),$$
$$q(r(a), s(s(\_))), q(r(b), s(a)), q(r(b), s(b)), q(r(b), s(r(\_))),$$
$$q(r(b), s(s(\_))), q(r(r(\_)), s(a)), q(r(r(\_)), s(b)), q(r(r(\_)), s(r(\_))),$$
$$q(r(r(\_)), s(s(\_))), q(r(s(\_)), s(a)), q(r(s(\_)), s(b)), q(r(s(\_)), s(r(\_))),$$
$$q(r(s(\_)), s(s(\_))), p(r(a), s(a)), p(r(a), s(b)), p(r(a), s(r(\_))),$$
$$p(r(a), s(s(\_))), p(r(b), s(a)), p(r(b), s(b)), p(r(b), s(r(\_))), p(r(b), s(s(\_))),$$
$$p(r(r(\_)), s(a)), p(r(r(\_)), s(b)), p(r(r(\_)), s(r(\_))), p(r(r(\_)), s(s(\_))),$$
$$p(r(s(\_)), s(a)), p(r(s(\_)), s(b)), p(r(s(\_)), s(r(\_))), p(r(s(\_)), s(s(\_)))$$

We then apply SLD_{d_2} to eliminate those candidate elements that are not provable from P by using SLD_{d_2}, we have

$\mathbf{T}^{d_2} \uparrow \omega$ = p(a,b),q(a,b),q(r(a)),s(b)),q(r(r(_)),s(s(_))), p(r(a),s(b)),p(r(r(_)),s(s(_)))

$q(r(r(\_)), s(s(\_)))$ has not been removed because it is provable from P by using SLD_{d_2}. The SLD_{d_2}-refutation process is as follows.

$G_0 = \leftarrow q(r(r(\_)), s(s(\_)))$

$$\begin{cases} C_0 = q(r(X1), s(Y1)) \leftarrow q(X1, Y1) \\ \sigma_0 = \{X1/r(X2), Y1/s(Y2)\} \end{cases}$$

$G_1 = \leftarrow q(r(X2), s(Y2))$

$$\begin{cases} C_1 = q(r(X3), s(Y3)) \leftarrow q(X3, Y3) \\ \sigma_1 = \{X3/X3, Y3/Y2\} \end{cases}$$

$G_2 = \leftarrow q(X2, Y2)$

$$\begin{cases} C_2 = q(a,b) \\ \sigma_2 = \{X2/a, Y2/b\} \end{cases}$$

$G_3 = \varepsilon$

Variables $X2$ and $Y2$ in

$$\sigma_0 = \{X1/r(X2), Y1/s(Y2)\},$$

occur neither in G_0 nor in the head of C_0. They are introduced by E_{d_2}-unification to indicate that they can be replaced by any other terms.

$p(r(s(\_)), s(r(\_)))$ has been eliminated because it is not a provable from P by using SLD_{d_2}. The E_{d_2}-refutation process is as follows.

$G_0 = \leftarrow p(r(s(\_)), s(r(\_)))$

$C_0 = p(X1, Y1) \leftarrow q(X1, Y1)$

$\sigma_0 = \{X1/r(s(X2)), Y1/s(r(Y2))\}$

$G_1 = \leftarrow q(r(r(X2)), s(s(Y2)))$

$C_1 = q(r(X3), s(Y3)) \leftarrow q(X3, Y3)$

$\sigma_1 = \{X3/r(X4), Y3/s(Y4)\}$

$G_2 = \leftarrow q(r(X4), s(Y4))$

The E_{d_2}-refutation fails because no clause head E_{d_2}-unifies with $q(r(X4), s(Y4))$.

4. Stump Abstractions

Xu and Warren have introduced a family of abstractions, called stump abstractions, that reflect recursiveness [19]. The idea is to detail each atom in $\mathbf{T} \uparrow \omega$ to the extent in which some function symbol has been repeated for a given times.

This section defines stump abstractions, constructs a refinement operator and an equational unification algorithm for such abstractions, and exemplifies incremental refinement of success patterns of logic programs with respect to stump abstractions.

4.1 Stump Abstractions

Let t be a term and s a sub-term of t. We define $fc(s,t)$ as a function which, for each function symbol g in Σ, registers the number of nodes labelled by g in the path from the root of the term tree of t to but excluding the root of the term tree of s. Let $w \in (\Sigma \mapsto N)$ where N is the set of natural numbers. Define $w \oplus f = w[f \mapsto w(f)+1]$ and if $w(f) > 0$ then $w!f =$

$w[f \mapsto w(f)-1]$ .Define $fc: Term \times Term \to (\Sigma \mapsto N)$ as follows. If $s \equiv t$ then $fc(s,t) = \lambda f.0$. If $t = f(t_1,\cdots,t_m)$ and $fc(s,t_i) = w$ for some $1 \le i \le m$ then $fc(s,t) = w \oplus f$ .Otherwise, $fc(s,t)$ is undefined. If $s = g(s_1,\cdots,s_k)$ then the repetition depth of *s* in *t*, denoted as $rd(s,t)$ is defined as $fc(s,t)(g)$ .For instance, letting $t = f(g(h(1),g(1,2)),h(f(h(1),f(3,2))))$, $rd(f(3,2),t) = 2$, and $rd(g(1,2), t) = 1$.

Definition 3 Let $t \in Term$, and $w \in \Sigma \mapsto N$. $s_w(t)$ is obtained by replacing each sub-term $s = g(s_1,\cdots,s_k)$ of t satisfying $rd(s,t) = w(g)$ with $g(\_,\cdots,\_)$. Formally,

$$s_w(f(t_1,\cdots,t_m)) = \begin{cases} f(s_{w!f}(t_1),\cdots,s_{w!f}(t_m)) & w(f) \neq 0 \\ f(\_,\cdots,\_) & w(f) = 0 \end{cases}$$

For instance, letting $w = \{r \mapsto 1, g \mapsto 0, s \mapsto 1\}$, $s_w(r(g(s(g(1))))) = r(g(\_))$.

Lemma 6 For any $w \in \Sigma \mapsto N$, s_w is stable.

4.2 A Refinement Operator for Stump Abstractions

Let $x, y \in (\Sigma \mapsto N)$ and define

$$x \langle\langle y = \forall f \in \Sigma . x(f) \le y(f) .$$

As shown later, $x \langle\langle y \leftrightarrow s_y \subseteq s_x$. Intuitively, the bigger the limit for each function symbol, the weaker the abstraction.

Definition 4 Define

$$\bar{s}: (\Sigma \mapsto N) \times \Sigma \mapsto \wp(Term(\Sigma \cup \{\_\}, \varnothing))$$

as follows.

$$\begin{aligned} \bar{s}(w, f) &= \{f(\_,\cdots,\_)\} && w(f) = 0 \\ \bar{s}(w, f) &= \{f(t_1,\cdots,t_m) \mid t_j \in \bigcup_{g \in \Sigma} \bar{s}(w!f, g)\} && w(f) \neq 0 \end{aligned}$$

For given $w \in \Sigma \mapsto N$ and $f \in \Sigma$, $\bar{s}(w, f)$ is the set of the abstract terms identifying the $\approx_{s_w}$ equivalence classes of those ground terms whose main functors is *f*.

The following defined function

$$\tilde{s}: (\Sigma \mapsto N) \times Term(\Sigma \cup \{\_\}, \varnothing) \mapsto \wp(Term(\Sigma \cup \{\_\}, \varnothing))$$

splits an equivalence class of ground terms for a coarser stump abstraction into the set of equivalence classes of ground terms for a finer stump abstraction.

$$\tilde{s}(w, \_) = \bigcup_{f \in \Sigma} \bar{s}(w, f)$$

$$\tilde{s}(w, g(t_1,\cdots,t_m)) = \{g(s_1,\cdots,s_m) \mid \forall 1 \le j \le m . s_j \in \tilde{s}(w!g, t_j)\}$$

Its extension as in the following gives rise to a refinement operator for stump abstractions

$$\tilde{s}: (\Sigma \mapsto N) \times Atom(\Pi, Term(\Sigma \cup \{\_\}, \varnothing)) \mapsto \wp(Atom(\Pi, Term(\Sigma \cup \{\_\}, \varnothing)))$$

$$\tilde{s}(w, p(t_1,\cdots,t_m)) = \{p(s_1,\cdots,s_m) \mid \forall 1 \le j \le m . s_j \in \tilde{s}(w, t_j)\}$$

$$\tilde{s}^*: (\Sigma \mapsto N) \times \wp(Atom(\Pi, Term(\Sigma \cup \{\_\}, \varnothing))) \mapsto \wp(Atom(\Pi, Term(\Sigma \cup \{\_\}, \varnothing)))$$

is the extension of $\tilde{s}$ to sets of abstract atoms.

$$\tilde{s}^*(w, S) = \bigcup_{A^\# \in S} \tilde{s}(w, A^\#)$$

Lemma 7 For any $x \langle\langle y$, $R_{s_x, s_y} = \tilde{s}(y, \cdot)$ and $R^*_{s_x, s_y} = \tilde{s}^*(y, \cdot)$.

4.3 An E_{s_w} -Unification Algorithm

The E_{s_w} -unification algorithm is given in algorithm 2. The function *Sunif* has three parameters. The first parameter *w* maps each function symbol into the limit of its repetition depth. The second and third parameters are terms to be unified. For any variable *X* and term *t*, $occur(w, X, t)$ is true iff *X* occurs in $s_w(t)$.

Algorithm 2 This algorithm decides if t_1 and t_2 are E_{s_w} -unifiable and, if so, returns an E_{s_w} -mgu of t_1 and t_2.

```
01  function Sunify( w , t1 , t2 ) ⇒ ( unifiable, σ )
02  begin
03    if t1 or t2 is a variable then
04    begin let X be the variable and t the other
            term
05      if X = t then (unifiable, σ) ← (true, ∅)
06      else if occur(w, X, t) then (unifiable, σ))
                                  ←Sunify(w, X, t{X ↦ t}
07      else (unifiable, σ) ← (true, {X ↦ sw(t)})
08    end else
09    begin let t1 = f(x1,···, xn) and t2 = g(x1,···, xm)
10      if f ≠ g or m ≠ n then unifiable← false else
11      if w(f) = 0 then (unifiable, σ) ← (true, ∅)
          else
12        begin j←0,  (unifiable, σ0) ← (true, ∅)
13          while j < m and unifiable do
14            begin j ← j +1
15              (unifiable, τj)←Sunify(w!f,xjσj-1,yjσj-1)
16              if unifiable then σj ← σj-1τj
17            end
18          σ ← σm
19        end
20  end
```


21 return $(unifiable, \sigma)$

22 end

The line 06 in algorithm 2 deals with E_{s_w}-unification of X and t where X occurs in $s_w(t)$ by reducing the problem of E_{s_w}-unification of X and t into the problem of E_{s_w}-unification of X and $t\{X \mapsto t\}$.

Lemma 8 Let t_1 and t_2 be terms. If t_1 and t_2 are E_{s_w}-unifiable, then algorithm 2 terminates and gives an unique (module renaming) E_{s_w}-mgu of t_1 and t_2. Otherwise, the algorithm terminates and reports the fact.

4.4 Refinement of Success Patterns for Stump Abstractions

The following lemma establishes the appropriateness of incremental refinement method for stump abstractions.

Lemma 9 For any $x, y \in (\Sigma \mapsto N)$, $x \langle\langle y \leftrightarrow s_y \subseteq s_x$.

Lemma 9 implies that if $[A]_{\approx_{s_y}} \in \mathbf{T}^{s_y} \uparrow \omega$ then $[A]_{\approx_{s_x}} \in \mathbf{T}^{s_x} \uparrow \omega$ for any $x \langle\langle y$. This enables us to refine success patterns of P by increasing repetition depths for some function symbols. Suppose that success patterns in $\mathbf{T}^{s_x} \uparrow \omega$ are not fine enough and it is necessary to compute $\mathbf{T}^{s_y} \uparrow \omega$ for some y such that $x \langle\langle y$. Rather than throwing away $\mathbf{T}^{s_x} \uparrow \omega$ and computing $\mathbf{T}^{s_y} \uparrow \omega$ from scratch, we compute $\mathbf{T}^{s_y} \uparrow \omega$ by

1) applying $\tilde{s}^*(y, \cdot)$ to $\mathbf{T}^{s_x} \uparrow \omega$ resulting in a set of candidate elements for $\mathbf{T}^{s_y} \uparrow \omega$ since $\mathbf{T}^{s_y} \uparrow \omega \subseteq \tilde{s}^*(y, \mathbf{T}^{s_x} \uparrow \omega)$;

2) applying SLD_{s_y} to eliminate those candidate elements that are not from P using SLD_{s_y}.

The following two examples illustrate incremental refinement of success patterns for stump abstractions.

Example 4 Let

$$P = \{p(a,b), p(X,Y) \leftarrow q(X,Y), q(a,b), q(r(X), s(Y)) \leftarrow q(X,Y)\}$$

We have

$$\mathbf{T}^{s_{\lambda f.1}} \uparrow 0 = \varnothing$$

$$\mathbf{T}^{s_{\lambda f.1}} \uparrow 1 = \{p(a,b), q(a,b)\}$$

$$\mathbf{T}^{s_{\lambda f.1}} \uparrow 2 = \{p(a,b), q(a,b), q(r(a), s(b))\}$$

$$\mathbf{T}^{s_{\lambda f.1}} \uparrow 3 = \left\{\begin{array}{l} p(a,b), q(a,b), q(r(a), s(b)), \\ p(r(a), s(b)), q(r(r(\_)), s(s(\_))) \end{array}\right\}$$

$$\mathbf{T}^{s_{\lambda f.1}} \uparrow 4 = \left\{\begin{array}{l} p(a,b), q(a,b), q(r(a), s(b)), p(r(a), s(b)), \\ q(r(r(\_)), s(s(\_))), p(r(r(\_)), s(s(\_))) \end{array}\right\}$$

$$\mathbf{T}^{s_{\lambda f.1}} \uparrow 5 = \mathbf{T}^{s_{\lambda f.1}} \uparrow 4$$

So, $\mathbf{T}^{s_{\lambda f.1}} \uparrow \omega = \mathbf{T}^{s_{\lambda f.1}} \uparrow 5 =$

$$\left\{\begin{array}{l} p(a,b), q(a,b), q(r(a), s(b)), p(r(a), s(b)), \\ q(r(r(\_)), s(s(\_))), p(r(r(\_)), s(s(\_))) \end{array}\right\}$$

Example 5 Let P be the same as example 4. Suppose that success patterns in $\mathbf{T}^{s_{\lambda f.1}} \uparrow \omega$ are not fine enough. We compute $\mathbf{T}^{s_{\lambda f.2}} \uparrow \omega$ as follows. We first compute $\tilde{s}^*(\lambda f.2, \mathbf{T}^{s_{\lambda f.1}} \uparrow \omega)$ and then use $SLD_{s_{\lambda f.2}}$ to eliminate those candidates in $\tilde{s}^*(\lambda f.2, \mathbf{T}^{s_{\lambda f.1}} \uparrow \omega)$ that are not provable from P using $SLD_{s_{\lambda f.2}}$. The result is

$$\mathbf{T}^{s_{\lambda f.2}} \uparrow \omega = \left\{\begin{array}{l} p(a,b), q(a,b), q(r(a), s(b)), p(r(a), s(b)), \\ q(r(r(a)), s(s(b))), p(r(r(a)), s(s(b))), \\ q(r(r(r(\_))), s(s(s(\_)))), p(r(r(r(\_))), s(s(s(\_)))) \end{array}\right\}$$

$q(r(r(r(\_))), s(s(s(\_))))$ has not been removed because it is provable from P using $SLD_{s_{\lambda f.2}}$ as follows.

$$\begin{array}{l} G_0 = \leftarrow q(r(r(r(\_))), s(s(s(\_)))) \\ \quad C_0 = q(r(X1), s(Y1)) \leftarrow q(X1, Y1) \\ \quad \sigma_0 = \{X1 / r(r(X2)), Y1 / s(s(Y2))\} \\ G_1 = \leftarrow q(r(r(X2)), s(s(Y2))) \\ \quad C_1 = q(r(X3), s(Y3)) \leftarrow q(X3, Y3) \\ \quad \sigma_1 = \{X3 / r(X2), Y3 / s(Y2)\} \\ G_2 = \leftarrow q(r(X2), s(Y2)) \\ \quad C_2 = q(r(X4), s(Y4)) \leftarrow q(X4, Y4) \\ \quad \sigma_2 = \{X4 / X2, Y4 / Y2\} \\ G_3 = \leftarrow q(X2, Y2) \\ \quad C_3 = q(a,b) \\ \quad \sigma_3 = \{X2 / a, Y2 / b\} \\ G_4 = \varepsilon \end{array}$$

$q(r(r(s(a))), s(s(s(\_))))$ has been eliminated because it can not proved from P using $SLD_{s_{\lambda f.2}}$ as shown in the following.

$$\begin{array}{l} G_0 = \leftarrow q(r(r(s(a))), s(s(s(\_)))) \\ \quad C_0 = q(r(X1), s(Y1)) \leftarrow q(X1, Y1) \\ \quad \sigma_0 = \{X1 / r(s(a)), Y1 / s(s(Y2))\} \\ G_1 = \leftarrow q(r(s(a)), s(s(Y2))) \\ \quad C_1 = q(r(X3), s(Y3)) \leftarrow q(X3, Y3) \\ \quad \sigma_1 = \{X3 / s(a), Y3 / s(Y2)\} \\ G_2 = \leftarrow q(s(a), s(Y2)) \end{array}$$

The refutation process fails because there is no clause of P whose head $E_{s_{\lambda f.2}}$-unifies with $q(s(a), s(Y2))$.

5. Conclusions and Future Work

We have proposed a method for incrementally computing success patterns of logic programs for stable abstractions. We have introduced a partial order on abstractions to reflect relative strength of abstractions. The method makes use of a fixed-point and a procedural abstract semantics of logic programs with respect to stable abstractions, a refinement operator that splits an equivalence class induced by a coarser abstraction into a set of equivalence classes induced by a finer abstraction, and equational unification. The refinement operator is specified.

We have applied the method for incremental refinement of success patterns of logic programs for depth abstractions and stump abstractions by constructing suitable refinement operators and equational unification algorithms. For depth abstractions, abstraction depth can be increased uniformly while for stump abstractions, repetition depth for each function symbol can be increased independently.

For depth abstractions, abstraction depth can only be increased uniformly. That means that every equivalence class has to be split when analysis is refined. It would be better to be able to split some equivalence classes and keep others intact. However, it is not clear if such a fine-tuning approach will guarantee the stability of the resulting abstraction α which is a prerequisite of using SLD_α to eliminate false candidates.

Another interesting topic on incremental refinement of success patterns of logic programs is to study the possibility of applying $\mathbf{T}^{\alpha'}$ to eliminate false candidates where α' is the abstraction resulting from refinement. Yet another interesting topic on incremental refinement of success patterns of logic programs is to combine domain refinement such as that proposed in this paper with compositional approach towards logic program analysis proposed by Codish *et al.* [3] since compositional approach is the only feasible way to analyze large programs. It is necessary to study the interaction between the refinement of analyses of program modules and the composition of analyses of program modules.

6. Acknowledgements

This work was supported, in part, by NSF grant CCR-0131862.

REFERENCES

[1] P. Cousot and R. Cousot, "Systematic design of program analysis frameworks," Proceedings of the 6th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages, The ACM Press, New York, pp. 269–282, 1979.

[2] P. Cousot and R. Cousot, "Abstract interpretation and application to logic programs," The Journal of Logic Programming, Vol. 13, No. 2–3, pp. 103–179, 1992.

[3] H. Søndergaard, "An application of abstract interpretation of logic programs: Occur check problem," In: B. Robinet and R. Wilhelm, Ed., European Symposium on Programming, Lecture Notes in Computer Science, Springer, Vol. 213, pp. 324–338, 1986.

[4] C. Mellish, "Some global optimizations for a Prolog compiler," Journal of Logic Programming, Vol. 2, No. 1, pp. 43–66, 1985.

[5] C. Mellish, "Abstract interpretation of Prolog programs," In: S. Abramsky and C. Hankin, Ed., Abstract interpretation of declarative languages, Ellis Horwood Ltd., pp. 181–198, 1987.

[6] M. Bruynooghe and G. Janssens, "An instance of abstract interpretation integrating type and mode inferencing," Proceedings of the Fifth International Conference and Symposium on Logic Programming, The MIT Press, Seattle, pp. 669–683, 15–19 August 1988.

[7] D. Jacobs and A. Langen, "Static analysis of logic programs for independent and parallelism," Journal of Logic Programming, Vol. 13, No. 2–3, pp. 291–314, 1992.

[8] X. Li, A. King, and L. Lu, "Collapsing closures," In: S. Etalle and M. Truszczynski, Ed., Proceedings of the Twenty Second International Conference on Logic Programming, Lecture Notes in Computer Science, Vol. 4079, pp. 148–162, 2006.

[9] M. Bruynooghe, G. Janssens, A. Callebaut, and B. Demoen, "Abstract interpretation: Towards the global optimisation of Prolog programs," Proceedings of the 1987 Symposium on Logic Programming, The IEEE Computer Society Press, San Francisco, pp. 192–204, 31 August–4 September 1987.

[10] L. Lu, "Improving precision of type analysis using non-discriminative union," Theory and Practice of Logic Programming, Vol. 8, pp. 33–80, 2008.

[11] K. Marriott, H. Søndergaard, and N. D. Jones, "Denotational abstract interpretation of logic programs," ACM Transactions on Programming Languages and Systems, Vol. 16, No. 3, pp. 607–648, 1994.

[12] K. Marriott and H. Søndergaard, "Bottom-up abstract interpretation of logic programs," Proceedings of the Fifth International Conference and Symposium on Logic Pro- gramming, The MIT Press, Seattle, pp. 733–748, 15–19 August 1988.

[13] M. H. van Emden and R. A. Kowalski, "The semantics of predicate logic as a programming language," Artificial Intelligence, Vol. 23, No. 10, pp. 733–742, 1976.

[14] J. Jaffar, J. L. Lassez, and M. J. Maher, "Theory of complete logic programs with equality," Journal of Logic Programming, Vol. 1, No. 3, pp. 211–23, October 1984.

[15] T. Sato and H. Tamaki, "Enumeration of success patterns in logic programs," Theoretical Computer Science, Vol. 34, No. 1, pp. 227–240, 1984.

[16] P. M. Hill and F. Spoto, "Generalizing Def and Pos to type analysis," Journal of Logic and Computation, Vol.

12, No. 3, pp. 497–542, 2002.

[17] M. Li, Z. Li, H. Chen, and T. Zhou, "A novel derivation framework for definite logic program," Electronic Notes in Theoretical Computer Science, Vol. 212, pp. 71–85, 2008.

[18] J. A. Robinson, "A machine-oriented logic based on the resolution principle," Journal of the ACM, Vol. 12, No. 1, pp. 23–41, 1965.

[19] J. Xu and D. S. Warren, "A type inference system for Prolog," Proceedings of the 5th International Conference and Symposium on Logic Programming, The MIT Press, Seattle, pp. 604–619, 15–19 August 1988.

[20] M. Codish, S. K. Debray, and R. Giacobazzi, "Compositional analysis of modular logic programs," Proceedings of the 20th Annual ACM Symposium on Principles of Programming Languages, The ACM Press, New York, USA, pp. 451–464, January 1993.

J. Software Engineering & Applications, 2010, 3: 208-220
doi:10.4236/jsea.2010.33026 Published Online March 2010 (http://www.SciRP.org/journal/jsea)

Automated Identification of Basic Control Charts Patterns Using Neural Networks

Ahmed Shaban[1], Mohammed Shalaby[2], Ehab Abdelhafiez[2], Ashraf S. Youssef[1]

[1]Faculty of Engineering, Fayoum University, Fayoum, Egypt; [2]Faculty of Engineering, Cairo University, Giza, Egypt.
Email: ass00@fayoum.edu.eg, mashalaby@aucegypt.edu

Received November 11th, 2009; revised December 5th, 2009; accepted December 10th, 2009.

ABSTRACT

The identification of control chart patterns is very important in statistical process control. Control chart patterns are categorized as natural and unnatural. The presence of unnatural patterns means that a process is out of statistical control and there are assignable causes for process variation that should be investigated. This paper proposes an artificial neural network algorithm to identify the three basic control chart patterns; natural, shift, and trend. This identification is in addition to the traditional statistical detection of runs in data, since runs are one of the out of control situations. It is assumed that a process starts as a natural pattern and then may undergo only one out of control pattern at a time. The performance of the proposed algorithm was evaluated by measuring the probability of success in identifying the three basic patterns accurately, and comparing these results with previous research work. The comparison showed that the proposed algorithm realized better identification than others.

***Keywords*:** *Artificial Neural Networks* (*ANN*), *Control Charts*, *Control Charts Patterns*, *Statistical Process Control* (*SPC*), *Natural Pattern*, *Shift Pattern*, *Trend Pattern*

1. Introduction

With the widespread usage of automatic data acquisition system for computer charting and analysis of manufacturing process data, there is a need to automate the analysis of process data with little or no human intervention [1]. Many researchers tried to automate the analysis of control chart patterns by developing Expert Systems to limit the human intervention in the analysis process of the control chart [2–4]. More recently; Artificial Neural Network (ANN) approach had been investigated. Dislike expert systems approaches; ANN does not require explicit rules to identify patterns. It acquires knowledge of how to identify patterns by learning. Moreover ANN models are expected to overcome the problem of high false alarm rate; because it does not depend on any statistical tests that are usually required for the traditional methods. Also, no human intervention will be required when applying ANN, and thus pattern identification can be readily integrated with inspection and rapid manufacturing technologies.

Control charts patterns are categorized as natural and unnatural patterns. The presence of an unnatural pattern such as runs, shifts in process mean, or trends as shown in **Figure 1** means that a process is out of control. The accurate identification of these unnatural patterns will help the quality practitioners to determine the assignable causes for process variation; because each unnatural pattern has its related assignable causes.

Traditional control charts use only recent sample data point to determine the status of the process based on the control limits only. They do not provide any pattern related information. To increase a control chart sensitivity many supplementary rules like zone tests or run rules have been suggested by Grant and Leavenworth [5], Nelson [6], and Western Electric [7] to assist quality practitioners in detecting unnatural patterns. The primary problems with applying run rules are that the application of all the available rules simultaneously can yield an excess of false alarms due to the natural variability in the process.

This paper proposes an Artificial Neural Network algorithm to detect and identify the three basic control chart patterns; natural, shift, and trend. Natural variation is represented by normal (0, 1) variation, shift in process mean is expressed in terms of number of standard deviations and trend is expressed as the general slope of a trend line. This identification of each pattern is in addition to the traditional statistical detection of data runs. A run is a sequence of observations of increasing (decreas-

ing) points or a sequence of observations above or below the process mean [8]. It is assumed that a process starts in control (has natural pattern) and then may undergo only one out of control pattern at a time (see **Figure 1**). For sake of simplicity only cases of upward shift and trend patterns are investigated. The proposed algorithm aims to provide a practitioner with a reliable and automated identification tool; the ANN is designed to maximize the probability of success in identifying accurately only these three basic patterns. The paper presents next a literature review, the design of the ANN network, the proposed approach for ANN, testing of the ANN algorithm and finally the performance evaluation of the algorithm.

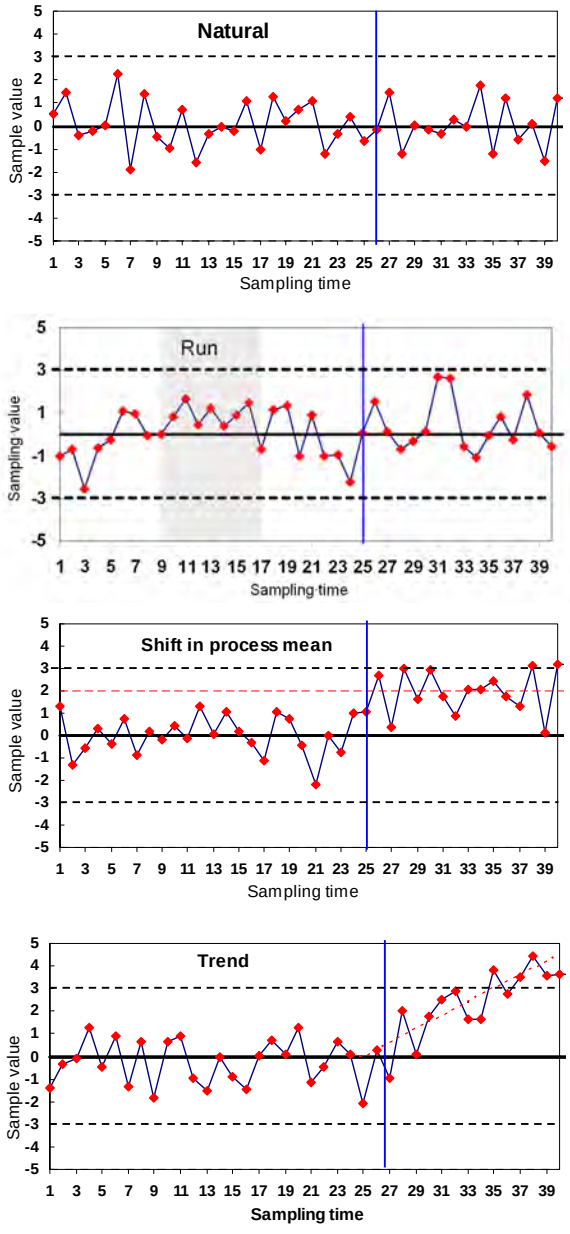


Figure 1. Basic patterns of control charts

2. Artificial Neural Network Approaches

ANN is investigated as an alternative tool for the traditional statistical process control tools. Some researchers tried to develop ANN models to detect sudden shifts in mean, shifts in variance, or both, and the others tried to develop ANN models to detect and identify all control chart patterns.

Smith [9] trained a single neural network to simultaneously model X-bar and R charts. The single output from the network was then interpreted simply as either no shift; a shift in mean; or a shift in variance. The input to smith's model was the subgroup observations plus some statistical characteristics obtained from the observations. Guo and Dooley [9] looked at positive shifts in both mean and variance using back propagation neural networks. Pugh [11,12] developed a back propagation neural network to detect a sudden shift in a process mean. Chang and Ho [13] developed a NN model that consists of two stages; stage one to detect the process variance change; and stage two to estimate the process variance magnitude. Their work resembles the R-chart function; where the R-chart signals out of control when the process variance had shifted. They extended their work and proposed an integrated neural network model to detect both a sudden process mean shift, and variance shift [14]. Also Dedeakayogullari and Burnak [15] developed two independent ANNs networks, one to detect the process mean change, and the second to detect the process variance change. The outputs of these two networks are analyzed to decide which shift has occurred. Cheng and Cheng [16] combined the traditional variance charts and ANN to detect the variance changes sources in a multivariate process. The traditional generalized variance chart works as a variance shift detector. When an out-of-control signal is generated, a classifier based ANN will determine which variable is responsible for the variance shift. Chena and Wang [17] developed an artificial neural network model to supplement the multivariate X^2 chart. The method identifies the characteristic or group of characteristics that cause the signal, and also classifies the magnitude of the shifts when the X^2 statistic signals a mean shift has occurred.

Guh and Hsieh [18] proposed a neural network model to identify control chart unnatural patterns and estimate key parameters of the identified pattern. The model was intended to identify natural, upward shift, downward shift, upward trend, downward trend, and cyclic pattern. Guh [19] developed a hybrid learning-based model, which integrates ANN and decision tree learning techniques, to detect typical unnatural patterns, while identifying the major parameter (such as the shift displacement or trend slope) and the starting point of the detected pattern. The model comprises two modules in series, Module I and Module II. Module I, comprises a general-purpose system that was designed and trained to

detect various types of unnatural patterns, and implements a procedure for classifying the actual type of the detected pattern. Module II is a special-purpose system that comprises seven specialized networks that are designed and trained to estimate the major parameters of the unnatural patterns. Similarly [20–24], and [25] utilized ANN to develop pattern recognizers that identify the abnormal control chart patterns.

Of special interest are the works of Cheng [26], Guh *et al.* [19], and Gauri and Chakraborty [27]. Cheng [26] developed a neural network model to detect gradual trends and sudden shifts in the process mean. The network structure was consisting of, an input layer consists of 17 nodes (neuron), hidden layer consist of 9 nodes, and output layer consist of only one node. The output node is the decision node about the presence of trend or sudden shift. His work emphasized only the detection (not the identification) of the present pattern. He evaluated his network by calculating the average run length and comparing with traditional control charts CUSUM and EWMA.

This paper will focus on the identification of the three basic patterns of control chart natural, upward shift and upward trend see **Figure 1**. A new neural network design will be discussed and a compatible training algorithm with the network structure will be selected. Also A new strategy to design the contents of the training data set will be introduced to minimize the required training data set size while improving the network performance.

3. Basic Neural Network Design

An artificial neural network consists mainly of an input layer, hidden layers, and output layer. Each layer consists of a set of processing elements or neurons, **Figure 2**. **Figure 3** represents a single neuron with R-elements input vector. The individual element inputs p_1, p_2, p_3... p_R are multiplied by weights $w_{1,1}$, $w_{1,2}$…$w_{1,R}$ and the weighted values are fed to the summing junction. Their sum is simply Wp, the dot product of the (single row) matrix W and the vector p. The neuron has a bias b, which is summed with the weighted inputs Wp to form the net input n. This sum, n, is the argument of the transfer function f. The structure of the single neuron in **Figure 3** is the same for all the neurons in the network. The network connection weights and biases are being optimized to learn the network to do its function.

The design of a network for a certain application consists of the determination of the number of hidden layers, number of neurons in each layer and the type of the transfer function where there are many types of transfer functions. The design of a suitable network is not an easy task, as there are many NN architectures which would satisfy an intended application [28]. Sagiroglu, Besdok and Erler [29] emphasized that no systematic method to

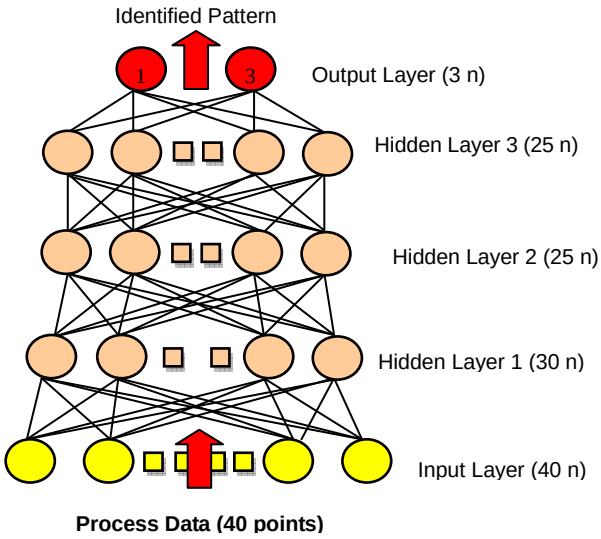


Figure 2. Network structure

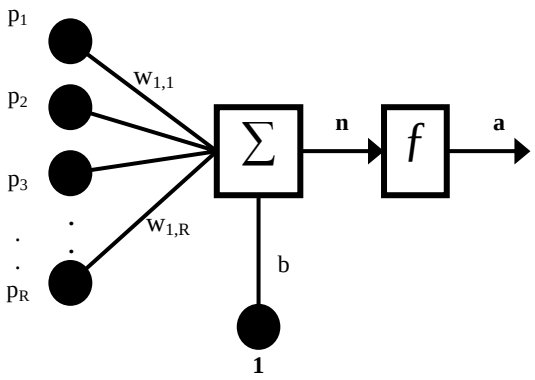


Figure 3. A single neuron [31]

select the optimum parameters. Guh [30] used the Genetic Algorithm (GA) to determine the neural networks configurations and the training parameters instead of using the trial and error method but in his latest work he used the selected parameters by trial and error method to construct his network.

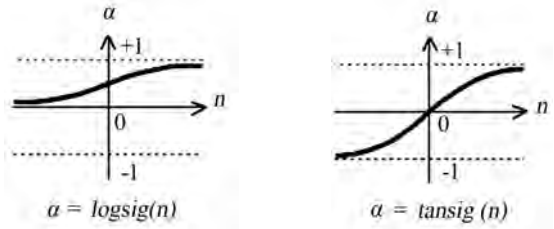
In this research a Multilayer feed forward Neural Network trained with the back ropagation learning rule is adopted to develop and train the network. In the literature the number of the hidden layers ranges between 1 and 2. Guh and Hsieh [18], Cheng [26], Gauri and Chakraborty [27] and Assaleh and Al-assaf [21] used only one hidden layer; Guh *et al.* [1] and Guh [19] used two hidden layers in their networks. Guh *et al.* [1] reported that, networks with two hidden layers performed better than those with one hidden layer. In this research 3 hidden layers were selected because this structure realized a good performance in a set of preliminary experiments. Also the size of the network is selected to be large enough to overcome the over fitting problem, one of the problems that occur

during neural network training is called over fitting [31]. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations. One method for improving network generalization is to use a network that is just large enough to provide an adequate fit. The larger network, the more complex the functions the network can create. Also the number of neurons in each hidden layer was selected based on a set of preliminary experiments. Guh *et al*. [1] also reported that, as the number of hidden neurons is increased, the learning results are usually improved too. As shown in **Figure 2** the network architecture was selected to be (40-30-25-25-3). The numbers of the hidden layers were selected to be 3, the first hidden layer consists of 30 neurons, the second and the third consists of 25 neurons each; the output layer consists of 3 neurons where each neuron is assigned for a certain pattern from the three patterns of interest.

The transfer function is an essential element in neural networks and has a great effect on their performance. In the literature the hyperbolic tangent function (tansig) and sigmoid transfer function (logsig) were adopted by many researchers in developing their networks. As shown in **Figure 4** the tansig function receives an input and transfers to an output ranges between -1 and 1, and the logsig function transfers the input to the range 0 and 1. These two functions work well with the back propagation algorithm because they are differentiable functions and the back propagation adjusts network weights based on the MSE function's gradient which is calculated by the first partial derivatives of the MSE function. Guh *et al.* [1] selected the sigmoid transfer function for the hidden and output layers, but Gauri and Chakraborty [27] selected the tansig function for the hidden layers and the logsig for the output layer. Based on a set of preliminary experiments the sigmoid transfer function was selected for both the hidden and output layers of the neural network in this study.

4. Network Training Data Generation

Neural networks can not perform their functions without training. In this research the supervised learning approach was adopted to optimize the network weights.



Log-Sigmoid Transfer Function Tan-Sigmoid Transfer Function

Figure 4. Transfer functions [31]

This approach was used by many previous researchers in this area. The training process is performed by introducing the training examples to the network; the output neuron value is compared with a target value, the difference between the target and the actual values is calculated and represented by MSE. With the aid of the training algorithm the network weights should be optimized to minimize the MSE see **Figure 7**. This process is repeated until a satisfactory MSE value is obtained. So that a sufficient number of training examples sets are required to train the neural networks. A Monte-Carlo simulation approach was adopted by the previous researchers to generate the training and testing data.

The Monte-Carlo simulation approach was adopted to generate the basic three control chart patterns (**Figure 1**) that may be exhibited by a control chart. The following equations were used to generate these patterns.

$$\text{Natural pattern} \qquad x(t) = \mu + n(t)\,\sigma \qquad (1)$$

$$\text{Upward shift pattern} \qquad x(t) = \mu + n(t)\,\sigma + d \qquad (2)$$

$$\text{Upward trend pattern} \qquad x(t) = \mu + n(t)\,\sigma + s\,t \qquad (3)$$

In the above equations $x(t)$ is a random variable following the normal distribution and represents the sample average value at a certain time t, μ is the natural process mean, σ is the process standard deviation, and $n(t)$ is the natural variability element in the process, and follows the normal distribution with $\mu_n = 0$ and $\sigma_n = 1$. The term d in the shift pattern equation represents the positive shift displacement from the process in control mean. The term s in the trend pattern equation represents the positive trend pattern slope. The training data was generated with $\mu = 0$ and $\sigma = 1$ to ensure the generality of the network for any process parameters.

In this research the identification window size consists of 40 points, or 40 sampled observations, also this size represents the size of the training examples. These 40 points actually represents the recently drawn samples from the process. This size is nearly the average of the different sizes used in the literature. In practical situations the process starts in control and then goes out of control. Cheng [26] recommended training a network with a mixture of natural and unnatural pattern to avoid high type II error. Guh *et al*. [1] and Guh [19] assumed that the process starts in-control and then goes out-of control in the practical situations, and generated training examples have size of 24 points include both in-control and out of control points. This strategy will allow large process change to be detected quickly by changing the start time of the different parameters of the unnatural patterns. In this study all the unnatural patterns start at the 26th point in the training examples see **Figure 5**. All the shift and trend training examples were generated to have the first 25 points in control (natural pattern) and

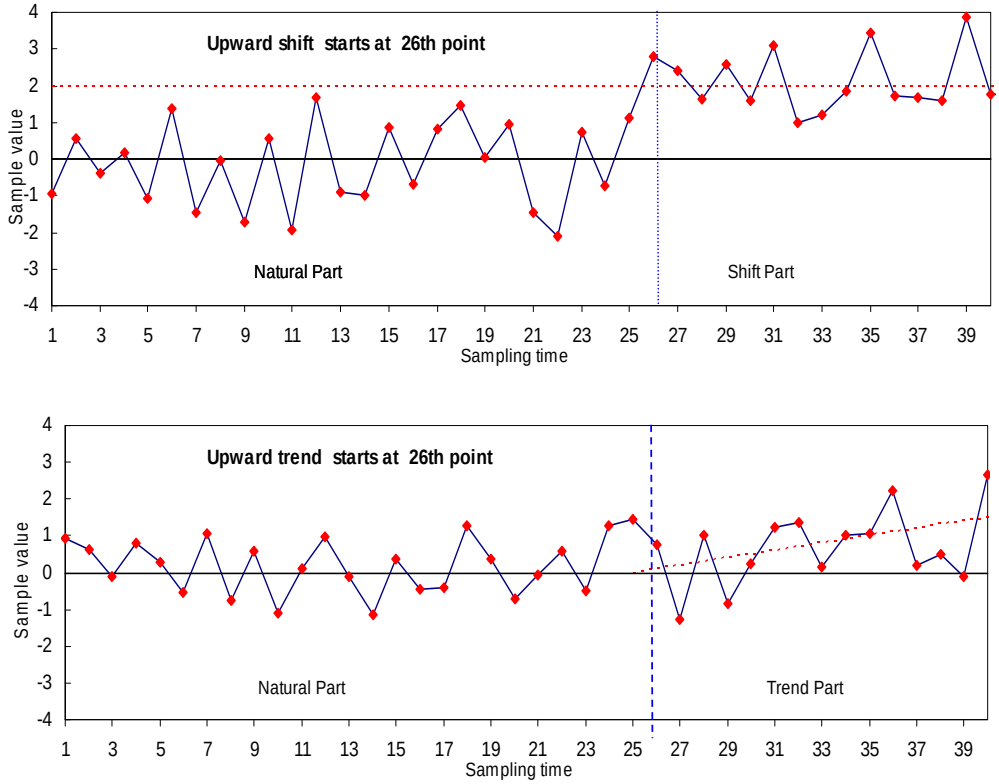


Figure 5. Two training examples

the last 15 points are out of control and contain the unnatural pattern (shift or trend) to simulate the practical situations. The flowchart of the random patterns generation routine is given in Appendix A.

The details of the training data set and their related target values for the output neurons are recorded in **Table 1**.

The design of the contents of the training data set is very important and has a great effect on the network performance. No particular rules to design the appropriate training data set were followed in the literature. Previous researchers were generating training examples at multiple parameters for a single pattern to cover a wide range of the pattern parameters. It was claimed that this strategy will make the network handle a general problem. Guh [19] for example, generated the shift pattern training sets to cover a range of shift magnitudes from 1σ to 3σ. Nine training sets were generated for the shift pattern within this range; in increments of 0.25σ. Each set consisted of 90 examples. Similarly 9 training data sets were generated for the upward trend pattern, each data set consists of 90 examples. This strategy was adopted nearly by all the reviewed work in this area. Training a network with multiple patterns and multiple parameters for each pattern is expected to make the network confused and may lead to misclassifying the actual patterns to other patterns that have similar features and will also make the learning process more difficult. Based on a set of preliminary experiments, the misclassification problem appeared at a certain pattern parameters. For example, when a network was trained with trend of 0.05σ slope, and 1σ shift the trend pattern was misclassified as natural and vice versa. The network confusion happened because the trend slope is very small and the length of the trend pattern is also small this make the trend pattern very similar to the natural. Confusion between the shift and trend happens when training the network with multiple trend and shift patterns parameters.

A new training strategy is adopted in this study by using less training data sets for each single pattern. The injection of a certain training data sets with lesser number of parameters may help to solve this problem. In the above example, when the training data set was only supported with another trend data set having a slope of 0.1 sigma the classification accuracy was improved over a wide range of trend slopes. **Table 2** presents five alternative training data sets each set contains a certain pattern parameters. After investigating these alternatives, classification accuracy improved by training the network with the lesser number parameters or with the addition of specific data set to solve a specific misclassification problem. In **Table 2**, Set (5) was the best alternative where it realized small MSE and excellent identification. Thus lesser

number of pattern parameters are recommended to train the network. To train the network only one shift and two trend slope parameters were used. Preliminary experiments showed that networks trained by this way perform better over a wide range of shift and trend parameters. This approach also will minimize the required time to train the network; because it requires smaller number and sizes of training data sets. The selected structure of the training data set is Set (5). Two hundred training examples were generated for each pattern parameter within the selected data set, the sum of all the training examples is 800 which represents the size of the training data set. All sampled observations within a particular example were randomly generated from normal distribution, where a single example consists of 40 observations. All examples were filtered from runs, since runs can be easily detected by traditional computational rules without the need for the ANN algorithm. Any training example consists of 25 points of a natural pattern and the 15 points of a selected unnatural pattern. All generated observations within any example from a normal (0, 1) distribution will be filtered from runs.

The presence of the runs affects the identification process of the ANN badly. As shown in **Figure 6(a)** a run starts at the 29th point in a natural pattern training example, the run makes the series like shift, in **Figure 6(b)** the run makes the natural pattern like the trend pattern, in **Figures 6(c, d)** the shift pattern may be approximated to trend pattern. Runs could be randomly generated during the random generation of the different examples. A simple computational process for runs was applied to identify two types of runs.

1) If 5 out of 6 points are monotone increasing or decreasing;

2) If 5 out of 6 points above or below the mean value.

Once run is detected, the corresponding example is excluded from the data set. **Figure 6** shows generated training examples have runs.

5. Network Training Process

After preparing the training data set, the network must be initialized for training. MATLAB 7 Neural Toolbox and environment were used to develop and train the network. The training patterns also were generated by MATLAB 7 generator. After initiating the network, the initial connection weights and biases were initialized by the built-in Nguyen-Widrow initialization algorithm in MATLAB [31].

Neural network is trained based on a comparison of the output and the target, until the network output matches the desired target (see **Figure 7**). Typically many input/target pairs are used, in this supervised learning approach to train a network. In **Figure 7** the input represents the training examples of the control chart patterns, output is the obtained output neurons based on the current values of the weights and biases, and the target is the desired neurons' output. As shown in **Table 1** each input pattern has its desired neuron's output, target value.

The training process is an optimization process in which the (performance or objective) function is the Mean Square Error (MSE) and the decision variables are the connection weights and biases. The target is to minimize the MSE by changing and adjusting the weights and biases to realize minimum MSE. There are many variations of the back propagation training algorithm, they adjust the network weights by making many learning cycles until the weights and biases reach their optimum values which realize the minimum MSE. In the literature the delta rule algorithm was adopted by many researchers to train their networks. In this study the Resilient Back propagation algorithm was adopted to train the network.

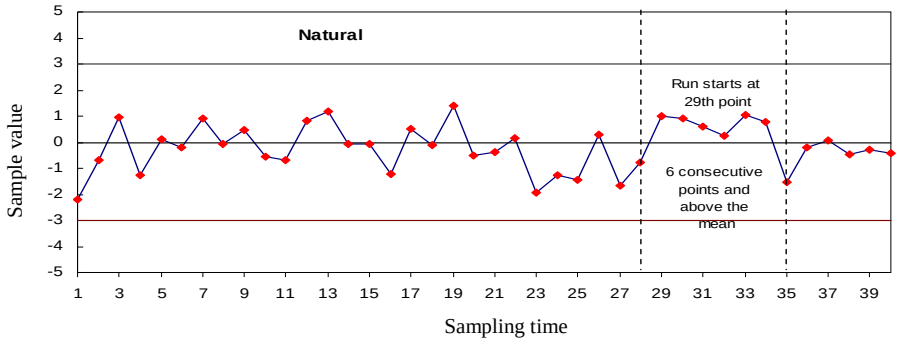
Table 1. Training data set structure

Pattern	Parameters	Pattern start time	Size of training sets	Output neurons desired output		
				neuron1	neuron2	neuron3
Natural	$\mu = 0, \sigma = 1$	-	200	**1**	0	0
Upward Shift	$d = 1\sigma$	26	200	0	**1**	0
Upward Trend	$s = 0.05\sigma, 0.1\sigma$	26	400	0	0	**1**

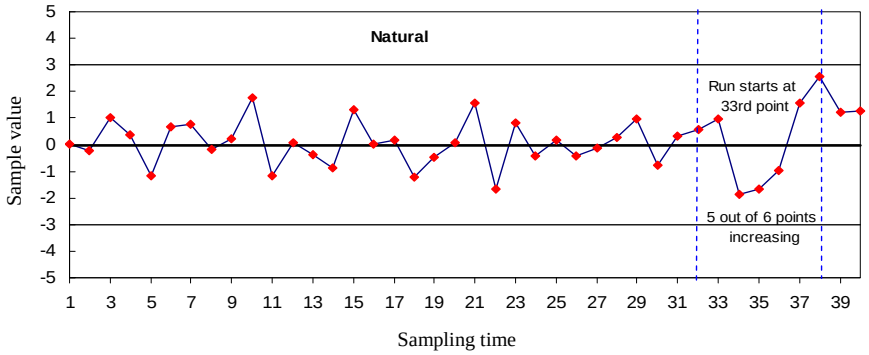
Table 2. Different alternatives training data sets

Training data set	Training data set structure		
	Natural	Shift magnitudes	Trend slopes
Set(1)	$\mu = 0, \sigma = 1$	$1\sigma, 2\sigma, 3\sigma$	$0.05\sigma, 0.1\sigma, 0.3\sigma, 0.5\sigma$
Set(2)	$\mu = 0, \sigma = 1$	2σ	0.1σ
Set(3)	$\mu = 0, \sigma = 1$	2σ	$0.05\sigma, 0.1\sigma$
Set(4)	$\mu = 0, \sigma = 1$	1σ	0.05σ
Set(5)	$\mu = 0, \sigma = 1$	1σ	$0.05\sigma, 0.1\sigma$

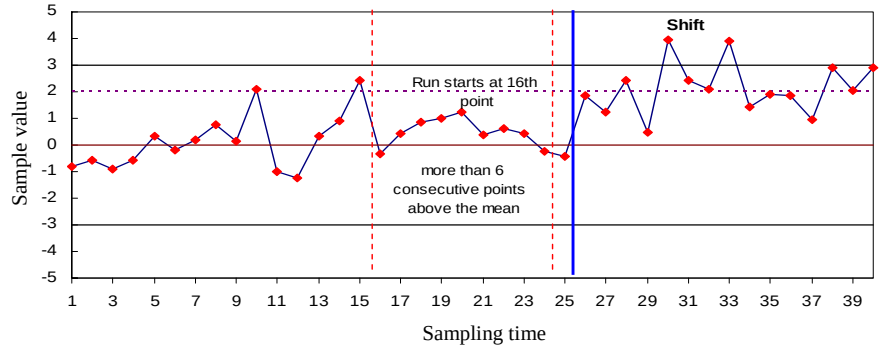
**μ is the process mean and equal 0; and σ is the process standard deviation and equal 1.



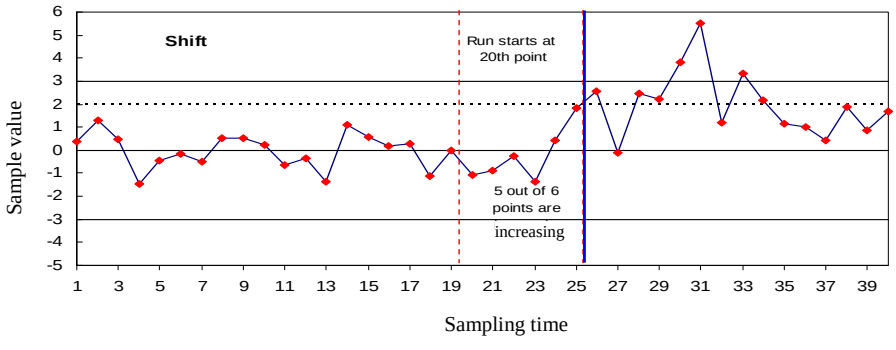
(a)



(b)



(c)



(d)

Figure 6. Training examples exhibiting runs

This algorithm was selected to be compatible with the network structure to eliminate harmful effects of the selected sigmoid functions. Sigmoid functions are characterized by the fact that their slopes must approach zero

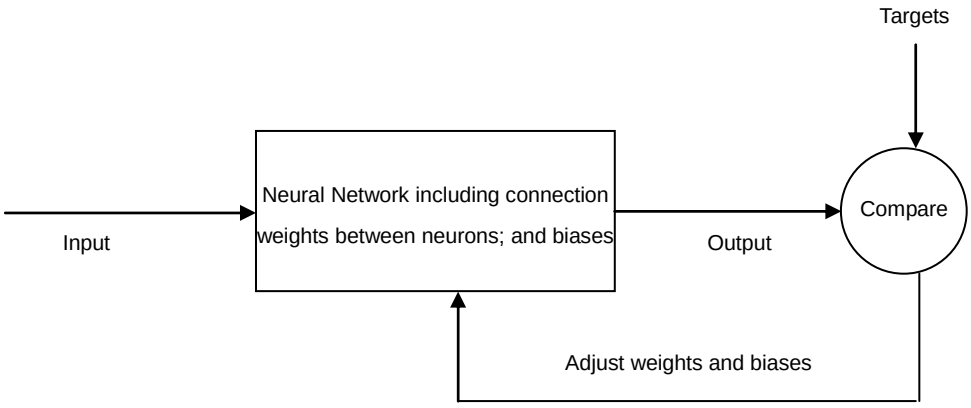


Figure 7. Supervised training

as the input gets large, this causes a problem when using steepest descent to train a multilayer network with sigmoid functions. The gradient in this case can have a very small magnitude and, therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values. The purpose of the resilient Back propagation training algorithm is to eliminate these harmful effects of the magnitudes of the partial derivatives. Only the sign of the derivative is used to determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update.

The network training convergence condition was set to MSE = 10^{-40} and the maximum number of learning cycles allowed to reach was set to be 100 epochs. While network is trained by these parameters, the network converged within 100 epochs as seen in **Figure 8** with MSE = 2.055×10^{-35}. The small MSE value was realized by using less number of pattern parameters to train the network. After training, the network was tested by the training data set and realized 100% correct identification for all the patterns.

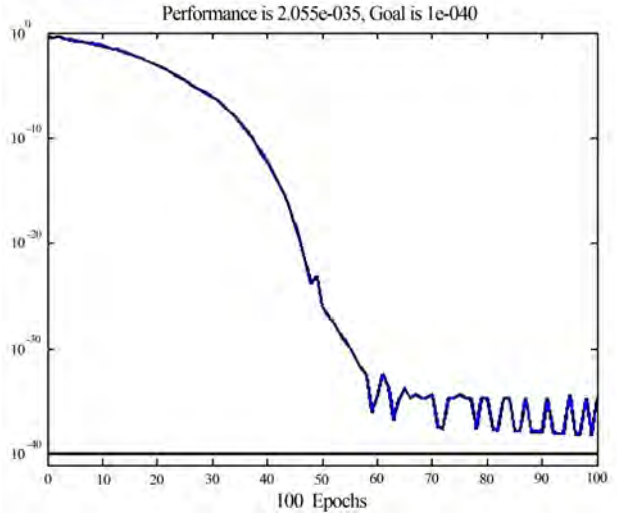


Figure 8. Training process output (MSE vs. Epoch number)

6. Network Testing and Performance Evaluation

After training the network, it must be tested and evaluated to measure its effectiveness for use. Probability of success measure was used by Guo and Dooly [10], Smith [9], Assaleh and Al-assaf [21], Guh [19], and Gauri and Chakraborty [27] to evaluate the performance of their trained neural networks. Probability of success expresses the percentage of correct identification, and it measures the capability of the network to detect and classify the pattern to the target class. In the literature the probability of success was found under different names such as the classification rate, the classification accuracy, recognition rate, and recognition accuracy. In this study the probability of success term will be used instead of the previous names. Al-assaf [32] and Guh [19] defined the classification rate as the number of correctly recognized examples divided by total number of examples.

To measure the probability of success a new set of data was randomly generated by using the patterns equations of Section 4. The trained network is tested at multiple parameters for each pattern to assure the network generalization for different pattern parameters. A set of 200 testing example was generated for each pattern parameter. Thus, a set of 200 testing example was generated for natural pattern; a total of 600 testing example was generated for shift pattern to test the network at 1σ, 2σ, and 3σ; and a total of 800 testing example was generated for trend pattern to test the network at slopes of 0.05σ, 0.1σ, 0.3σ, and 0.5σ. As mentioned earlier, each example consists of 40 points; 25 points of natural pattern followed by 15 points of the tested patterns.

Table 3 exhibits the three target patterns (known by construction of the testing examples), three possible identifications patterns, and the percentage of success of the ANN to identify a given target pattern. Three pattern

parameters were used for upward shift and 4 parameters were used for upward trend. Table entries represent the average percentages of success resulted from testing 10 different randomly generated data sets for a single given parameter. **Table 4** exhibits the percentage mean, standard deviation, max and min of success of the 10 sampled data sets for each pattern parameter. The fourth column in **Table 3** represents the percentage that the ANN was unable to make an identification decision. The following procedure was applied to all generated examples to obtain **Table 3** results.

Step1: Input a testing example to the trained network;

Step2: Find the network output (the three values of the three output neurons v_1, v_2, v_3); and find maximum output value $v_{max;}$

Step3: If v_{max} >= 0.01, then Identify the present pattern based on v_{max}, if v_{max} comes from the first neuron the pattern is natural; else if it comes from the second neuron the pattern is upward shift; else the pattern is upward trend;

Step4: Else if v_{max} < 0.01, the present pattern is unknown.

Results of **Table 3** show that the network can perform well in identifying the three basic patterns of control chart at a wide range of parameters. Moreover, the variation between Min and Max percentage of success for replications of the data sets is minimal which implies robustness in identification. However, a misclassification problem happened between the natural and the trend pattern that has small slope, where 1.8% of the natural testing examples were miss-classified as upward trend and 1.6% of the upward trend that has slope 0.05σ was miss-classified as natural. The misclassification happened because the upward trend at small slopes like 0.05σ is very similar to natural pattern; also the small pattern length in the testing examples makes the upward trend very similar to natural. **Figure 9** shows two cases of similarity between natural and upward trend with slope 0.05σ.

The ANN performance is compared to the reported results in the literature. The percentage of success results are compared with Al-assaf [32], Guh [19], and Gauri and Chakraborty [27] results. Their reported results are the most recent and appear to be the highest percentage of success in identifying control chart patterns. Gauri and

Table 3. Average probability of success results based on 10 runs

Target Pattern	Testing Parameter	ANN Identification Percentages			
		Natural	Upward shift	Upward trend	Unknown
Natural		98.2	0	1.8	0
Upward shift	1 σ	0	100	0	0
	2 σ	0	100	0	0
	3 σ	0	100	0	0
	Average	**0**	**100**	**0**	**0**
Upward trend	0.05 σ	1.6	0	98.35	0.05
	0.1 σ	0	0.15	99.85	0
	0.3 σ	0	0	100	0
	0.5 σ	0	0	100	0
	Average	**0.4**	**0.038**	**99.55**	**0.013**

Table 4. Probability of success results summary of the 10 runs

Target Pattern		Actual Identification			
		Average	Standard deviation	Max	Min
Natural		98.2	0.258	98.5	98
Upward shift	1 σ	100	0	100	100
	2 σ	100	0	100	100
	3 σ	100	0	100	100
Upward trend	0.05 σ	98.35	0.669	99.5	97
	0.1 σ	99.85	0.242	100	99.5
	0.3 σ	100	0	100	100
	0.5 σ	100	0	100	100

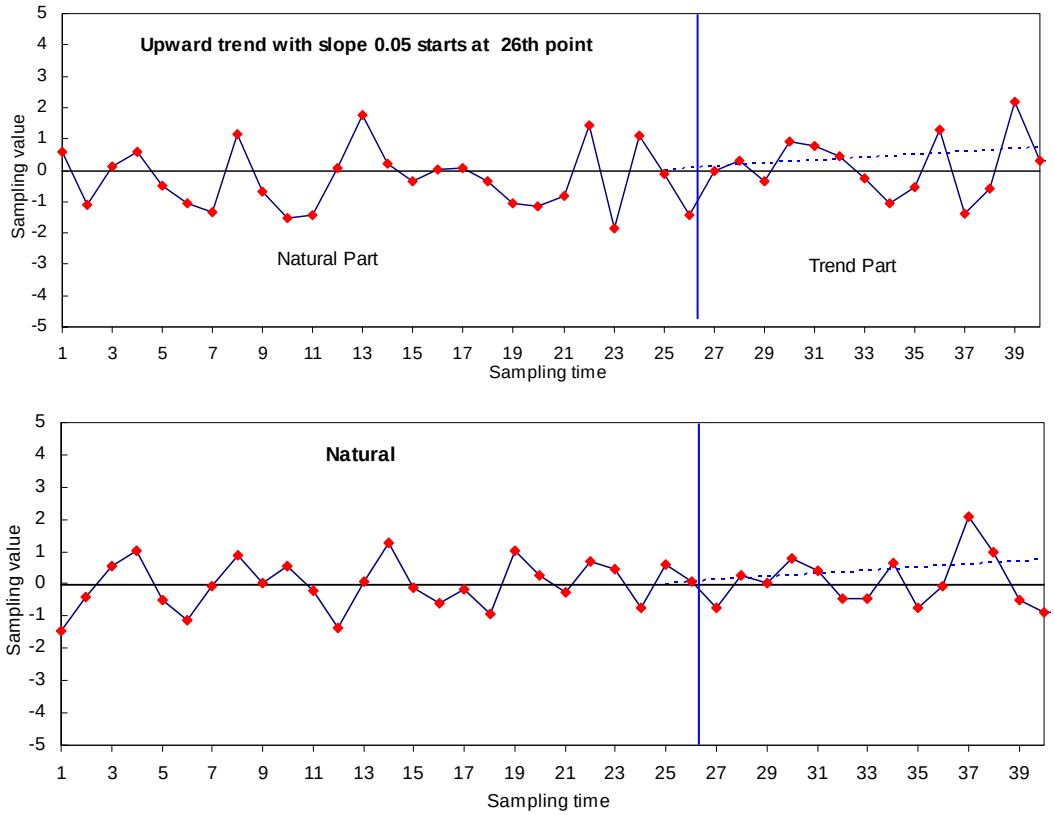


Figure 9. The similarity between trend pattern and natural at small slopes

Chakraborty [27] developed two feature-based approaches using heuristics and artificial neural network, which are capable of recognizing eight control chart patterns. They compared the results of the neural network with a heuristic approach and stated that the neural network results were better than heuristic results. Al-assaf [32] used the probability of success to compare between three approaches (DC, MRWA, and MRWA + DSC) to detect and classify the control chart unnatural patterns, his best results was obtained by using MRWA + DSC, so these results was used in the comparison. Guh [19] developed a hybrid learning-based model for on-line detection and analysis of control chart patterns; he trained a network to recognize eight control chart patterns and integrated this network in an algorithm to make it capable for on-line control chart analysis. In his work, the neural network testing results were reported based on the probability of success. **Table 5** summarizes the comparison with their results.

The comparisons indicate that the trained network in this study is comparable if not superior. It has a good uniformly identification performance with the three basic control chart patterns. This proves that changing the network structure and using a compatible training algorithm with the network structure has a great effect on the network performance.

Table 5. Results comparison based on the percentage of success with the other authors

Pattern	Proposed ANN	Al-Assaf (2004)	Guh (2005)	Gauri and Chakraborty (2006)
Natural	98.2	88.60	90.59	94.87
Upward shift	100	93.20	93.33	93.40
Upward trend	99.55	94.60	95.43	96.53

7. Conclusions

This paper investigates a new approach to train a neural network to detect and identify the basic three control chart patterns natural, upward shift, and upward trend in addition to the traditional identification of runs. Instead of using a large training data set only small one can be used to do the job. Using a smaller training data set will make the network training convergence easier and using smaller set of patterns parameters in the training will eliminate the network to confusion and misclassification. Also a new network structure and a compatible training algorithm were suggested to make the network perform effectively. The results show that network can perform effectively and the percentage of suc-

cess to identify a wide range of patterns is high and comparable if not superior to the previous reported results. This proves that changing the network structure and using a compatible training algorithm with the network structure has a great effect on the network performance.

REFERENCES

[1] R.-S. Guh, F. Zorriassatine, J. D. T. Tannock, and C. O' Brien, "On-line control chart pattern detection and discrimination: A neural network approach," Artificial Intelligence in Engineering, Vol. 13, pp. 413–425, 1999.

[2] T. L. Lucy-Bouler, "Using autocorrelations, CUSUMs and runs rules for control chart pattern recognition: An expert system approach," PhD dissertation, University of Alabama, Tuscaloosa, 1991.

[3] C.-S. Cheng and N. F. Hubele, "Design of a knowledge-based expert system for statistical process control," Computers and Industrial Engineering, Vol. 22, No. 4, pp. 501–517, 1992.

[4] J. A. Swift and J. H. Mize, "Out-of-control pattern recognition and analysis for quality control charts using Lisp-based systems," Computers and Industrial Engineering, Vol. 28, No. 1, pp. 81–91, 1995.

[5] E. L. Grant and R. S. Leavenworth, "Statistical quality control," 7th Edition, McGraw-Hill, New York, 1996.

[6] L. S. Nelson, "The Shewhart control chart: Tests for special causes," Journal of Quality Technology, Vol. 16, pp. 237–239, 1984.

[7] Western Electric, "Statistical quality control handbook," AT&T, Princeton, 1956.

[8] D. C. Montgomery, "Introduction to statistical quality control," 3rd Edition, Wiley, New York, 1996.

[9] A. E. Smith, "X-Bar and R control chart interpretation using neural computing," International Journal of Production Research, Vol. 32, pp. 309–320, 1994.

[10] Y. Guo and K. J. Dooley, "Identification of change structure in statistical process control," International Journal of Production Research, Vol. 30, pp. 1655–1669, 1992.

[11] G. A. Pugh, "Synthetic neural networks for process control," Computers and Industrial Engineering, Vol. 17, pp. 24–26, 1989.

[12] G. A. Pugh, "A comparison of neural networks to SPC charts," Computers and Industrial Engineering, Vol. 21, pp. 253–255, 1991.

[13] S. I. Chang and E. S. HO, "A two-stage neural network approach for process variance change detection and classification," International Journal of Production Research, Vol. 37, No. 7, pp. 1581–1599, 1999.

[14] S. I. Chang and E. S. Ho, "An integrated neural network approach for simultaneous monitoring of process mean and variance shifts a comparative study," International Journal of Production Research, Vol. 37, pp. 1881–1901, 1999.

[15] I. Dedeakayogullari and N. Burnak, "Determination of mean and/or variance shifts with artificial neural networks," International Journal of Production Research. Vol. 37, No. 10, pp. 2191–2200, 1999.

[16] C.-S. Cheng and H.-P. Cheng, "Identifying the source of variance shifts in the multivariate process using neural networks and support vector machines," Expert Systems with Applications, Vol. 35, No. 1–2, pp. 198–206, 2008.

[17] L.-H. Chena and T.-Y. Wang, "Artificial neural networks to classify mean shifts from multivariate X^2 chart signals," Computers & Industrial Engineering, Vol. 47, pp. 195–205, 2004.

[18] R.-S. Guh and Y.-C. Hsieh, "A neural network based model for abnormal pattern recognition of control charts," Computers and Industrial Engineering, Vol. 36, pp. 97–108, 1999.

[19] R.-S. Guh, "A hybrid learning-based model for on-line detection and analysis of control chart patterns," Computers and Industrial Engineering, Vol. 49, pp. 35–62, 2005.

[20] M. B. Perry, J. K. Spoerre, and T. Velasco, "Control chart pattern recognition using back propagation artificial neural networks," International Journal of Production Research, Vol. 39, pp. 3399–3418, 2001.

[21] K. Assaleh and Y. Al-assaf "Features extraction and analysis for classifying causable patterns in control charts," Computers and Industrial Engineering, Vol. 49. pp. 168–181, 2005.

[22] A. Hassan, M. S. N. Baksh, A. M. Shaharoun, and H. Jamaluddin, "Improved SPC chart pattern recognition using statistical features," International Journal of Production Research, Vol. 41, No. 7, pp. 1587–1603, 2003.

[23] A. Hassan, M. S. N. Baksh, A. M. Shaharoun, and H. Jamaluddin, "Feature selection for SPC chart pattern recognition using fractional factorial experimental design," Intelligent Production Machines and System: 2nd I*IPROMS Virtual International Conference, In: D. T. Pham, E. E. Eldukhri, and A. J. Soroka Ed., Elsevier, pp. 442–447, 3–14 July 2006.

[24] T. Zan, R.-Y. Fei, and M. Wang, "Research on abnormal pattern recognition for control chart based on neural network," Beijing Gongye Daxue Xuebao, Journal of Beijing University of Technology, Vol. 32, No. 8, pp. 673–676, 2006.

[25] Z. Chen, S. Lu, and S. Lam, "A hybrid system for SPC concurrent pattern recognition," Advanced Engineering Informatics, Vol. 21, No. 3, pp. 303–310, 2007.

[26] C.-S. Cheng, "A multi-layer neural network model for detecting changes in the process mean," Computers and Industrial Engineering, Vol. 28, No. 1, pp. 51–61, 1995.

[27] S. K. Gauri and S. Chakraborty, "Feature-based recognition of control chart patterns," Computers and Industrial Engineering, Vol. 51, pp. 726–742, 2006.

[28] F. Zorriassatine and J. D. T. Tannock, "A review of neural networks for statistical process control," Journal of Intelligent Manufacturing, Vol. 9, pp. 209–224, 1998.

[29] S. Sagiroglu, E. Besdok, and M. Erler, "Control chart pattern recognition using artificial neural networks," Turkish Journal of Electrical Engineering, Vol. 8, No. 2, pp. 137–147, 2000.

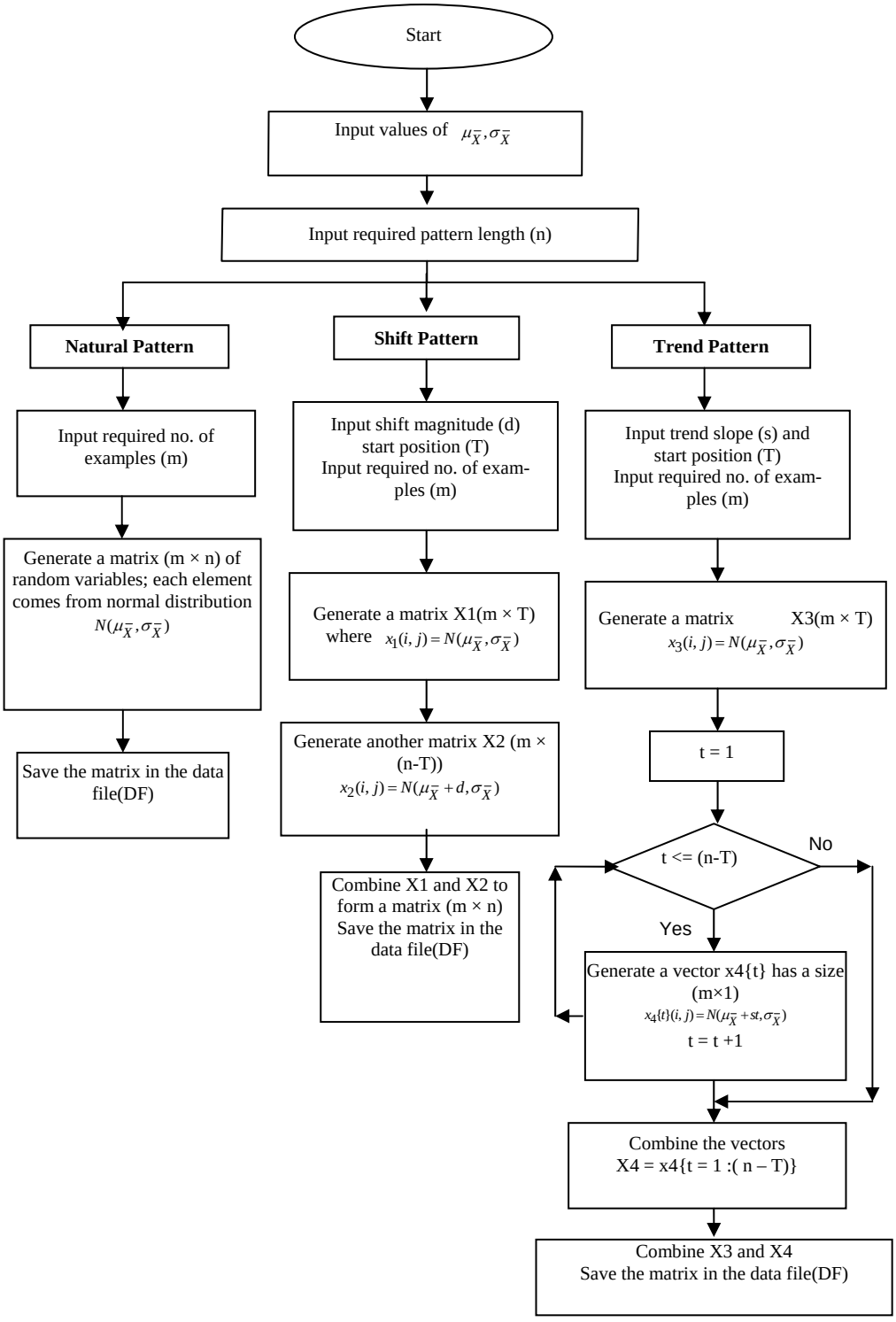
[30] R.-S. Guh, "Optimizing feedforward neural networks for control chart pattern recognition through genetic algorithms," International Journal of Pattern Recognition and Articial Intelligence, Vol. 18, No. 2, pp. 75–99.

[31] H. Demuth, M. Beale, and M. Hagan "Neural network toolbox user's guide," Math Works, Natick, MA, 2009.

[32] Y. Al-Assaf, "Recognition of control chart patterns using multi-resolution wavelets analysis and neural networks," Computers and Industrial Engineering, Vol. 47, pp. 17–29, 2004.

Appendix A

Pattern generation detailed flowchart



J. Software Engineering & Applications, 2010, 3: 221-229
doi:10.4236/jsea.2010.33027 Published Online March 2010 (http://www.SciRP.org/journal/jsea)

Parameter Identification Based on a Modified PSO Applied to Suspension System

Alireza Alfi, Mohammad-Mehdi Fateh

Faculty of Electrical and Robotic Engineering, Shahrood University of Technology, Shahrood, Iran.
Email: a_alfi@shahroodut.ac.ir

Received November 25th, 2009; revised January 15th, 2010; accepted January 20th, 2010.

ABSTRACT

This paper presents a novel modified particle swarm optimization algorithm (*MPSO*) *for both offline and online parametric identification of dynamic models. The MPSO is applied for identifying a suspension system introduced by a quarter-car model. A novel mutation mechanism is employed in MPSO to enhance global search ability and increase convergence speed of basic PSO* (*BPSO*) *algorithm. MPSO optimization is used to find the optimum values of parameters by minimizing the sum of squares error. The performance of the MPSO is compared with other optimization methods including BPSO and Genetic Algorithm* (*GA*) *in offline parameter identification. The simulating results show that this algorithm not only has advantage of convergence property over BPSO and GA, but also can avoid the premature convergence problem effectively. The MPSO algorithm is also improved to detect and determine the variation of parameters. This novel algorithm is successfully applied for online parameter identification of suspension system.*

Keywords**:** *Particle Swarm Optimization, Genetic Algorithm, Parameter Identification, Suspension System*

1. Introduction

A mathematical model can be provided to describe the behavior of a system based on obtained data for its inputs and outputs by system identification. It is necessary to use an estimated model for describing the relationships among the system variables for this purpose. The values of parameters in the estimated model of a system must be found such that the predicted dynamic response coincides with that of the real system [1].

The basic idea of parameter identification is to compare the time dependent responses of the system and parameterized model based on a performance function giving a measure of how well the model response fits the system response. It should be mentioned that the model of system must be regarded fixed through identification procedure. It means that data are collected from the process under a determined experimental condition and after that, the characteristic property of system will stay the same.

Many traditional techniques for parameter identification have been studied such as the recursive least square [2], recursive prediction error [3], maximum likelihood [4], and orthogonal least square estimation [5]. Despite their success in system identification, traditional optimization techniques have some fundamental problems including their dependence on unrealistic assumptions such as unimodal performance landscapes and differentiability of the performance function, and trapping in local minima [6].

Evolutionary algorithms (EAs) and swarm intelligence (SI) techniques seem to be promising alternatives as compared with traditional techniques. First, they do not rely on any assumptions such as differentiability, continuity, or unimodality. Second, they can escape from local minima. Because of this, they have shown superior performances in numerous real-world applications. Among them, genetic algorithm (GA) and particle swarm optimization (PSO) are frequently used algorithms in the area of EAs and SI, respectively. Owing these attractive features, these algorithms are applied in the area of system identification [7–10].

Comparing GA and PSO, both are population based optimization tools. However, unlike GA, PSO has no evolution operators such as crossover and mutation. Easy to implement and the less computational complexity are advantages of PSO in comparing with GA. The basic PSO (BPSO) algorithm has good performance when dealing with some simple benchmark functions. However, it is difficult for BPSO algorithm to overcome local minima when handling some complex or multimode functions. Hence, a modified PSO (MPSO) is proposed

to overcome this shortage. In this paper, a novel mutation mechanism is introduced to enhance global search of algorithm. Then, it is demonstrated how to employ the MPSO method to obtain the optimal parameters of a dynamic system.

In order to show the effectiveness of MPSO in system identification, a quarter-car model of suspension system is identified as an application. Although a linear model is proposed for a suspension system for control purposes [11–14], the MPSO can be applied well to identify the non-linear systems, as well. It should be noticed that a suspension system operates under various operating conditions, where parameter variations are unavoidable. Accurate knowledge of these parameters is important to form the control laws. Therefore, it is of our interest to investigate an efficient model parameter tracking approach to achieve precise modeling results under different conditions without using complicated model structures.

In this paper, the MPSO is compared to GA and BPSO in offline parameter identification of suspension system. It can be shown that the MPSO has a better performance than the aforementioned algorithms in solving the parameter estimation of suspension system. Because of the superiority of MPSO in offline identification, it can be used for online parameter identification of suspension system, as well. In the propose method, the estimated parameters will not be updated unless any changes in system parameters is detected by algorithm. A sentry particle is introduced to detect any change in system parameters. If a change is detected, the algorithm scatters the particles around the global best position and forces the algorithm to forget its global memory, then runs the MPSO to find the new values for parameters. Therefore, MPSO runs further iterations if any changes in parameters are detected.

The rest of the paper is organized as follow: Next section describes problem description. Section 3 introduces optimization algorithms. The proposed algorithms in both offline and online parametric identification are presented in Section 4. Simulation results are shown in Section 5. Finally, conclusion and future works are presented in Section 6.

2. Problem Description

This section presents a quarter-car model of suspension system and a proper fitness function for optimization algorithms.

2.1 Suspension System Dynamics

Modeling of vehicle suspension system has been studiedfor many years. In order to simplify the model, a quarter-car model was introduced in response the vertical force for the suspension system [15] as shown in **Figure 1**. In this figure, b is damping coefficient, m_1 and m_2 are unsprung and sprung mass, respectively, k_1 and k_2 are tire

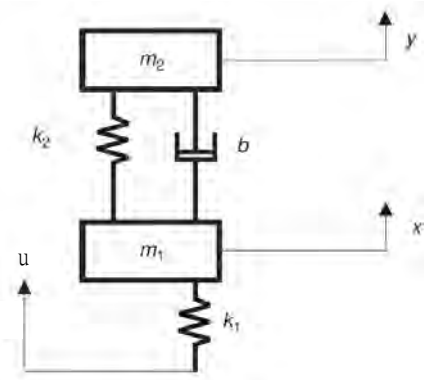


Figure 1. Schematic diagram of the quarter-car model

and suspension stiffness, respectively, u is the road displacement and y is the vertical displacement of sprung mass. The linearized dynamic equations at equilibrium point with an assumption that the tire is in contact with the road are given as:

$$m_1 \frac{d^2x}{dt^2} = k_2(y-x) + b\left(\frac{dy}{dt} - \frac{dx}{dt}\right) + k_1(u-x) \quad (1)$$

$$m_2 \frac{d^2y}{dt^2} = -k_2(y-x) - b\left(\frac{dy}{dt} - \frac{dx}{dt}\right) \quad (2)$$

2.2 Problem Statement

When the model of system is fixed through identification procedure, the parameter identification problem can be treated as an optimization problem. The basic idea of parameter estimation is to compare the system responses with the parameterized model based on a performance function giving a measure of how well the model response fits the system response. Moreover, a common rule in identification is to use excitation signals that correspond to a realistic excitation of the system such that the identified linear model is a good approximation of the system for that type of excitation. Consequently, in order to estimate the system parameters, excitation signal is chosen Gaussian band-limited white noise. The bandwidth is set to 50 Hz, which is sufficiently higher than the desired closed-loop bandwidth [14].

Considering **Figure 2** the excitation input is given to both the real system and the estimated model. Then, the outputs from the real system and its estimated model are input to the fitness evaluator, where the fitness will be calculated. The sum of squares error between real and estimated responses for a number of given samples is considered as fitness of estimated model. So, the fitness function is defined as follow:

$$\mathrm{SSE} = \sum_{k=1}^{N} e^2 = \sum_{k=1}^{N} \left(y(kT_s) - \hat{y}(kT_s)\right)^2 \quad (3)$$

where N is the number of given sampling steps, $y(kT_s)$

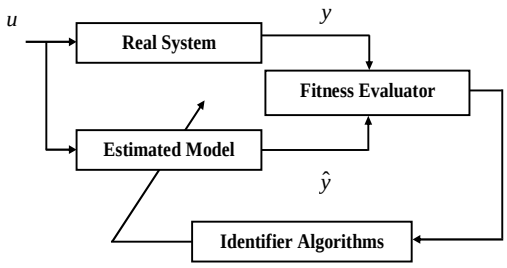


Figure 2. The estimation process

and $\hat{y}(kT_s)$ are real and estimated values in each sample time, respectively. The calculated fitness is then input to the identifier algorithms, *i.e.* GA- BPSO and MPSO, to identify the best parameters for estimated system in fitting procedure by minimizing the sum of square of residual errors in response to excitation input.

3. Optimization Algorithms

As mentioned before, the parameter identification problem can be treated as an optimization problem. The proposed MPSO optimization algorithm is compared with frequently used algorithms in optimization problems, namely GA and BPSO in the optimization problem in hand. These algorithms are taken from two main optimization groups namely evolutionary algorithms (EAs) and swarm intelligence (SI). These algorithms are currently used for numerical optimization problems of stochastic search algorithms.

3.1 Evolutionary Algorithms (EAs)

EAs algorithms are population based, instead of using a single solution. EAs mimic the metaphor of natural biological evolution. EAs operate on a population of potential solutions applying the principle of survival of the fittest to produce better approximations to a solution. At each generation, a new set of approximations is created by two processes. First, selecting individuals according to their level of fitness in the problem domain. Second, breeding them together using operators borrowed from natural genetics.

This process leads to the evolution of populations of individuals that are better suited to their environment than they were created from, just as in natural adaptation. Evolutionary algorithms model natural processes, such as selection, recombination, mutation, migration, locality and neighborhood. The majority of the present implementations of EA come from any of these three basic types, which are strongly related although independently developed: Genetic Algorithms (GA), Evolutionary Programming (EP) and Evolutionary Strategies (ES). Hence, in this paper, the proposed method is compared to GA.

3.2 Swarm Intelligence (SI)

SI is the artificial intelligence based on the collective behavior of decentralized and self-organized systems. SI systems are typically made up of a population of simple agents interacting locally with one another and with their environment. The agents follow very simple rules. Although there is no centralized control structure dictating how individual agents should behave, local interactions between such agents lead to the emergence of complex global behavior. Natural examples of SI include ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling. Among them, PSO is a new and frequently used SI technique.

3.2.1 Basic PSO

PSO is used to search for the best solution by simulating the movement and flocking of birds [16]. The algorithm works by initializing a flock of birds randomly over the searching space, where every bird is called as a "particle". These "particles" fly with a certain velocity and find the global best position after some iteration. At each iteration, each particle can adjust its velocity vector based on its momentum and the influence of its best position as well as the best position of the best individual. Then, the particle flies to a new computed position. Suppose that the search space is n-dimensional, and then the position and velocity of particle i are represented by $X_i = [x_{i1},\ x_{i2},\ \ldots\ , x_{in}]^T$ and $V_i = [v_{i1},\ v_{i2},\ \ldots\ , v_{in}]^T$, respectively. The fitness of each particle can be evaluated according to the objective function of optimization problem. The best previously visited position of the particle i is noted as its personal best position denoted by $P_i = [p_{i1},\ p_{i2},\ \ldots\ , p_{in}]^T$. The position of the best individual of the swarm is noted as the global optimum position $G = [g_1,\ g_2,\ \ldots\ , g_n]^T$. At each step, the velocity of particle and its new position will be assigned as follows:

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (P_i - X_i) + c_2 r_2 (G - X_i) \quad (4)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (5)$$

where ω is called the inertia weight that controls the impact of previous velocity of particle on its current one. r_1 and r_2 are independently uniformly distributed random variables in a range of [0,1]. c_1 and c_2 are positive constant parameters called acceleration coefficients which control the maximum step size. In the references [17,18], several strategies of inertial weight ω were given. Generally, the inertial weight ω should be reduced rapidly in the beginning stages of algorithm but it should be reduced slowly around optimum. If the velocity exceeds the predefined limit, another restriction called $V_{\max}$ is used.

In BPSO, (4) is used to calculate the new velocity ac-

cording to its previous velocity, the distance of its current position from both its own personal best position and the global best position of the entire population. Then the particle flies toward a new position according (5). This process is repeated until a stopping criterion is reached.

3.2.2 The Proposed Modified PSO

As mentioned before, possible trapping in local minima when handling some complex or multimode functions is a shortage of BPSO [19–21]. Hence, the motivation of the proposed method is to overcome this drawback. In BPSO, as time goes on, some particles become quickly inactive because their states are similar to the global optimum. As a result, they lose their velocities. In the subsequent generations, they will have less contribution to the search task due to their very low global search activity. In turn, this will induce the emergence of a state of premature convergence.

To deal with the problem of premature convergence, several investigations have been undertaken to avoid the premature convergence. Among them, many approaches and strategies are attempted to improve the performance of PSO by variable parameters. A linearly decreasing weight into PSO was introduced to balance the global exploration and the local exploitation in [17]. PSO was further developed with time-varying acceleration coefficients to modify the local and the global search ability [18]. A modified particle swarm optimizer with dynamic adaptation of inertia weight was presented [19]. Moreover, some approaches aim to divert particles in swarm among the iterations in algorithm. Mutation PSO employed to improve performance of PSO [20]. The concepts of ‘‘subpopulation” and ‘‘breeding” adopted to increase the diversity [21]. An attractive and repulsive PSO developed to increase the diversity [22].

In this paper, a modified particle swarm optimization (MPSO) algorithm is proposed to avoid premature convergence and increase the convergence speed of algorithm. In our proposed method, after some iteration, the algorithm measures the search ability of all particles and mutates a percentage of particles which their search ability is lower than the others. Our motivation is that particles with low search ability become inactive as their fitness do not grow and need to mutate for getting a chance to search new areas in solution space, which may not been meet already. Also the mutation rate is not constant and if the global best doesn't grow, the rate of mutation is increased. If the fitness of global optimum does not grow, the algorithm can get stuck in local minima forever or at least for some iteration, which lead to a slow convergence speed. In the other words, if the global best of the present population is equal to that of the previous population (solution converges), mutation rate is set to a higher value P_{mh}, such that the diversity of particles is increased so to avoid premature convergence. Otherwise (solution diverges), P_m is set to a lower value P_{ml}, since the population already has enough diversity. The adaptive mutation rate scheme is described as

$$P_m = \begin{cases} P_{mh} & ;\text{if } G(t) = G(t\text{-}1) \\ P_{ml} & ;\text{if } G(t) > G(t\text{-}1) \end{cases} \tag{6}$$

where P_{mh} and P_{ml} are 0.2 and 0.1, respectively. Consequently the mutation rate will be increased by 0.1, if the global optimum doesn't grow, until a growth on the fitness of global optimum occurs. In order to measure the search ability for particle i at each iteration, we take advantage of the fitness increment of the local optima in a designated interval ∇T, from iteration $t-\nabla T$ to t. consequently; the parameter C_i is used to measure the search capability of particle i.

$$C_i = \left| F(P_i(t)) - F(P_i(t-\nabla T)) \right| \tag{7}$$

where $F(P_i(t))$ is the fitness of the best position for i-th particle in t-th iteration $P_i(t)$, and ∇T is a designated interval. Particles with low search ability (small C_i) have low increment in the best local value for their low global search activity and so in our algorithm, they have a bigger change to mutate to get a chance to search a new area in the search space.

Generally, In MPSO algorithm, after iteration ∇T, all particles are sorted according to their C parameter. Then the swarm is divided into two parts: The active part including the top $(1-P_m)\times S$ with higher C and the inactive part consisting of the rest $P_m \times S$ particles with smaller C whereas S is size of the swarm. Particles in the first part update their velocities and position the same as BPSO algorithm. Finally, in order to increase the diversity of the swarm, the inactive particles are chosen to mutate by adding a Gauss random disturbance to them as follows:

$$x_{ij} = x_{ij} + \beta_{ij} \tag{8}$$

where x_{ij} is the j-th component of the i-th inactive particle. β_{ij} is a random variable, which follows a Gaussian distribution with a mean value of zero and a variance value of 1, namely $\beta_{ij} = N(0,1)$. This strategy prevents all particles to divert from the local convergence. Instead, only inactive particles are mutated.

4. Implementation of the MPSO

In this section, the procedure of MPSO in online and offline system parameter identification is described.

4.1 Offline Identification

MPSO algorithm is applied to find the best system parameter, which simulates the behavior of dynamic system. Each particle represents all parameters of estimated model. The procedure for this algorithm can be summarized as follows:

Step 1: Initialize positions and velocities of a group of particles in an M-dimensional space as random points where M denotes the number of system parameters;

Step 2: Evaluate each initialized particle's fitness value using (3);

Step 3: Set P_i as the positions of current particles while G is the global best position of initialized particles. The best particle of current particles is stored;

Step 4: The positions and velocities of all particles are updated according to (4) and (5), and then a group of new particles are generated;

Step 5: Evaluate each new particle's fitness value. If the new position of *i*-th particle is better than P_i, set P_i as the new position of the *i*-th particle. If the fitness of best position of all new particles is better than fitness of G, then G is updated and stored;

Step 6: If iteration $> \Delta T$, calculate the mutation rate (P_m) and search ability of each particle using (6) and (7), respectively. Then mutate $P_m \times S$ number of particle with lower search ability;

Step 7: Update the velocity and location of each particle according to the (4) and (5). If a new velocity is beyond the boundary $[V_{\min}, V_{\max}]$, the new velocity will be set as $V_{\min}$ or $V_{\max}$;

Step 8: Output the global optimum if a stopping criteria is achieved, else go to Step 5.

When a stopping criterion is occurred, the global optimum is the best answer for the problem in hand (the best estimated system parameters).

4.2 Online Identification

The proposed algorithm sequentially gives a data set by sampling periodically. The optimized values of parameters for the first data set are determined by using a procedure described in Subsection 4.1. The estimated parameters will not be updated unless a change in the system parameters is detected. In order to detect any change in system parameters, the global optimum in the later period is noticed as a sentry particle. In each period, the sentry particle is evaluated at first and if the fitness of the sentry particle in the current period is bigger than the previous one, the changes in parameters are confirmed. If no changes are detected, the algorithm leaves this period without changing the positions of particles. When any changes in parameters occur, the algorithm runs further to find the new optimum values. For this purpose, a new coefficient (Δ) is introduced as follows:

$$\Delta = \frac{fitness(\mathrm{S_p}(i)) - fitness(\mathrm{S_p}(i-1))}{fitness(\mathrm{S_p}(i))} \quad (0 \le \Delta < 1) \tag{9}$$

where $\mathrm{S_p}(i)$ is the sentry particle in the *i*-th period. Δ will be bigger than zero if the fitness of the sentry particle at the current period is bigger than the previous one. Thus, changes in model parameters are detected by inspecting Δ at each period. In this case, the particles in population must forget their current global and personal memories in order to find the new global optimum. The fitness of global optimum particle and personal bests of all particles are then evaporated at the rate of a big evaporation constant. As a result, other particles have a chance of fitness bigger than the previous global optimum. Moreover, the velocities of particles are increased to search in a bigger solution space for new optimal solution. When a change in system parameters is detected, the following changes are considered.

$$fitness(P_i) = fitness(P_i) \times T \quad i = 1,..., S \tag{10}$$

$$fitness(\mathrm{G})_{new} = fitness(\mathrm{G})_{old} \times T \tag{11}$$

$$V_{new} = V_{old} + \Delta \times V_{\max} \tag{12}$$

T is an evaporation constant. Also (12) shows that the velocity of particles increase by $\Delta \times V_{\max}$ only in one iteration. Notice that a bigger Δ, *i.e.* greater changes in parameters, causes a bigger velocity. This means that if significant changes in system parameters occur, the particles must search a bigger space and if a little change occurs, particle search around the previous position to find the new position. This strategy accelerates convergence speed of the algorithm, which is an important issue in online identification.

5. Simulation Results

In this section the proposed MPSO algorithm is applied to identify parameters of a suspension system, which its nominal parameters are summarized in **Table 1** [15]. In order to show the performance of the proposed MPSO in

Table 1. Suspension parameters [15]

Parameters	Nominal value
m_1	26 kg
m_2	253 kg
k_1	90000 N/m
k_2	12000 N/m
b	1500 N·sec/m

the problem in hand, it is compared to two frequently used optimization algorithms, including GA and BPSO. Simulation results have been carried out in two parts.

In the first part, in order to show the effectiveness of the proposed MPSO in offline identification, it has been compared with GA and BPSO. In the second part, the proposed MPSO is applied to online parameter identification for suspension system. In both BPSO and MPSO algorithms, $c_1 = c_2 = 2$, and the inertia weight is set to 0.8. Also, the simulation results are compared with GA, where the crossover probability P_c and the mutation probability P_m are set to 0.8 and 0.1, respectively.

5.1 Offline Parameter Identification

Owing to the randomness of the heuristic algorithms, their performance cannot be judged by the result of a single run. Many trials with different initializations should be made to acquire a useful conclusion about the performance of algorithms. An algorithm is robust if it gives consistent result during all the trials. The searching ranges are set as follows:

$20 \le m_1 \le 30$, $200 \le m_2 \le 300$, $85000 \le k_1 \le 90000$, $10000 \le k_2 \le 15000$, $1200 \le b \le 1700$.

In order to run BPSO, MPSO and GA algorithms, a population with a size of 10 for 100 iterations is used.

Comparison of results on sum of squares error resulted from 20 independent trials with N = 1000, 2000 and 2500 are shown in **Tables 2–4**, respectively. This comparison shows that the MPSO is superior to GA and BPSO. Moreover, MPSO is significantly more robust than other algorithms because the best and the mean values obtained by MPSO are very close to the worst value. In addition, the convergence speed of GA, BPSO and MPSO are compared. **Figure 3** shows the convergence speed of these algorithms during 100 iterations which proves that the convergence speed of the proposed MPSO is faster than GA and BSO which can be conclude that MPSO is more proper than aforementioned algorithms. **Figure 4** confirms the success of optimization process by using MPSO algorithm. The identified parameters are m_1, m_2, k_1, k_2 and b, respectively. In this figure, the data set is formed by 1000 samples. In addition, to compare computational time of these algorithms, a threshold of 10^{-5} is considered as stopping condition, in contrast to a predefined number of generation. Then each algorithm runs 20 times and the average of elapsed time is considered as a criteria for computational time. **Table 5** illustrates the results obtained by GA, BPSO, and MPSO. It is clearly obvious that, the proposed algorithm spends extremely fewer iteration and less computational time to reach a predefined threshold as compared with other algorithms. Hence, it can be concluded that IPSO is more proper than

Table 2. Comparison of GA, BPSO and MPSO in offline identification for N=1000

	SSE		
	Best	Mean	Worst
GA	3.11×10^{-6}	2.11×10^{-4}	3.1117×10^{-3}
BPSO	6.45×10^{-8}	1.78×10^{-6}	2.18×10^{-5}
MPSO	3.12×10^{-10}	1.64×10^{-9}	8.46×10^{-9}

Table 3. Comparison of GA, BPSO and MPSO in offline identification for N = 2000

	SSE		
	Best	Mean	Worst
GA	6.98×10^{-6}	1.24×10^{-3}	4.65×10^{-3}
BPSO	7.75×10^{-8}	3.32×10^{-6}	5.13×10^{-5}
MPSO	2.32×10^{-9}	8.68×10^{-9}	6.98×10^{-8}

Table 4. Comparison of GA, BPSO and MPSO in offline identification for N = 2500

	SSE		
	Best	Mean	Worst
GA	7.24×10^{-6}	2.31×10^{-3}	1.12×10^{-2}
BPSO	8.12×10^{-7}	5.21×10^{-6}	4.65×10^{-5}
MPSO	5.43×10^{-9}	1.95×10^{-8}	7.63×10^{-8}

Table 5. Iterations and time required

Algorithm	GA	BPSO	MPSO
Iterations	182	121	49
Elapse Time (*sec*)	28.21	10.34	4.69

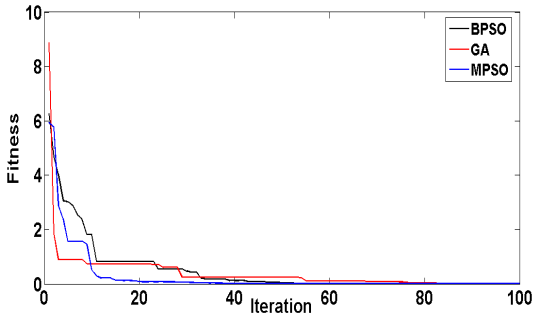
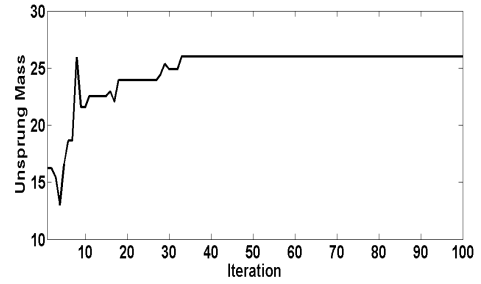
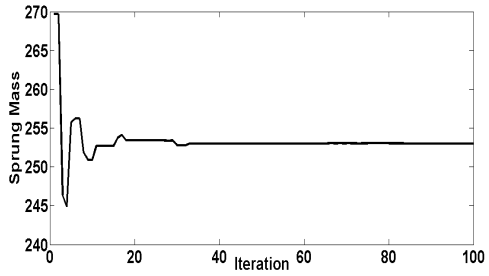


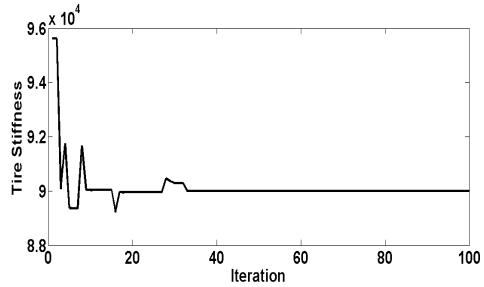
Figure 3. Comparison of convergence speed for GA, BPSO and MPSO



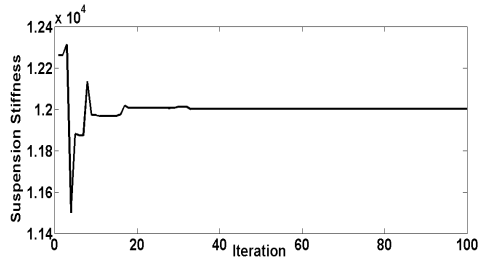
(a) Unsprung mass m_1



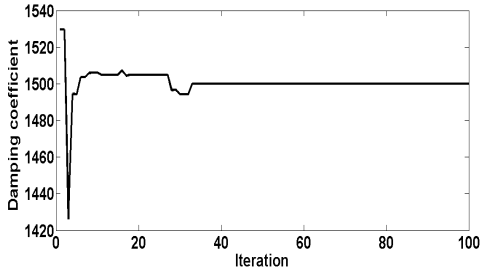
(b) Sprung mass m_2



(c) Tire stiffness k_1



(d) Suspension stiffness k_2



(e) Damping coefficient b

Figure 4. MPSO process for identification of suspension parameters

aforementioned algorithms in terms of accuracy and convergence speed.

5.2 Online Parameter Identification

Based on the pervious section, the MPSO has more accuracy and faster convergence speed than GA and BPSO in off-line identification. Because of this, the proposed method is applied for online identification of suspension system parameters. During online simulation, the sampling frequency is set to 100 kHz such that 1000 pairs of data are sampled within 0.01 msec in each period to form a data set. If a change in the model parameter is detected by sentry particle in a period, the MPSO continues to run. When the fitness of global best becomes lower than a threshold, the simulation for this period is then stopped. There will be no MPSO iteration unless another change in system parameter detect.

Figure 5 shows the fitness evaluation of the proposed method when some changes in system parameters are occurred. First nominal values of parameters are used and MPSO detects these parameters after 37 iterations for a threshold 10^{-5}. If changes in parameters occur the MPSO algorithm runs further. To show the performance of the proposed method in tracking time-varying parameters, two sudden changes are applied to suspension parameters. At the first stage, damping coefficient is changed from 1500 to 1550. At the second stage, tire stiffness is varied from 90000 to 95000. It can be seen that after the first change the algorithm detects new optimal parameters after only 19 iterations. And, after the second change the algorithm finds the new optimal parameters after only 27 iterations. It can see that the proposed method can track any change in parameters. Also since the particles are scattered around the previous global optimum depending on the values of changes in parameters, the new global optimum is found fast. **Figures 6** and **7** show the online identification results of the proposed algorithm when k_1 and b are considered as time varying parameters. It can be seen that the proposed approach can identify time-varying parameters successfully. The dashed lines in **Figures 5–7** signify the moment that the sentry particle has detected some change in system parameters.

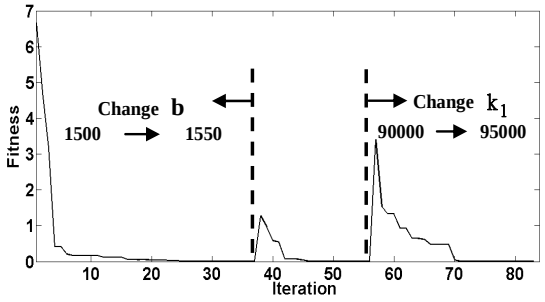


Figure 5. MPSO process in online identification of suspension system

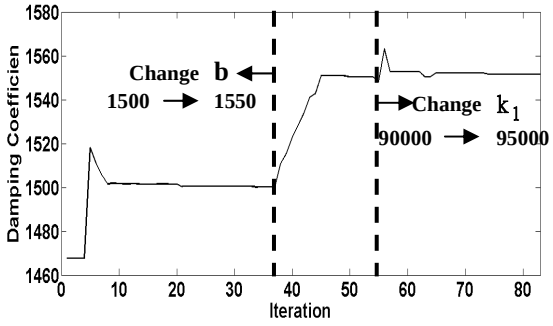


Figure 6. Identifying a time-varying damping coefficient parameter by MPSO

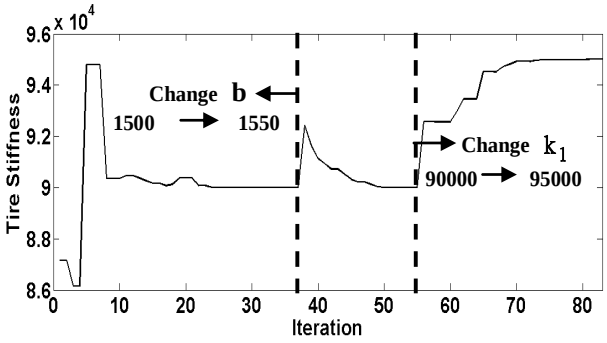


Figure 7. Identifying a time-varying tire stiffness parameter by MPSO

6. Conclusions

A quarter-car model of suspension system was used to show the effectiveness of MPSO in system identification. It has been shown that MPSO is superior to GA and BPSO in offline identification. Owing these attractive features, MPSO is applied to online identification. The estimated parameter will be updated only if a change in system parameters is detected. Thus, the proposed algorithm is a promising particle swarm optimization algorithm for system identification. Future works in this area willinclude considering variable parameters in nonlinear suspension model.

7. Acknowledgments

Our Sincere thanks to Mr. Hamidreza Modarress for his helpful comments and his advice to improve this research work.

REFERENCES

[1] L. Ljung, "System identification: Theory for the user," Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

[2] K. Godfrey and P. Jones, "Signal processing for control," Springer-Verlag, Berlin, 1986.

[3] S. A. Billings and H. Jamaluddin, "A comparison of the back propagation and recursive prediction error algorithms for training neural networks," Mechanical Systems and Signal Processing, Vol. 5, pp. 233–255, 1991.

[4] K. C. Sharman and G. D. McClurkin, "Genetic algorithms for maximum likelihood parameter estimation," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Glasgow, pp. 2716–2719, 23–26 May 1989.

[5] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," International Journal of Control, Vol. 50, pp. 1873–1896, 1989.

[6] R. K. Ursem and P. Vadstrup, "Parameter identification of induction motors using stochastic optimization algorithms," Applied Soft Computing, Vol. 4, pp. 49–64, 2004.

[7] L. Liu, W. Liu, and D. A. Cartes, "Particle swarm optimization-based parameter identification applied to permanent magnet synchronous motors," Engineering Applications of Artificial Intelligence, Vol. 21, pp. 1092–1100, 2008.

[8] Z. Wang and H. Gu, "Parameter identification of bilinear system Based on genetic algorithm," Proceedings of the International Conference on Life System Modeling and Simulation, Shanghai, pp. 83–91, 14–17 September 2007.

[9] M. Ye, "Parameter identification of dynamical systems based on improved particle swarm optimization," Intelligent Control and Automation, Vol. 344, pp. 351–360, 2006.

[10] M. Dotoli, G. Maione, D. Naso, and B. Turchiano, "Genetic identification of dynamical systems with static nonlinearities," Proceedings of the IEEE Mountain Workshop on Soft Computing in Industrial Applications, Blacksburg, pp. 65–70, 2001.

[11] M. M. Fateh and S. S. Alavi, "Impedance control of an active suspension system," Mechatronics, Vol. 19, pp. 134–140, 2009.

[12] H. Du and N. Zhang, "H_∞ control of active vehicle suspensions with actuator time delay," Journal of Sound and Vibration, Vol. 301, pp. 236–252, 2007.

[13] S. J. Huang and H. Y. Chen, "Adaptive sliding controller with self-tuning fuzzy compensation for vehicle suspension control," Mechatronics, Vol. 16, pp. 607–622, 2006.

[14] C. Lauwerys, J. Swevers, and P. Sas, "Robust linear control of an active suspension on a quarter car test-rig," Control Engineering Practice, Vol. 13, pp. 577–586, 2005.

[15] H. Peng, R. Strathearn, and A. G. Ulsoy, "A novel active suspension design technique e-simulation and experimental results," Proceedings of the American Control Conference, Albuquerque, pp. 709–713, 4–6 June 1997.

[16] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," Proceedings of the IEEE International Conference on Neural Networks, Perth Vol. 4, pp. 1942–1948, 1995.

[17] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," Proceedings of the Conference on Evolutionary Computation, Alaska, pp. 69–73, 4–9 May 1998.

[18] A. Ratnaweera and S. K. Halgamuge, "Self-organizing

hierarchical particle swarm optimizer with time-varying acceleration coefficient," IEEE Transactions on Evolutionary Computation, Vol. 8, pp. 240–255, 2004.

[19] X. Yang, J. Yuan, J. Yuan, and H. Mao, "A modified particle swarm optimizer with dynamic adaptation," Applied Mathematics and Computation, Vol. 189, pp. 1205–1213, 2007.

[20] N. Higashi and H. Iba, "Particle swarm optimization with Gaussian mutation," Proceedings of 2003 IEEE Swarm Intellihence Symposium, Indianapolis, pp. 72–79, 24–26 April 2003.

[21] M. Lovbjerg, T. K. Rasmussen, and T. Krink, "Hybrid particle swarm optimizer with breeding and subpopulations," Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco, pp. 126–131, 7–11 July 2001.

[22] J. Riget and J. S. Vesterstroem, "A diversity-guided particle swarm optimizer-the ARPSO," Technical Report 2002–02, EVA Life, Department of Computer Science, University of Aarhus, pp. 1–13, 2002.

J. Software Engineering & Applications, 2010, 3: 230-239
doi:10.4236/jsea.2010.33028 Published Online March 2010 (http://www.SciRP.org/journal/jsea)

Applying Neural Network Architecture for Inverse Kinematics Problem in Robotics

Bassam Daya, Shadi Khawandi, Mohamed Akoum

Institute of Technology, Lebanese University, Saida, Lebanon.
Email: b_daya@ul.edu.lb

Received November 12th, 2009; revised December 15th, 2009; accepted December 22nd, 2009.

ABSTRACT

One of the most important problems in robot kinematics and control is, finding the solution of Inverse Kinematics. Inverse kinematics computation has been one of the main problems in robotics research. As the Complexity of robot increases, obtaining the inverse kinematics is difficult and computationally expensive. Traditional methods such as geometric, iterative and algebraic are inadequate if the joint structure of the manipulator is more complex. As alternative approaches, neural networks and optimal search methods have been widely used for inverse kinematics modeling and control in robotics This paper proposes neural network architecture that consists of 6 *sub-neural networks to solve the inverse kinematics problem for robotics manipulators with* 2 *or higher degrees of freedom. The neural networks utilized are multi-layered perceptron* (*MLP*) *with a back-propagation training algorithm. This approach will reduce the complexity of the algorithm and calculation (matrix inversion) faced when using the Inverse Geometric Models implementation* (*IGM*) *in robotics. The obtained results are presented and analyzed in order to prove the efficiency of the proposed approach.*

Keywords**:** *Inverse Geometric Model, Neural Network, Multi-Layered Perceptron, Robotic System, Arm*

1. Introduction

The task of calculating all of the joint angles that would result in a specific position/orientation of an end-effector of a robot arm is called the inverse kinematics problem. In the recent years, the robot control problem has received considerable attention due to its complexity. Inverse kinematics modeling has been one of the main problems in robotics research, there has been a lot of research on the use of neural networks for control The most popular method for controlling robotic arms [1–5].

In inverse kinematics learning, the complexity is in the geometric and non linear equations (trigonometric equations) and in the matrix inversion, this in addition to some other difficulties faced in inverse kinematics like having multiple solutions. The traditional mathematical solutions for inverse kinematics problem, such as geometric, iterative and algebraic, may not lead always to physical solutions. When the number of manipulator degrees of freedom increases, and structural flexibility is included, analytical modeling becomes almost impossible. A modular neural network architecture was proposed by Jacobs *et al*. and has been used by many researches [2,3,5,6].

However, the input-output relation of their networks is continuous and the learning method of them is not sufficient for the non-linearity of the kinematics system of the robot arm.

This paper proposes neural network architecture for inverse kinematics learning. The proposed approach consists of 6 sub-neural networks. The neural networks utilized are multi-layered perceptron (MLP) with a back-propagation training algorithm. They are trained with end-effector position and joint angles.

In the sections that follow, we explain the inverse kinematics problem, and then we propose our neural network approach; we present and analyze the results in order to prove that neural networks provide a simple and effective way to both model the manipulator inverse kinematics and circumvent the problems associated with algorithmic solution methods.

The proposed approach is presented as a strategy that could be reused and implemented to solve the inverse kinematics problems faced in robotics with highest degrees of freedom. The basics of this strategy are explained in details in the sections that follow.

2. Inverse Kinematics Problem

Inverse kinematics computation has been one of the main problems in robotics research. This problem is generally more complex for robotics manipulators that are redundant or with high degrees of freedom. Robot kinematics is the study of the motion (kinematics) of robots. They are mainly of the following two types: *forward kinematics* and *inverse kinematics*. Forward kinematics is also known as direct kinematics. In forward kinematics, the length of each link and the angle of each joint are given and we have to calculate the position of any point in the work volume of the robot. In inverse kinematics, the length of each link and position of the point in work volume is given and we have to calculate the angle of each join. In this section, we present the inverse kinematics problem.

2.1 Inverse Position Kinematics and IGM

The inverse position kinematics (IPK) solves the following problem: "Given the desired position of the robot's hand; what must be the angles at the robot joints?" In contrast to the forward problem, the solution of the inverse problem is not always unique: The same end effector's pose can be reached in several configurations, corresponding to distinct joint position vectors. The conversion of the position and orientation of a robot manipulator end-effector from Cartesian space to joint space is called inverse kinematics problem. For any position in the X-Y plane for a 2R robot, there is a possibility of 2 solutions for any given point. This is due to the fact that there are 2 configurations that might be possible to reach the desired point as **Figure 1**.

The math involved in solving the Inverse Kinematics problem requires some background in linear algebra, specifically in the anatomy and application of transformation matrices.

Therefore, an immediate attempt to solve the inverse kinematics problem would be by inverting forward kinematics equations.

Let's illustrate how to solve the inverse kinematics problem for robot manipulators on a simple example. **Figure 2** shows a simple planar robot with two arms. The underlying degrees of freedom of this robot are the two angles dictating the rotation of the arms. These are labeled in **Figure 2** as θ_1 and θ_2. The inverse kinematics question in this case would be: What are the values for the degrees of freedom so that the end effector of this robot (the tip of the last arm) lies at position (x, y) in the two-dimensional Cartesian space? One straightforward approach to solving the problem is to try to write down the forward kinematics equations that relate (x, y) to the two rotational degrees of freedom, then try to solve these equations. This solution, named **IGM** (Inverse Geometric Model) will give us an answer to the inverse kinematics problem for this robot. The calculation is presented in **Figure 3**.

As it can be seen in the example above, the solutions to

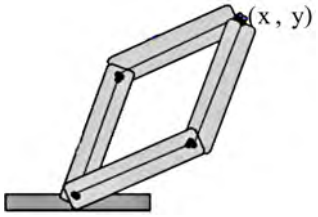


Figure 1. Two solutions depicted for the inverse kinematics problem

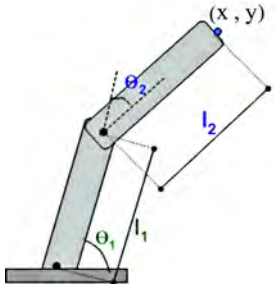


Figure 2. Steer end-effector (x, y) target position

$$x = l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2)$$
$$y = l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2)$$
$$x^2 + y^2 = l_1^2 + l_2^2 + 2 l_1 l_2 cos(\theta_2)$$

$$cos(\theta_2) = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}$$

$$x = l_1 cos(\theta_1) + l_2 (cos(\theta_1) cos(\theta_2) - sin(\theta_1) sin(\theta_2))$$
$$x = cos(\theta_1)(l_1 + l_2 cos(\theta_2)) - sin(\theta_1)(l_2 sin(\theta_2))$$
$$y = cos(\theta_1)(l_2 sin(\theta_2)) + sin(\theta_1)(l_1 + l_2 cos(\theta_2))$$
$$cos(\theta_1) = \frac{x + sin(\theta_1) l_2 sin(\theta_2)}{l_1 + l_2 cos(\theta_2)}$$

$$sin(\theta_1) = \frac{(l_1 + l_2 cos(\theta_2)) y - l_2 sin(\theta_2) x}{l_1^2 + l_2^2 + 2 l_1 l_2 cos(\theta_2)}$$

Figure 3. Finding solutions from the forward kinematics equations

an inverse kinematics problem are not necessarily unique. In fact, as the number of degrees of freedom increases, so does the maximum number of solutions, as depicted in the figure. It is also possible for a problem to have no solution if the point on the robot cannot be brought to the target point in space at all.

While the above example offers equations that are easy to solve, general inverse kinematics problems require solving systems of nonlinear equations for which there are no general algorithms. Some inverse kinematics problems

cannot be solved analytically. In robotics, it is sometimes possible to design systems to have solvable inverse kinematics, but in the general case, we must rely on approximation methods in order to keep the problem tractable, or, in some cases, even solvable.

It is known that there is a finite number of solutions to the inverse kinematics problem. There are, however, 3 types of solutions: complete analytical solution, numerical solutions and semi-analytical solution. In the first type, all of the joint variables are solved analytically according to given configuration data. In the second solution type, all of the joint variables are obtained iterative computational procedures. There are four disadvantages in these: 1) incorrect initial estimations, 2) before executing the inverse kinematics algorithms, convergence to the correct solution cannot be guaranteed, 3) multiple solutions are not known, 4)there is no solution, if the Jacobian matrix is singular. In the third type, some of the joint variables are determined analytically and some computed numerically. Disadvantage of numeric approaches to inverse kinematics problem is also heavy computational calculation and big computational time.

3. Neural Network Approach

The true power and advantage of neural networks lies in their ability to represent both linear and non-linear relationships and in their ability to learn these relationships directly from the data being modeled. Traditional linear models are simply inadequate when it comes to modeling data that contains non-linear characteristics. The most common neural network model is the multilayer perceptron (MLP). This type of neural network is known as a supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the model can then be used to produce the output when the desired output is unknown. A graphical representation of an MLP is shown in **Figure 4**.

The MLP and many other neural networks learn using an algorithm called back propagation. With back propagation, the input data is repeatedly presented to the neural network. With each presentation the output of the neural network is compared to the desired output and an error is computed. This error is then fed back (back propagated) to the neural network and used to adjust the weights such that the error decreases with each iteration and the neural model gets closer and closer to producing the desired output. This process is known as "training".

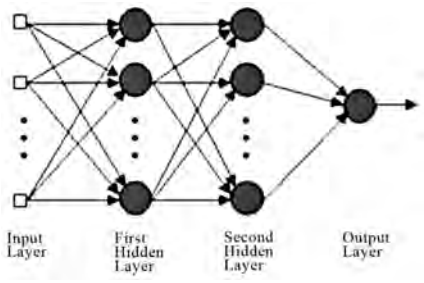


Figure 4. A two hidden layer multiplayer perceptron (MLP)

As mentioned previously, for any position in the X-Y plane for a 2R robot, there is a possibility of 2 solutions, so our approach started by implementing one sub-networks for each solution. For each network, we will try to obtain from the DGM model a series of q_1(or θ_1) and q_2(or θ_2) for a given position of the end-effector. The data training for the 2 sub-networks will be constructed as following:

- Neural Network NN1: (x, y) → (q_1, q_2);
  - q_1 between 0 and 2π, and q_2 between 0 and π.
- Neural Network NN2: (x, y) → (q_1, q_2);
  - q_1 between 0 and 2π, but q_2 between $-\pi$ and 0.

As we are going to use DGM for generating our input parameters for the MLP (q_1 and q_2), for singular configurations, for the same point X and Y, there are two solutions for it within the same aspect. For example, as shown in **Figure 5**, for 2 close positions of (x, y) we have big difference in q_1 values (first value of q_1 is close to 0 and the other q_1 is close to 2π). Thus, this will create a problem during training for our data.

So the above implementation (2 neural networks) leads to a big difficulty in the learning process. In order to solve this problem, we will use 4 Neural Networks (MLP); one for each quadrant.

By implementing 4 Neural Networks (MLP), we prevent two main problems: 1) no more problem to construct the training data for each aspect, and 2) no more problem in the learning phase (one output for one input).

For the error there will be two other Neural Networks:

- One for the Cartesian space (MLP5), used to classify whether the given position is within or not in the accessible region. The error will be representing whether the given solution is within the limitation of the robots.
- One for the joint space (MLP6) to respect the joint limits of our robot.

The approach will use the schematic of the neural network architecture given in **Figure 6**. The DGM model is used to handle the problem of identifying which MLP from the four MLPs is giving the right answer.

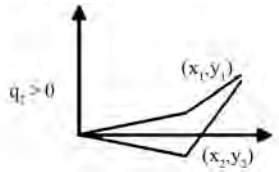


Figure 5. Two close positions for (x, y) but too different values for q_1 (0 and 2π) and this is for the same Neural Network NN1 ($q_2 > 0$)

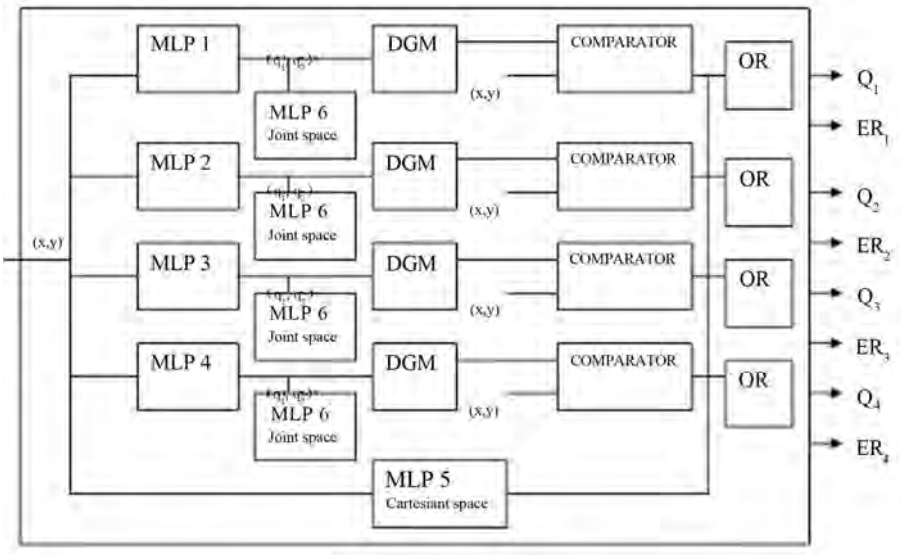


Figure 6. Multi-layer perceptron architecture using 4 MLPs (1 to 4) for the 4 quadrants, MLP5 for the Cartesian space and MLP6 for the joint space. Qi represent the output (q_1, q_2) for MLPi (i = 1 to 4) and ERi represent the error output (if ERi = 0 then Qi accepted; if ERi = 1 then Qi rejected).

4. Training, Experiments and Results

In this section, we present the configuration and preparing of the neural network, and the experiments, with their results, executed in this approach.

We will use 6 MLPs to solve this problem. There will be 1 MLP for each quadrant of the joint space, as shown in **Figure 7**, mainly for q_1 positive and q_2 positive, q_1 negative and q_2 positive, q_1 positive and q_2 negative and lastly q_1 negative and q_2 negative. For the error there will be 2 other neural networks: one to classify whether the given position is within accessible region and one for the joint space.

4.1 Creating MLP Network for IGM of 2R Planar Robot

As mentioned above, for any position in the X-Y plane for a 2R robot, there is a possibility of 2 solutions for any given point. The approach will try to obtain a series of q_1 and q_2 for a given position of the end-effector in the X-Y plane. This is due to the fact that there are 2 configurations that might be possible to reach the desired point.

For this MLP problem, we will be using 12 neurons for the first layer and 8 neurons for the second layer and 2 neurons for the output. However, the difference is that we will be using 6 MLPs to solve this problem. There will be 1 MLP for each quadrant of the joint space, as shown in **Figure 7**, mainly for ($q_1 > 0$ and $q_2 > 0$), ($q_1 > 0$ and $q_2 < 0$), ($q_1 < 0$ and $q_2 > 0$) and lastly ($q_1 < 0$ and $q_2 < 0$).

This is done due to the fact that we are going to use DGM for generating our input parameters for the MLP(q_1 and q_2). For singular configurations, for the same point

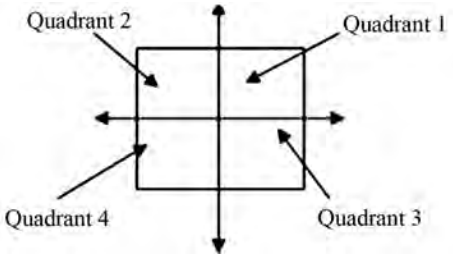


Figure 7. X-Y plane in 4 quadrants

X and Y, there are two solutions for it within the same aspect. For example, for $q_1 = \pi$ and $q_2 = -\pi$, they will have the same X and Y for any value of q_2. Thus, this will create a problem during training for our data. Thus, to solve this problem, we will be using one MLP for each quadrant of data. For the error there is perceptron used to classify whether the given X and Y is within accessible and non accessible region. The error will be representing whether the solution given is within the limitation of the robots. For example, if we give a point inside the non-accessible region of the working space, the error will be 1 and –1 for inside the accessible region.

For each quadrant, we created 10 variables for our target between $q_1 = 0$ to π and equivalently for q_2. Then, we will use the Direct Geometry Model (DGM) function to generate the data training for our MLPs. The tansig function, used in our MLPs, will have its input from +3 to –3 and its output between +1 and –1. The input and output for the MLP is then scaled according to the tansig function.

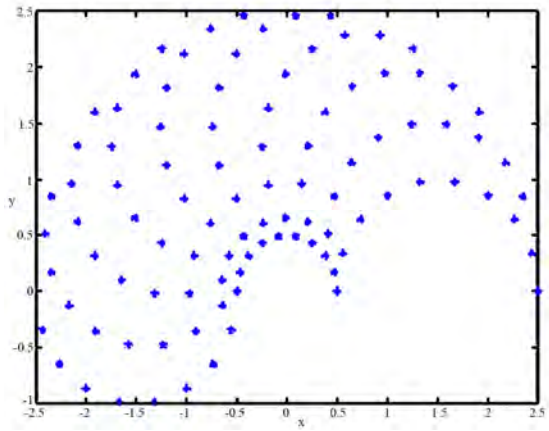
To achieve this, we multiply the input x by 2.5 such that, we will obtain $-3 < x \cdot 2.5 < 3$ (the same for the 4 MLPs). The other scaling is given in **Table 1.**

Table 1. Scaling of the input y and the outputs q_1 and q_2 for each MLP according to the tansig function

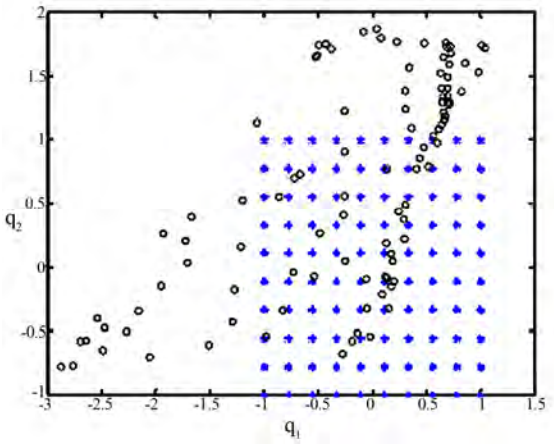
No. MLP	q_1	q_2	Y
MLP1	$-1 < (q_1 \cdot 2/\pi) - 1 < 1$	$-1 < (q_2 \cdot 2/\pi) - 1 < 1$	$-3 < (y \cdot 5) - 2 < 3$
MLP2	$-1 < (q_1 \cdot 2/\pi) + 1 < 1$	$-1 < (q_2 \cdot 2/\pi - 1 < 1$	$-3 < (y \cdot 5) + 2 < 3$
MLP3	$-1 < (q_1 \cdot 2/\pi) - 1 < 1$	$-1 < (q_2 \cdot 2/\pi) + 1 < 1$	$-3 < (y \cdot 5) - 2 < 3$
MLP4	$-1 < (q_1 \cdot 2/\pi) + 1 < 1$	$-1 < (q_2 \cdot 2/\pi) - 1 < 1$	$-3 < (y \cdot 5) + 2 < 3$

4.1.1 The First MLP for the First Quadrant

For the first quadrant, we created 10 variables for our target between $q_1 = 0$ to π and equivalently for q_2. Then, we will use our DGM2R function to generate the input for our MLP. The input and output for the MLP is then scaled according to the tansig function. The tansig function will have its input from +3 to –3 and its output between +1 and -1. To achieve this, will be dividing our output and multiplying our inputs with factors. The resulting input and output of for the MLP is shown in **Figure 8**.



(a)



(b)

Figure 8. (a) Input for first quadrant MLP; (b) target output for first quadrant MLP

It is shown as well in **Figure 8** that the initial result from our MLP prior to training. For our MLP, we will be using "trainlm" function since it is faster and more accurate in producing the result.

The MLP managed to produce accurate data with error of 1.74×10^{-5} within 25 epochs. The result is then plotted back to our target. **Figure 9** shows the result for the MLP restoring the data prior to scaling. From **Figure 9**, we know that our MLP has managed to produce quite an accurate result since the result of the MLP is pretty close to our target values.

For this MLP, we have used learning rate equal to 0.2. The performance of the MLP is shown in **Figure 10.**

4.1.2 The Second MLP for the Second Quadrant

In the second MLP for the second quadrant of the joint limit, we will do the same algorithm for training the MLP. We will input our data in the range of q_1 from 0 to –2.8 and q_2 from 0 to π. We will then input our data to "tansig" transfer function. The inputs and outputs as well the initial output of our MLP are presented in **Figure 11**. Using the

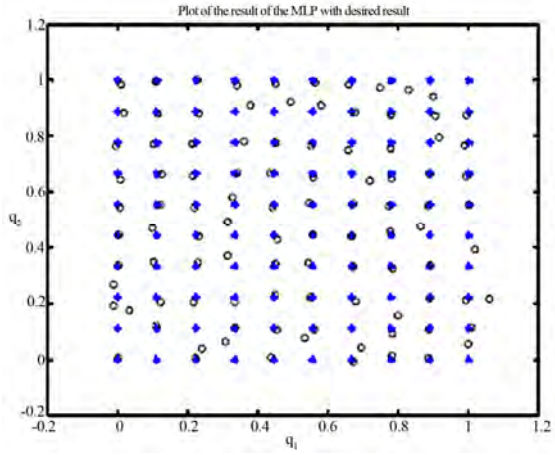


Figure 9. Result of the first quadrant MLP plotted onto the target data

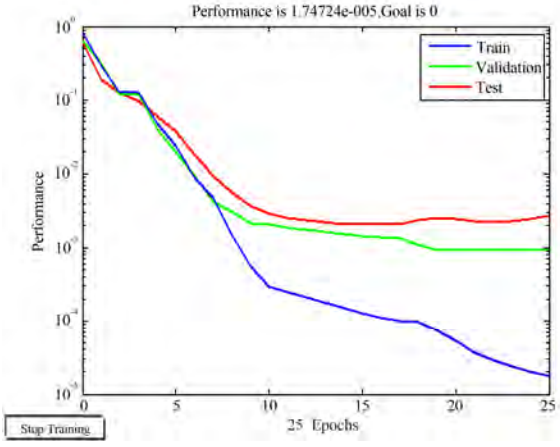


Figure 10. Performance result for this first MLP

"trainlm" function, we managed to obtained accuracy of 1.38×10^{-5} within 41 epochs.

The result of the MLP is presented in **Figure 12**. The result of the MLP presented is prior to rescaling back to the original data. For the third MLP we will be performing the similar operation by scaling the input and the output before inputting it to the MLP to be learnt.

4.1.3 The Third MLP for the Third Quadrant

For the third MLP, we are trying to generate result for q_1 in the range of 0 and π and q_2 in the range of 0 to –2.8. We are using –2.8 because of the requirement of the joint limit present in the system. The initial input and output of the system is presented in the following **Figure 13**.

After training our MLP for 15 epochs, we managed to get an error in performance of 4.64×10^{-5}. The plot of the result and the plot of the outputs are given in the following **Figure 14**.

4.1.4 The Fourth MLP for the Fourth Quadrant

For the last MLP to generate the result for IGM model, we are trying to generate result for q_1 in the range of 0 and –2.8 and q_2 in the range of 0 to –2.8. For the same reason, we are using –2.8 because of the requirement of the joint limit present in the system. The initial input and output of the system is presented in **Figure 15**. After training for 13 epochs, we managed to get an error of 5.24×10^{-5} and the result of the MLP is plotted against the desired result. We can observe that the resulting points from the MLP are close to the desired target. The result of the MLP and the performance of the MLP are presented in **Figure 16**.

4.1.5 The Fifth MLP

The fifth MLP is designed to define the workspace of the robot. The robot workspace is a circle with an internal circle upon which the robot will not be able reach. Thus, there is a limitation to the area upon which the robot is able to access the area. When the given x and y is within the internal circle or outside the working circle as depicted in **Figure 17**, the error of the equation will be 1.

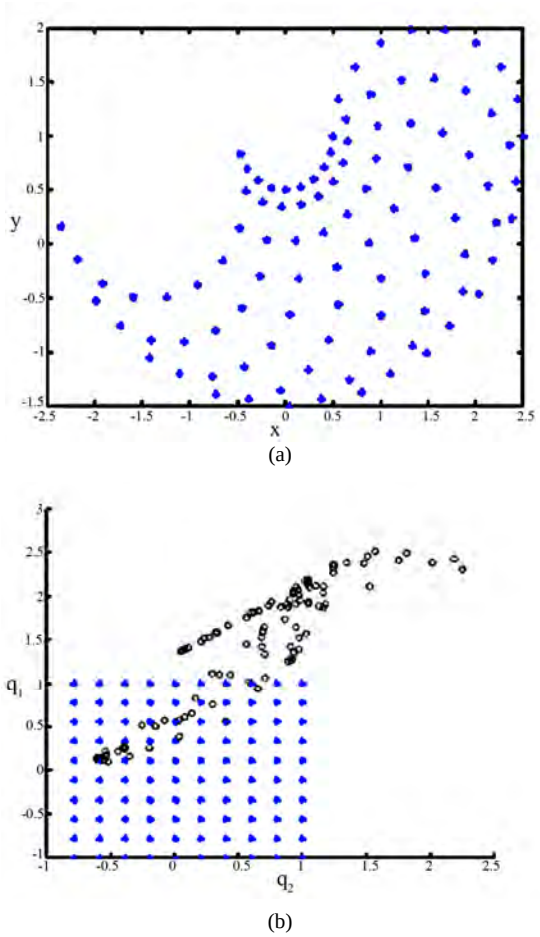


Figure 11. (a) Input data for the second quadrant MLP; (b) output of the second MLP plotted together with the desired result

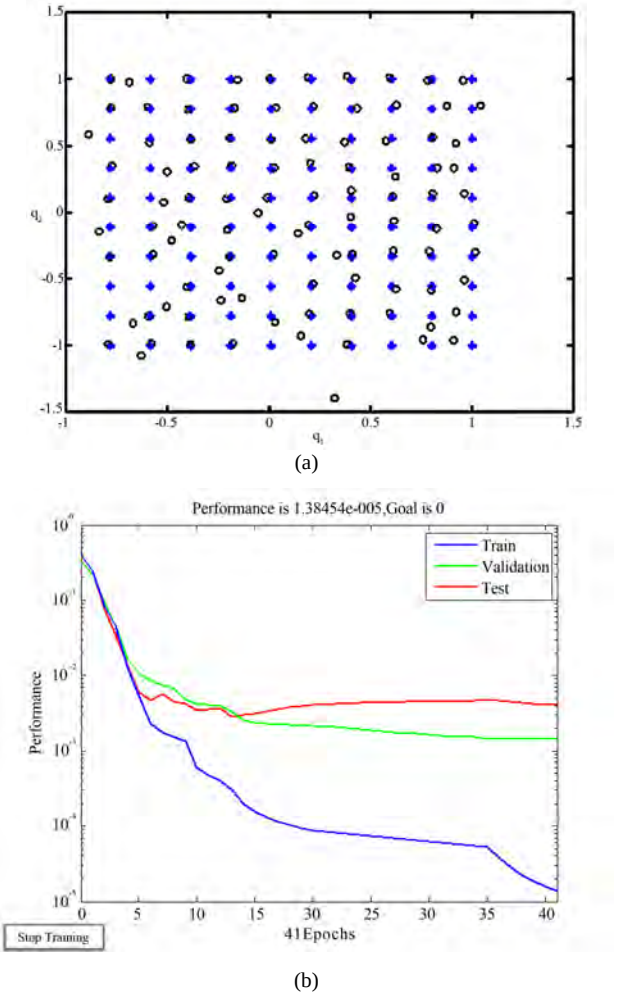
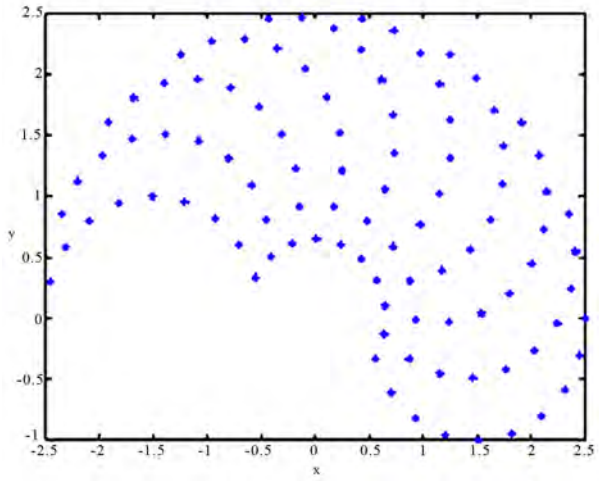
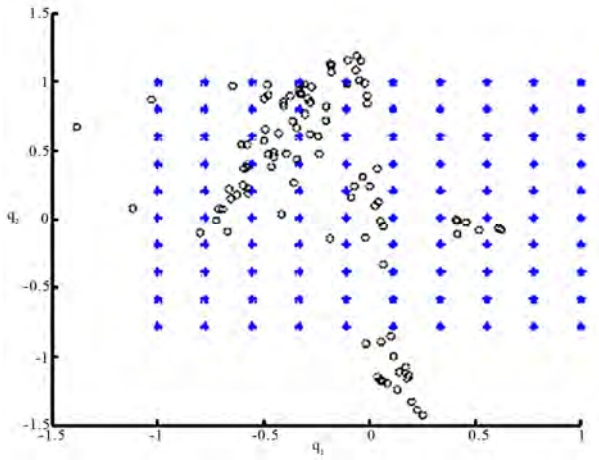


Figure 12. (a) Result of MLP plotted with the desired target; (b) Performance of MLP with trainml function

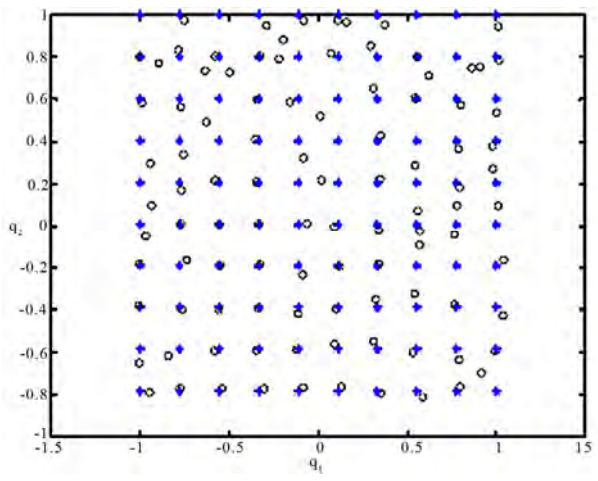


(a)

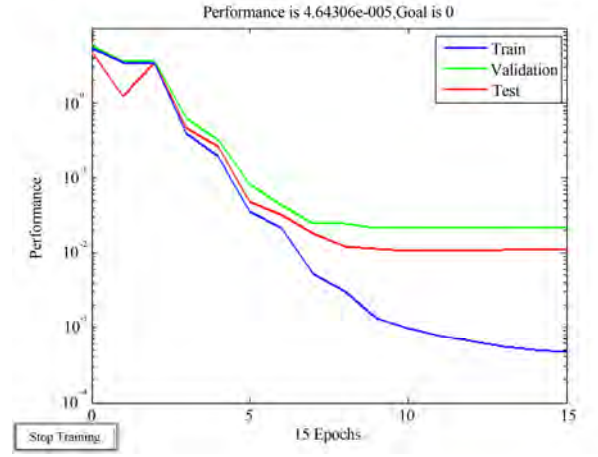


(b)

Figure 13. (a) Input for the 3rd MLP; (b) output of the 3rd MLP plotted with the desired result



(a)



(b)

Figure 14. (a) Output of the result plotted together with the desired target of the 3rd MLP; (b) Performance of the 3rd MLP

In order to do this, we will find the relationship between the length of the robots arms to the radius of its working space. We know that the radius of the large cir cle is given by the formula $R=L_1+L_2$. Thus, we know that within the gray circle, $(R-L_1)^2 = L_2^2$. Expanding the equation, we know that $R^2 - 2L_1L_2 + (L_2^1 + L_2^2) = 0$.

Thus, we notice that if the desired point is within the gray area, the value of the equation above will be less than 0, and otherwise if the value of R is smaller than $L_1 - L_2$ or R is greater than L_1+L_2.

We have created 20 numbers of data of X_1 and X_2 for the input to the MLP.

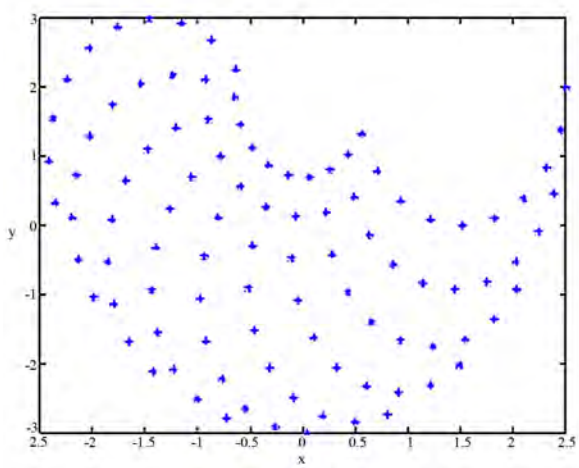
Then, using these inputs, we calculate our desired target using the "error3" function using a notation that if *error is* 1 then the point is not inside working circle and if *error is* 0 then the robot is inside the working circle. The desired target is presented in **Figure 18**. Our result shows that the MLP has managed to classify the classes within 17 epochs with zero error. Thus, this error problem has been solved with only a single perceptron. The result of the MLP is presented in **Figure 18**.

The performance of the perceprton is shown in **Figure 19**.

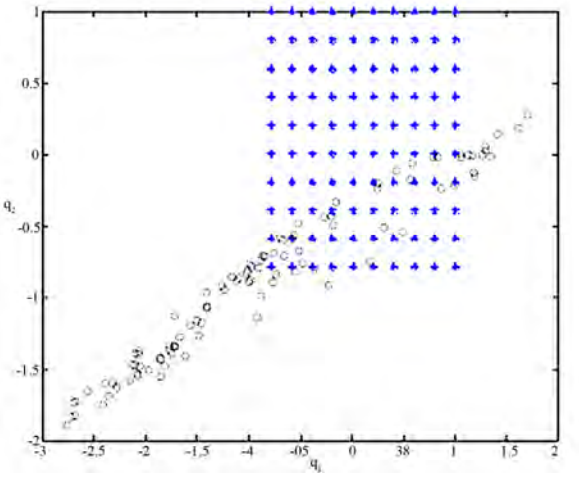
4.1.6 The Sixth MLP

The last step of the classification is to categorize the resulting Q_1 and Q_2 into either [0 0], [0 1] or [1 0]. This means on the other hand, we need to classify the elements of angles in the Q_1 and Q_2. If we draw the boundary limit of the angles, we would be able to find a rectangular area(as shown in **Figure 20**). Certainly, we can apply the method of perceptron with 3 layers for implementing classifier arbitrary linearly limited areas (polyhedron).

Thus, if our MLP is having 4 neurons on the first layer and 1 neuron on the second layer and taking Q_1 and Q_2 as the input parameters for the neuron and output of 1 if Q_1 or



(a)



(b)

Figure 15. (a) Input to the 4th MLP; (b) initial output of the 4th MLP plotted together with the desired result

Q_2 is within the grey area and -1 if it outside the gray area, we should be able to fully classify the problem.

The weights of our neurons are given as follows:

$$W^1 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \text{ and } b^1 = \begin{bmatrix} -3 \\ 2.8 \\ -3 \\ 2.8 \end{bmatrix}$$

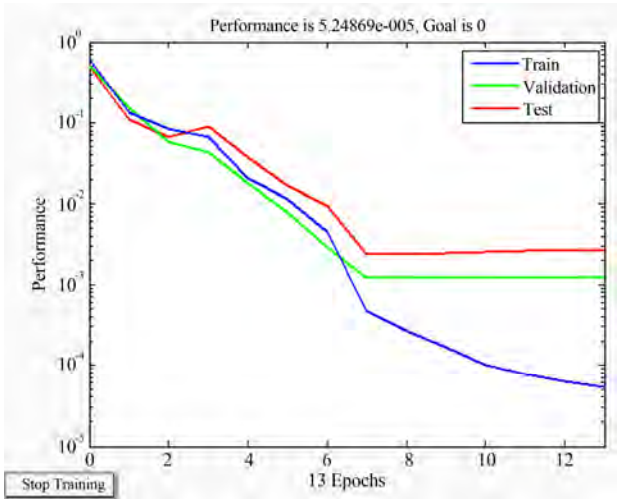
Marking the desired area (grey area), we obtained

	g_1	g_2	g_3	g_4
G_1	-	+	-	+

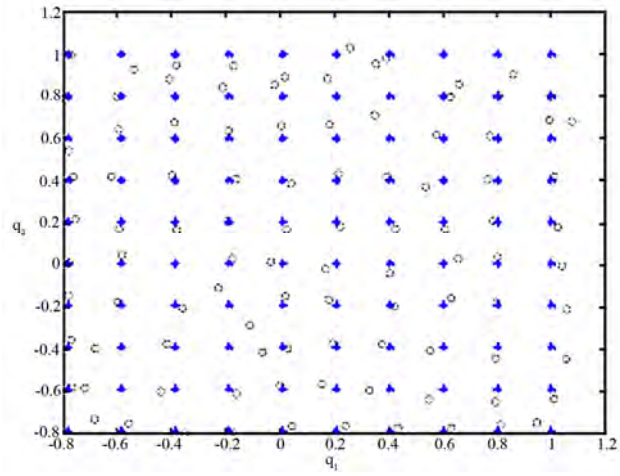
Thus, $W^2 = [-1 \quad 1 \quad -1 \quad 1]$, and $b^2 = [-3]$.

From this weights and biases, our convention is :

- output = 1 if the value of Q is within the joint limit.
- output = -1 if the value of Q is outside the joint limit.



(a)



(b)

Figure 16. (a) Performance of the 4th MLP; (b) Result of the 4th MLP plotted together with the desired target

With these values, we can create a MLP network and we will be able to separate the two results perfectly. The result is shown in **Figure 21**. Lastly, the final step is to combine all the 6 MLP together in a program that we can use to generate the desired Q_1 and Q_2 and error. We will need to classify for the y of the input to our joint network. Initially, when the input is having y > 0, there are two solutions that are possible, which is q_1 is positive and q_2 is positive or negative. Thus, we have to choose quadrant 1 or 4 to obtain a correct result. Otherwise, when y < 0, the two solutions that are possible are q_1 is negative and q_2 is positive or negative. After we have done the classification, then we can use our network to produce the desired result. The program will check whether the given X and Y is within the working circle. If it does not, then the error will be equal to [1 1] and the value of Q_1 and Q_2 will be of a null vector. On the other hand, if the point X, Y is within

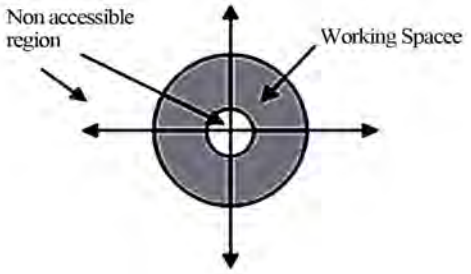
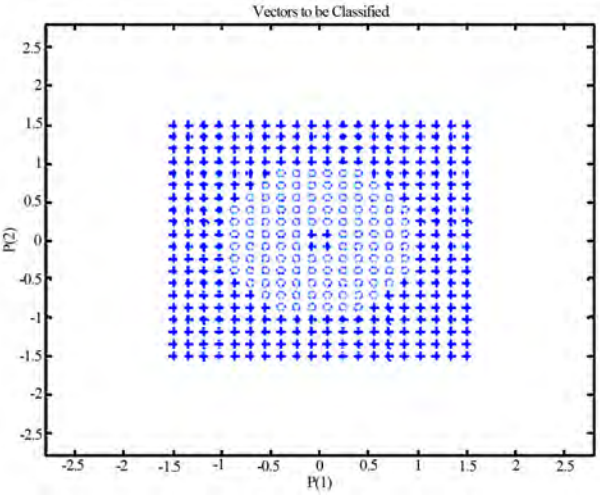
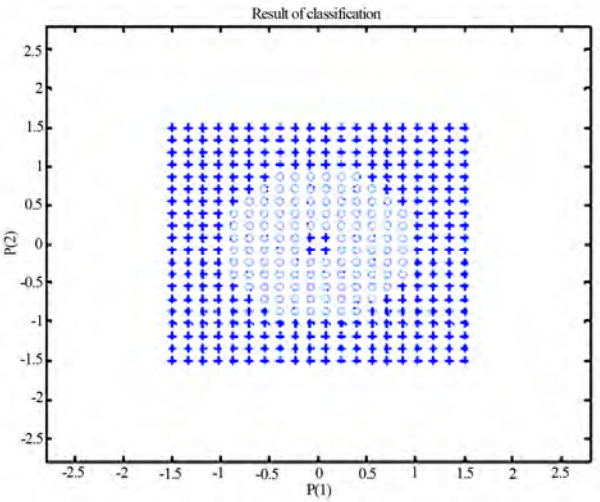


Figure 17. Working space of the robot



(a)



(b)

Figure 18. (a) Desired target result; (b) Result of the MLP

the circle, then it will input the X and Y according to the given area as described above. Then, the resulting result will be fed into the MLP 6 for determining whether the result is within the joint limit or not. If it does, then error will be equal to 0 and if it is not in the joint limit, then error will be equal to 1.

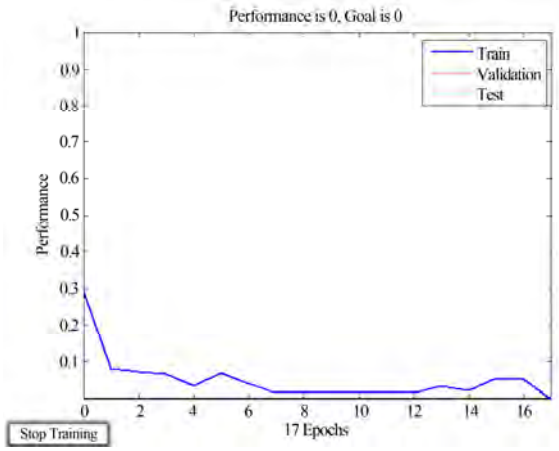


Figure 19. Performance of the perceptron

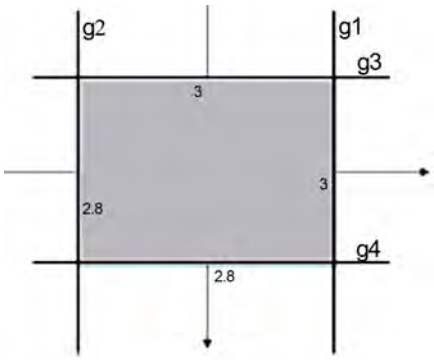


Figure 20. Classification to categorize Q_1 and Q_2

Testing the MLP with values of $X = [0.646 \quad -0.3196]$ which corresponds to $q = [-\frac{\pi}{3} \quad \frac{\pi}{2}]$, we obtained from our MLP,

$$Q_1 = [1.0936 \quad 1.6016]$$

and

$$Q_2 = [2.2538 \quad -1.5618]$$

Our optimum result for the Q_1 and Q_2 from the IGM model is

$$Q_1 = [1.047 \quad 1.570]$$

and

$$Q_2 = [2.2232 \quad -1.5708]$$

We know that the value from of MLP is pretty close to the real values of the IGM2R model. The error is [0 0] and [0 0] respectively.

Testing the MLP with values of

$$X = [-0.3196 \quad -0.6464]$$

which corresponds to $q = [-\frac{5\pi}{6} \quad \frac{\pi}{2}]$, we obtained from our MLP,

$$Q_1 = \begin{bmatrix} -2.6856 & 1.8241 \end{bmatrix}$$

and

$$Q_2 = \begin{bmatrix} -1.4809 & -1.6115 \end{bmatrix}$$

Our optimum result for the Q_1 and Q_2 from the IGM model is

$$Q_1 = \begin{bmatrix} -2.6180 & 1.5708 \end{bmatrix}$$

and

$$Q_2 = \begin{bmatrix} -1.4420 & -1.5708 \end{bmatrix}$$

We know that the value from of MLP is pretty close to the real values of the IGM2R model. The error is [0 0] and [0 0] respectively.

5. Conclusions

In this paper, experimental results on the control of robotic manipulator using neural networks have been provided and it has been demonstrated that neural networks do indeed fulfill the promise of providing model-free learning controllers for robotic systems and provide an excellent alternative for the control of robotic manipulators.

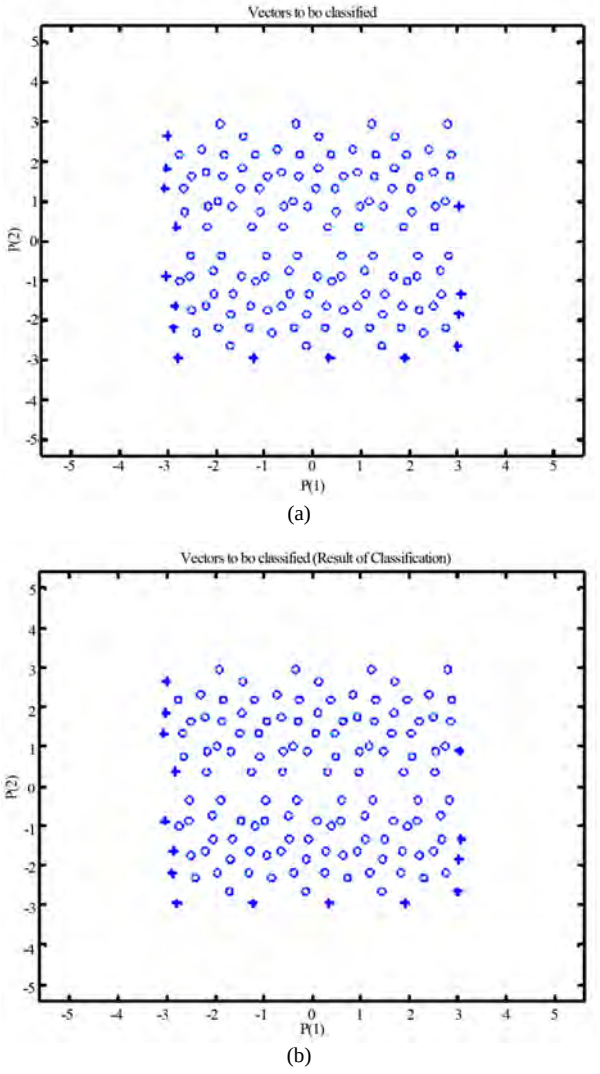


Figure 21. (a) Plot of the input Q with the desired classification; (b) Plot of the input Q with the result of classification

Here, neural network model (MLP) solve the issues faced when the Inverse Geometric Model (IGM) is used, which requires no matrix inversion and iterates directly on the joint position, being thus suitable for on-line application and also preserving repeatability. In other words, Multilayer Networks are applied to the robot inverse kinematics problem. The networks are trained with end-effector position and joint angles. After training, performance is measured by having the network generate joint angles for arbitrary end-effector trajectories.

It is found that neural networks provide a simple and effective way to both model the manipulator inverse kinematics and circumvent the problems associated with algorithmic solution methods.

The proposed approach in the paper can be treated as a strategy to be followed for any other future work in the same domain. Mainly it can be implemented for robotics manipulators that are redundant or with high degrees of freedom. It is useful to mention that, based on the proposed approach; new research has been started lately for applying neural networks approach for 3R robotics.

REFERENCES

[1] B. Choi and C. Lawrence, "Inverse kinematics problem in robotics using neural networks," National Aeronautics and Space Administration, Lewis Research Center, Cleveland, 1992.

[2] R. Köker, C. Öz, T. Çakar, and H. Ekiz , "A study of neural network based inverse kinematics solution for a three-joint robot," Robotics and Autonomous Systems, Vol. 49, pp. 227–234, 2004.

[3] L. Wei, H. Wang, and Y. Li , "A new solution for inverse kinematics of manipulator based on neural network," Machine Learning and Cybernetics, Vol. 2, pp. 1201–1203, 2003.

[4] J. Guo and V. Cherkassky, "A solution to the inverse kinematic problem in robotics using neural network processing", International Joint Conference on Neural Networks, Vol. 2, pp. 299–304, 1989.

[5] D. Pham, M. Castellani, and A. Fahmy "Accountability learning the inverse kinematics of a robot manipulator using the Bees Algorithm," 6th IEEE International Conference on Industrial Informatics, pp. 493–498, 2008.

[6] E. Gallaf , "Multi-fingered robot hand optimal task force distribution: Neural inverse kinematics approach," Robotics and Autonomous Systems, Vol. 54, No. 1, pp. 34–51, 2006.

J. Software Engineering & Applications, 2010, 3: 240-244
doi:10.4236/jsea.2010.33029 Published Online March 2010 (http://www.SciRP.org/journal/jsea)

Quantum Number Tricks

Takashi Mihara

Department of Information Sciences and Arts, Toyo University, Kawagoe, Japan.
Email: mihara@toyonet.toyo.ac.jp

Received November 23rd, 2009; revised December 21st, 2009; accepted December 29th, 2009.

ABSTRACT

Some results indicate that quantum information based on quantum physics is more powerful than classical one. In this paper, we propose new tricks based on quantum physics. Our tricks are methods inspired by the strategies of quantum game theory. In these tricks, magicians have the ability of quantum physics, but spectators have only classical one. We propose quantum tricks such that, by manipulating quantum coins and quantum cards, magicians guess spectators' values.

Keywords**:** *Quantum Trick, Entangled State, Game Theory*

1. Introduction

The studies on *quantum information* have succeeded in such as quantum computation, quantum cryptography, quantum communication complexity, and so on. For example, Shor's quantum factoring algorithm is one of representative results in these fields [1]. In addition, quantum game theory has been also proposed and it has been shown that quantum game theory is more powerful than classical one.

In 1998, for a coin flipping game, Meyer proposed a quantum strategy for the first time and showed that the quantum strategy has an advantage over classical ones [2]. Moreover, he also showed the importance of a relationship between quantum game theory and quantum algorithms.

After that, other types of quantum strategies have been also proposed. For example, Eisert *et al*. proposed a quantum strategy with entangled states for a famous two-player game called the *Prisoner's Dilemma* [3] (also see Du *et al*. [4,5], Eisert and Wilkens [6], and Iqbal and Toor [7]). For another famous two-player game called the *Battle of the Sexes*, Marinatto *et al*. also proposed a quantum strategy with entangled states [8]. For these games, they showed quantum Nash equilibriums different from classical ones.

In this paper, we propose quantum tricks based on methods inspired by the strategies of quantum game theory. Magicians have the ability of quantum physics, but spectators have only classical one. By manipulating quantum coins and quantum cards, magicians guess spectators' values. For example, we propose tricks such that by using entangled states, a magician transmits a spectator's value to another magician without communicating between them.

The remainder of this paper has the following organization. In Section 2, we define notations and basic operations used in this paper. In Section 3, we propose quantum coin tricks. In Section 4, we propose quantum card tricks. Finally, in Section 5, we provide some concluding remarks.

2. Preliminaries

First, we denote some basic notations. Let $\mathbf{B} = \{0,1\}$, $\mathbf{Z}_n = \{0,1,...,n-1\}$, and $\mathbf{Z}_n^+ = \{1,2,...,n-1\}$ for a positive integer n. Let a and b be integers. We say that a is congruent to b to modulus n if n is a divisor of $a-b$ and denote by $a \equiv b (\mathrm{mod}\, n)$, and we denote an inner product modulo 2 of a and b by $a \cdot b$. Finally, let $\oplus$ be an exclusive-OR operator, e.g., $(1,1,0,0) \oplus (1,0,1,0) = (0,1,1,0)$.

Next, we define some basic quantum notations. As states of *qubit*, let $|0\rangle = (1 \quad 0)^T$ and $|1\rangle = (0 \quad 1)^T$, where $|\cdot\rangle$ is Dirac notation and A^T is the transposed matrix of matrix A. Throughout this paper, we take $\mathbf{B}_q = \{|0\rangle, |1\rangle\}$ as a computational basis and a measurement basis. Moreover, we denote an n-qubit basis state by $|b_1\rangle \otimes |b_2\rangle \otimes \cdots \otimes |b_n\rangle = |b_1\rangle |b_2\rangle \cdots |b_n\rangle = |b_1, b_2, \ldots, b_n\rangle$, where $\otimes$ is a tensor product and $|b_i\rangle \in \mathbf{B}_q$ ($i = 1,2,...,n$). In addition, we denote a basis in an N-dimensional system,

a basis of *qudit* states, by $\mathbf{Z}_{N_q} = \{|x\rangle \mid x \in \mathbf{Z}_N\}$, where N (≥ 2) is an integer. We call $|x\rangle$ a *quantum register*.

Finally, we define some unitary matrices used for quantum tricks in this paper. Let I be the 2×2 identity matrix. This operation means no operation. A *Walsh-Hadamard* operation H is

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

($H|0\rangle = (1/\sqrt{2})(|0\rangle + |1\rangle)$ and $H|1\rangle = (1/\sqrt{2})(|0\rangle - |1\rangle)$).

Note that $H=H^{-1}$. This operation is used when we make a superposition of states. An operation used when a coin is flipped is X,

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

($X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$).

Moreover, we define an operation between two qubits. A *Controlled Not* gate, *CNOT*, is

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

($CNOT|c,t\rangle = |c, t \oplus c\rangle$, where the first bit c is the controlled bit and the second bit t is the target bit). We denote the operation by $CNOT_{(ij)}$ when the i-th bit is the controlled bit and the j-th bit is the target bit.

Entangled states can be made by using H and $CNOT$. For example,

$$|0\rangle|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle$$
$$\xrightarrow{CNOT} \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle).$$

Finally, we define two matrices for N-state transition. Let $x \in \mathbf{Z}_N$. A *quantum Fourier transform* [1], *QFT*, is

$$QFT|x\rangle = \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} e^{\imath 2\pi xy/N}|y\rangle,$$

and

$$QFT^{-1}|y\rangle = \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} e^{-\imath 2\pi xy/N}|x\rangle.$$

3. Quantum Coin Tricks

In this section, we show some quantum coin tricks using quantum states. Throughout this paper, we use Alice and Bob as names of magicians, and use Carol and Davis as names of spectators participating in a magic show. Moreover, Alice and Bob can cooperate but cannot communicate with each other during each show.

First, we show a simple coin trick using a two-qubit entangled state.

Coincidence: First, Alice prepares one coin and put it in a box. The box is a container such that no one cannot see the state of the coin but can operate it. Next, Carol flips the coin or not. Then, Bob guesses the state of the coin, *i.e.*, either head (H) or tail (T).

Method of Coincidence

We denote H and T by $|0\rangle$ and $|1\rangle$, respectively.

1) Beforehand, Alice and Bob share an entangled state

$$\frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle),$$

where Alice has the first qubit and Bob has the second qubit. Alice's qubit is in a box.

2) Carol flips Alice's coin or not. This means that Carol applies X to Alice's qubit if she wants to flip the coin; otherwise she applies I to it. Then, if she flips it, the state becomes

$$\frac{1}{\sqrt{2}}(|1\rangle|0\rangle + |0\rangle|1\rangle).$$

3) Alice and Bob apply H to the state. Then it becomes

$$\frac{1}{\sqrt{2}}(|0\rangle|0\rangle - |1\rangle|1\rangle)$$

if Carol flipped the coin; otherwise the state does not change, *i.e.*,

$$\frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle).$$

4) Bob measures his qubit and announces the value(H or T) to Carol.

5) Carol opens the box and confirms that her value is same as Bob's value.

This trick can be easily extended to multiple coins by preparing the entangled states $(1/\sqrt{2})(|0\rangle|0\rangle + |1\rangle|1\rangle)$ corresponding to the number of coins.

Next, we show a trick guessing the number of Carol flipping coins.

Flip-Flop1: First, Alice prepares k coins in all the coins being head. Next, Carol flips some coins such that the state of coins is $m \in \mathbf{B}^k$. Alice flips some coins. Carol flips some coins. Alice flips some coins. Then, Carol finds that the state of final coins is m.

Method of Flip-Flop1

1) Alice prepares a state $|0^k\rangle$ (all the coins are head), exhibits it to Carol, and puts it in a box.

2) Carol flips some coins and the state becomes $|m\rangle$.

3) Alice applies $H^{\otimes k}$ to it and the state becomes

$$\frac{1}{\sqrt{2^k}}\sum_{x=0}^{2^k-1}(-1)^{m\cdot x}|x\rangle.$$

4) Carol flips some coins and the state becomes

$$\frac{1}{\sqrt{2^k}}\sum_{x=0}^{2^k-1}(-1)^{m\cdot x}|x\oplus r\rangle,$$

where $r\in\mathbf{B}^k$.

5) Alice applies $H^{\otimes k}$ to it and the state becomes

$$\begin{aligned}&\frac{1}{\sqrt{2^{2k}}}\sum_{y=0}^{2^k-1}\sum_{x=0}^{2^k-1}(-1)^{m\cdot x}(-1)^{(x\oplus r)\cdot y}|y\rangle\\&=\frac{1}{\sqrt{2^{2k}}}\sum_{y=0}^{2^k-1}(-1)^{r\cdot y}\sum_{x=0}^{2^k-1}(-1)^{(m\oplus y)\cdot x}|y\rangle\\&=(-1)^{r\cdot m}|m\rangle.\end{aligned}$$

6) Carol opens the box and confirms m.

Finally, we show a trick modifying **Flip-Flop1**.

Flip-Flop2: First, Alice prepares k coins in all the coins being head. Next, Carol flips some coins such that the state of coins is $m_1\in\mathbf{B}^k$. Alice flips some coins. Carol flips some coins such that the added state of coins is $m_2\in\mathbf{B}^k$. Alice flips some coins. Carol flips some coins. Alice flips some coins. Then, Alice guesses the value of m_1 if Carol announces the value of m_2; otherwise, Alice guesses the value of m_2 if Carol announces the value of m_1.

Method of Flip-Flop2

1) Alice prepares a state $|0^k\rangle$, exhibits it to Carol, and puts it in a box.

2) Carol flips some coins and the state becomes $|m_1\rangle$.

3) Alice does not flip them in her turn.

4) Carol flips some coins and the state becomes $|m_1\oplus m_2\rangle$.

5) Alice applies $H^{\otimes k}$ to it and the state becomes

$$\frac{1}{\sqrt{2^k}}\sum_{x=0}^{2^k-1}(-1)^{(m_1\oplus m_2)\cdot x}|x\rangle.$$

6) Carol flips some coins and the state becomes

$$\frac{1}{\sqrt{2^k}}\sum_{x=0}^{2^k-1}(-1)^{(m_1\oplus m_2)\cdot x}|x\oplus r\rangle,$$

where $r\in\mathbf{B}^k$.

7) Alice applies $H^{\otimes k}$ to it and the state becomes

$$\begin{aligned}&\frac{1}{\sqrt{2^{2k}}}\sum_{y=0}^{2^k-1}\sum_{x=0}^{2^k-1}(-1)^{(m_1\oplus m_2)\cdot x}(-1)^{(x\oplus r)\cdot y}|y\rangle\\&=\frac{1}{\sqrt{2^{2k}}}\sum_{y=0}^{2^k-1}(-1)^{r\cdot y}\sum_{x=0}^{2^k-1}(-1)^{(m_1\oplus m_2\oplus y)\cdot x}|y\rangle\\&=(-1)^{r\cdot y}|m_1\oplus m_2\rangle.\end{aligned}$$

Then, Alice measures it and obtains $m_1\oplus m_2$.

8) Carol announces either m_1 or m_2. Then, Alice guesses m_2 if Carol announced m_1; otherwise she guesses m_1.

Let k be the number of coins. Then, the complexity of these methods mentioned in this section is in $O(k)$ time because each operation of X, H, and $CNOT$ can be executed in $O(1)$ time.

4. Quantum Card Tricks

In this section, we show some quantum card tricks using quantum states. Magicians Alice and Bob guesses the numbers selected by spectators Carol and Davis. Throughout this section, let arithmetic operations be executed to modulus a prime integer N.

First, let (Alice, Carol) and (Bob, Davis) be two pairs. Then, we show tricks such that Alice guesses Davis's number and Bob guesses Carol's number.

Telepathy: First, Alice prepares a card written a number, and puts in a box. The number of this card can be rewritten. Next, Carol multiplies it by m and adds a random r to it, where $m,r\in\mathbf{Z}_N^+$. Finally, Bob prepares the $N-1$ numbered cards. Carol opens the box and obtains a number. By turning over Bob's card corresponding the number, Carol confirms that the reverse side of the card is *m*.

Method of Telepathy

1) Beforehand, Alice and Bob share the following entangled state.

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|x\rangle,$$

where Alice has the first register and Bob has the second register. Alice's register is put in a box.

2) Carol multiplies Alice's register by m, and adds r to it. Then, the state becomes

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|mx+r\rangle|x\rangle.$$

3) Alice and Bob apply QFT to it and the state becomes

$$\frac{1}{\sqrt{N^3}}\sum_{y_1=0}^{N-1}\sum_{y_2=0}^{N-1}\sum_{x=0}^{N-1}e^{\imath 2\pi(mx+r)y_1/N}e^{\imath 2\pi xy_2/N}\,|\,y_1\rangle\,|\,y_2\rangle$$
$$=\frac{1}{\sqrt{N^3}}\sum_{y_1=0}^{N-1}e^{\imath 2\pi ry_1/N}\sum_{y_2=0}^{N-1}\sum_{x=0}^{N-1}e^{\imath 2\pi(my_1+y_2)x/N}\,|\,y_1\rangle\,|\,y_2\rangle$$
$$=\frac{1}{\sqrt{N}}\sum_{my_1+y_2\equiv 0(\mathrm{mod}\,N)}e^{\imath 2\pi ry_1/N}\,|\,y_1\rangle\,|\,y_2\rangle.$$

Then, Bob measures it and obtains y_2 satisfying $my_1+y_2\equiv 0(\mathrm{mod}\,N)$.

4) Bob prepares a set of pairs (m,y_1) satisfying $my_1+y_2\equiv 0(\mathrm{mod}\,N)$. That is, he writes y_1 to the surface of a card and writes m to the reverse side. He makes cards corresponding to all the possible pairs of (m,y_1). Then, he exhibits the set of the cards to Carol.

5) Carol opens the box and knows y_1. Then, she turns over Bob's card written y_1 and confirms that the value of the reverse side is *m*.

Mutual Telepathy: Let Alice and Carol be one pair, and Bob and Davis be another pair. First, Alice prepares a card written a number, and puts in a box. Bob also prepares a card written a number, and puts in another box. Next, Carol multiplies it by m_1, and adds a random r_1 to it. Davis multiplies it by m_2, and adds a random r_2 to it. Here, $m_1,m_2,r_1,r_2\in\mathbf{Z}_N^+$. Finally, Bob prepares the $N-1$ numbered cards. Carol opens the box and obtains a number. By turning over Bob's card corresponding the number, Carol confirms that the reverse side of the card is m_1. In addition, Alice prepares the $N-1$ numbered cards. Davis opens the box and obtains a number. By turning over Alice's card corresponding the number, Davis confirms that the reverse side of the card is m_2.

Method of Mutual Telepathy

1) Beforehand, Alice and Bob share the following entangled state.

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|\,x\rangle\,|\,x\rangle,$$

where Alice has the first register and Bob has the second register. Alice's register is put in a box, and Bob's register is put another box.

2) Carol multiplies Alice's register by m_1 and adds r_1 to it. Davis multiplies Bob's register by m_2 and adds r_2 to it. Then, the state becomes

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|\,m_1x+r_1\rangle\,|\,m_2x+r_2\rangle.$$

In addition, Davis announces m_2 to Bob.

3) Alice and Bob apply QFT to it and the state becomes

$$\frac{1}{\sqrt{N^3}}\sum_{y_1=0}^{N-1}\sum_{y_2=0}^{N-1}e^{\imath 2\pi(r_1y_1+r_2y_2)/N}\sum_{x=0}^{N-1}e^{\imath 2\pi(m_1y_1+m_2y_2)x/N}\,|\,y_1\rangle\,|\,y_2\rangle$$
$$=\frac{1}{\sqrt{N}}\sum_{m_1y_1+m_2y_2\equiv 0(\mathrm{mod}\,N)}e^{\imath 2\pi(r_1y_1+r_2y_2)/N}\,|\,y_1\rangle\,|\,y_2\rangle.$$

Then, Alice and Bob measure it and obtain y_1 and y_2, respectively, satisfying $m_1y_1+m_2y_2\equiv 0(\mathrm{mod}\,N)$.

4) Bob prepares a set of pairs (m_1,y_1) satisfying $m_1y_1+m_2y_2\equiv 0(\mathrm{mod}\,N)$. That is, he writes y_1 to the surface of a card and writes m_1 to the reverse side. He makes cards corresponding to all the possible pairs of (m_1,y_1). Then, he exhibits the set of the cards to Carol.

5) Carol opens the box and knows y_1. Then, she turns over Bob's card written y_1 and confirms that the value of the reverse side is m_1. Note that Alice can also know m_1 here.

6) Alice also prepares a set of pairs (m_2,y_2) satisfying $m_1y_1+m_2y_2\equiv 0(\mathrm{mod}\,N)$, and Davis can find the correct pair (m_2,y_2).

Next, we show a card trick similar to **Flip-Flop2**.

Prediction: First, Alice prepares a card written 0, and puts it in a box. Next, Carol adds $m_1\in\mathbf{Z}_N^+$ to it. Alice executes some operation. Carol multiplies it by m_2 and adds a random r to it, where $m_2,r\in\mathbf{Z}_N^+$. Alice executes some operation, opens the box, and obtains a number. Finally, Alice prepares the *N*–1 numbered cards. By turning over Alice's card corresponding m_1, Carol confirms that the reverse side of the card is m_2.

Method of Prediction

1) Alice prepares a state $|\,0\rangle$, exhibits it to Carol, and puts it in a box.

2) Carol adds m_1 to it and the state becomes $|\,m_1\rangle$.

3) Alice applies QFT to it and the state becomes

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}e^{\imath 2\pi m_1x/N}\,|\,x\rangle.$$

4) Carol multiplies it by m_2 and adds r to it. Then, the state becomes

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}e^{\imath 2\pi m_1x/N}\,|\,m_2x+r\rangle.$$

5) Alice applies QFT to it and the state becomes

$$\frac{1}{\sqrt{N^2}}\sum_{y=0}^{N-1}\sum_{x=0}^{N-1} e^{\imath 2\pi m_1 x/N} e^{\imath 2\pi (m_2 x+r)y/N} \mid y\rangle$$
$$= \frac{1}{\sqrt{N^2}}\sum_{y=0}^{N-1} e^{\imath 2\pi r y/N}\sum_{x=0}^{N-1} e^{\imath 2\pi (m_1+m_2 y)x/N} \mid y\rangle$$
$$= e^{\imath 2\pi r y'/N} \mid y'\rangle,$$

where $m_1 + m_2 y' \equiv 0 (\mathrm{mod}\ N)$. Then, Alice measures it and obtains y'.

6) Alice prepares a set of pairs (m_1, m_2) satisfying $m_1 + m_2 y' \equiv 0 (\mathrm{mod}\ N)$. That is, she writes m_1 to the surface of a card and writes m_2 to the reverse side. she makes cards corresponding to all the possible pairs of (m_1, m_2). Then, she exhibits the set of the cards to Carol.

7) Carol turns over Alice's card written m_1 and confirms that the value of the reverse side is m_2.

Finally, we show a trick such that Alice guesses the number selected by Carol in a situation that Alice prepares a set of cards beforehand.

Mindreading: Beforehand, Alice prepares a set of $N-1$ cards. She writes each $y \in \mathbf{Z}_N^+$ to each card and writes $\sigma(y)$ to the reverse side, where $\sigma(y)$ is a random permutation of y. First, Carol selects $m \in \mathbf{Z}_N^+$ and announces it to Alice. Alice prepares a card, and puts in a box. Next, Carol adds a random $r \in \mathbf{Z}_N^+$ to it. Alice executes some operation. Finally, Carol opens the box, and obtains a number. By turning over Alice's card corresponding to the number, Carol confirms that the reverse side of the card is *m*.

Method of Mindreading

1) Carol selects m and announces it to Alice.

2) Alice prepares a state

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1} e^{\imath 2\pi w x/N} \mid x\rangle,$$

where let $\sigma(w) = m$. This is put in a box.

3) Carol adds a random r to it, and the state becomes

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1} e^{\imath 2\pi w x/N} \mid x+r\rangle,$$

4) Alice applies QFT^{-1} to it and the state becomes

$$\frac{1}{\sqrt{N^2}}\sum_{y=0}^{N-1}\sum_{x=0}^{N-1} e^{\imath 2\pi w x/N} e^{-\imath 2\pi (x+r)y/N} \mid y\rangle$$
$$= \frac{1}{\sqrt{N^2}}\sum_{y=0}^{N-1} e^{-\imath 2\pi r y/N}\sum_{x=0}^{N-1} e^{\imath 2\pi (w-y)x/N} \mid y\rangle$$
$$= e^{-\imath 2\pi r w/N} \mid w\rangle.$$

5) Carol opens the box, obtains *w*, and confirms that the value of the reverse side is *m*.

Let $c(n)$ be the time complexity of arithmetic operations, where n is the size of the input. In addition, let $q(n)$ be the time complexity of QFT. It is know that both $c(n)$ and $q(n)$ are within polynomial of n. Then, the complexity of their methods mentioned in this section is in $O(c(\log N) + q(\log N))$ time.

5. Conclusions

In this paper, we proposed new coin tricks and card tricks based on quantum physics. In these tricks, magicians had the ability of quantum physics, but spectators had only classical one. Therefore, magicians could manipulate coins and cards as quantum states. Moreover, by sharing entangled states, they could transmit spectators' values without communicating between them.

Since our tricks are simple and straightforward ones using quantum states, they are somewhat clumsy. Therefore, it is a future work to construct polished tricks. Moreover, in our tricks, spectators had only classical power. Therefore, it is an interesting problem that we construct quantum tricks when spectators also have quantum power.

REFERENCES

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM Journal on Computing, Vol. 26, pp. 1484–1509, 1997.

[2] D. A. Meyer, "Quantum strategies," Physical Review Letters, Vol. 82, pp. 1052–1055, 1999.

[3] J. Eisert, M. Wilkens, and M. Lewenstein, "Quantum games and quantum strategies," Physical Review Letters, Vol. 83, pp. 3077–3080, 1999.

[4] J. Du, H. Li, X. Xu, M. Shi, J. Wu, X. Zhou, and R. Han, "Experimental realization of quantum games on a quantum computer," Physical Review Letters, Vol. 88, 2002.

[5] J. Du, H. Li, X. Xu, X. Zhou, and R. Han, "Entanglement enhanced multiplayer quantum games," Physical Review Letters, Vol. 302, pp. 229–233, 2002.

[6] J. Eisert and M. Wilkens, "Quantum games," Journal of Modern Optics, Vol. 47, pp. 2543–2556, 2000.

[7] A. Iqbal and A. H. Toor, "Evolutionarily stable strategies in quantum games," Physics Letters A, Vol. 280, pp. 249–256, 2001.

[8] L. Marinatto and T. Weber, "A quantum approach to static games of complete information," Physics Letters A, Vol. 272, pp. 291–303, 2000.

J. Software Engineering & Applications, 2010, 3: 245-254
doi:10.4236/jsea.2010.33030 Published Online March 2010 (http://www.SciRP.org/journal/jsea)

Lightweight Behavior-Based Language for Requirements Modeling

Zhengping Liang[1,2], Guoqing Wu[3], Li Wan[3]

[1]College of Computer and Software Engineering, Shenzhen University, Shenzhen, China; [2]State Key Laboratory of Software Engineering, Wuhan, China; [3]School of Computer, Wuhan University, Wuhan, China.
Email: liangzp@szu.edu.cn

Received December 22nd, 2009; revised January 15th, 2010; accepted January 19th, 2010.

ABSTRACT

Whether or not a software system satisfies the anticipated user requirements is ultimately determined by the behaviors of the software. So it is necessary and valuable to research requirements modeling language and technique from the perspective of behavior. This paper presents a lightweight behavior based requirements modeling language BDL with formal syntax and semantics, and a general-purpose requirements description model BRM synthesizing the concepts of viewpoint and scenario. BRM is good for modeling large and complex system due to its structure is very clear. In addition, the modeling process is demonstrated through the case study On-Line Campus Management System. By lightweight formal style, BDL & BRM can effectively bridge the gap between practicability and rigorousness of formal requirements modeling language and technique.

***Keywords*: *Behavior Description Language* (*BDL*), *Scenario*, *Viewpoint*, *Behavior Requirements Model* (*BRM*)**

1. Introduction

Software requirements modeling is an important phase of software development process. To obtain high quality requirements model, an effective and well-defined requirements modeling language and technique, which both has formal semantic and can be easily understood and used by all kinds of stakeholders, is needed.

The existing requirements modeling languages and techniques can be roughly divided into two categories. One is the semi-formal style based on graph symbol, the most famous representative of which is UML [1]. The other is the formal style based on mathematics symbol, such as Automata [2], Z [3], E-LOTOS [4], Petri net [5], Pi-calculus [6], etc. The former has the advantage of strong intuition, of being easy to be understood and used, but it usually lacks rigorous semantics and easily leads to an inconsistent and incomplete requirements model. On the contrary, the latter has rigorous semantics basis and is convenient to deduce and verify some properties, but it has poor practicability, and requires the user and analyzer with advanced skills.

How to deal with the gap between practicability and rigorousness of formal requirements modeling language and technique is a big challenge [7]. Some researches suggest to designating formal semantic for semi-formal language [8], and others believe the combination of graph symbol and formal language are more positiveness [9]. Although all of those approaches have some effect to bridge the gap, there are still inconvenient to put them into practice. At the same time, whether or not a software system satisfies the anticipated user requirements is ultimately determined by the behaviors of the software. That is to say, the requirements modeling language and technique need to support the description and validation of behavior. So it is necessary and valuable to research software requirements modeling language and technique both has practicability and rigorousness from the perspective of software behavior.

Due to lightweight formal style can help to bridge the gap between practicability and rigorousness [10], we established a lightweight formal language BDL (Behavior Description Language) to modeling user's requirements, which is based on the identifiable behaviors of software system. What should be emphasized is that the behaviors not only include the observable behaviors from the system external interface but also consist of the behaviors resided in the internal of the system. In addition, in order to support the requirements modeling of large and complex software, a general-purpose requirements description model BRM (Behavior Requirements Model) is proposed, which partly synthesizes some ideas of viewpoint-oriented requirements engineering [11] and scenario-oriented requirements engineering [12].

The structure of this paper is organized as follows: Section 2 introduces the formal syntax of BDL and its structural operational semantics. Section 3 introduces the requirements description model BRM and Section 4 demonstrates the modeling process through the case study On-line Campus Management System. Finally, the related works are discussed in Section 5 and the conclusions and future works are discussed in Section 6.

2. Behavior Based Requirements Modeling Language

A behavior is a certain interaction among two or more entities. For easy discussion, this paper presumes a behavior is an interaction only between two entities. We define a software behavior as a process during which a subject implements an operation, service, or action to an object. The subject and the object which may be physical or logistic, can be a person, a software or hardware component of system, or certain element of environment.

The structure of each behavior consists of a subject, an object, some properties, some inputs, some outputs, and an operation, service, or action. If a behavior can't be divided into two or more sub-behaviors, it is an atomic behavior. An atomic behavior is a simple behavior. Two or more simple behaviors form a composite behavior. In addition, according with the interact mode of software behaviors, the combine pattern of simple behaviors can be divided into five categories: sequence, certainty choice, uncertainty choice, parallel and shielding.

Based on the above consideration about software behavior, the followings are the syntax and structural operational semantics of behavior based requirements modeling language BDL.

2.1 Syntax of BDL

Suppose $ABehID, ABehID_i (i \in N)$ are atomic behavior identifier, $BehID, BehID_i (i \in N)$ are behavior identifier.

2.1.1 Atomic Behavior Expression

$ABehID : f(sub, obj[\&\ obj's\ additional\ remarks])$
[When *prepositive conditions*]
$[\text{INFrom}(ID)(u_1,...,u_n)]^*$
$[\text{OUTTo}(ID)(v_1,...,v_m)]^*$

where,

f is an operation or an action;

sub and *obj* are the behavior's subject and object respectively;

When clause denotes the *prepositive conditions* according to which the behavior can execute;

INFrom and OUTTo clause denote the behavior's input data and output data respectively;

ID denotes a certain atomic behavior identifier, a external entity or a viewpoint identifier related to INFrom or OUTTo ;

$u_i (i \in \{1...n\})$ and $v_i (i \in \{1...m\})$ are described with the format of *dataname* or *dataname* = *value* ;

The superscript $*$ denotes there are 0 or multiple items that belong to the same category. Besides, there are two kinds of special atomic behavior:

1) Null action: $ABehID : Idle$;

2) End action of composite behavior:

a) $ABehID : Return(ABehID_i)$

//jump to execute atomic behavior $ABehID_i$;

b) $ABehID : Return()$

//end of execute composite bahavior .

2.1.2 Simple Behavior

$\vdash ABehID$ //atomic behavior act as simple behavior

2.1.3 Composite Behavior

1) Sequence behavior:

a) $$\frac{\vdash ABehID \,\&\vdash BehID}{\vdash ABehID; BehID}$$

b) $$\frac{\vdash BehID \,\&\vdash ABehID}{\vdash BehID; ABehID}$$

c) $$\frac{\vdash BehID_1 \,\&\vdash BehID_2 \,\&...\&\vdash BehID_n}{\vdash BehID_1; BehID_2; ...; BehID_n}$$

2) Certainty choice behavior:

$$\frac{\vdash BehID_1 \,\&\vdash BehID_2 \,\&\, b \text{ is a boolean expression}}{\vdash If\ b\ Then\ BehID_1\ Else\ BehID_2\ Fi}$$

3) Uncertainty choice behavior:

$$\frac{\vdash BehID_1 \,\&\vdash BehID_2 \,\&...\&\vdash BehID_n}{\vdash BehID_1 + BehID_2 + ... + BehID_n}$$

4) Parallel behavior:

$$\frac{\vdash BehID_1 \,\&\vdash BehID_2 \,\&...\&\vdash BehID_n}{\vdash BehID_1 \,||\, BehID_2 \,||...||\, BehID_n}$$

5) Shielding behavior:

a) $$\frac{\vdash BehID \,\&\vdash ABehID}{\vdash BehID / ABehID}$$ //shielding atomic behavior

b) $$\frac{\vdash BehID \,\&\vdash BehID_1}{\vdash BehID / BehID_1}$$ //shielding composite behavior

2.2 Structural Operational Semantics of BDL

Definition1: Suppose B is a behavior expression, σ is a state of system, then $< B, \sigma >$ is a configuration. $< B, \sigma >$ denotes the current state is σ and the be-

havior expression to be executed is *B*. $<\sigma>$ is also a configuration, which denotes the current state is σ and there are no behavior expression need to be executed.

Definition2: Suppose *b* is a Boolean expression, σ is a state, $eval<b,\sigma>$ denotes the Boolean value of *b* at σ.

Suppose $\alpha, \alpha_i (i \in N)$ are atomic behavior, B, B_i $(i \in N)$ are behavior expression. The structural operational semantics of BDL can be defined in this way:

1) Semantic of atomic behavior expression:

$$<\alpha,\sigma>\rightarrow<\sigma'>$$

2) Semantic of Null action:

$$<Idle,\sigma>\rightarrow<\sigma>$$

3) Semantic of End action of composite behavior:
Suppose α is the first atomic behavior of *B*.

$$<Return(\alpha),\sigma>\rightarrow<B,\sigma>$$

$$<Return(),\sigma>\rightarrow<\sigma>$$

4) Semantic of sequence behavior:

$$\frac{<B_1,\sigma>\rightarrow<\sigma'>}{<B_1;B_2,\sigma>\rightarrow<B_2,\sigma'>}$$

$$\frac{<B_1,\sigma>\rightarrow<B_1',\sigma'>}{<B_1;B_2,\sigma>\rightarrow<B_1';B_2,\sigma'>}$$

5) Semantic of certainty choice behavior:

$$\frac{eval<b,\sigma>=TRUE}{<If\ b\ Then\ B_1\ Else\ B_2\ Fi,\sigma>\rightarrow<B_1,\sigma>}$$

$$\frac{eval<b,\sigma>=FALSE}{<If\ b\ Then\ B_1\ Else\ B_2\ Fi,\sigma>\rightarrow<B_2,\sigma>}$$

6) Semantic of uncertainty choice behavior:

$$\frac{<B_1,\sigma>\rightarrow<B_1',\sigma'>}{<B_1+B_2,\sigma>\rightarrow<B_1',\sigma'>}$$

$$\frac{<B_2,\sigma>\rightarrow<B_2',\sigma'>}{<B_1+B_2,\sigma>\rightarrow<B_2',\sigma'>}$$

7) Semantic of parallel behavior:

$$\frac{<B_1,\sigma>\rightarrow<B_1',\sigma'>}{<B_1\,||\,B_2,\sigma>\rightarrow<B_1'\,||\,B_2,\sigma'>}$$

$$\frac{<B_2,\sigma>\rightarrow<B_2',\sigma'>}{<B_1\,||\,B_2,\sigma>\rightarrow<B_1\,||\,B_2',\sigma'>}$$

8) Semantic of shielding behavior:
Suppose $B=\alpha';B'$.

$$\frac{<B,\sigma>\rightarrow<B',\sigma'>}{<B/\alpha,\sigma>\rightarrow<B'/\alpha,\sigma'>}\ (\alpha'\neq\alpha)$$

//shielding atomic behavior

Suppose $B=B_i;B'$.

$$\frac{<B,\sigma>\rightarrow<B',\sigma'>}{<B/B_1,\sigma>\rightarrow<B'/B_1,\sigma'>}\ (B_i\neq B_1)$$

//shielding composite bahavior

3. Behavior Based Requirements Description Model

As to small and simple software system, BDL can be used to describe its requirements model directly due to BDL's syntax is also simple and small. But it is hard to describe requirements model of large and complex software system using BDL directly because on the one side the software scale and structure may be very complicated, and on the other side many kinds of stakeholders who reside in different time zone and space, may be involved.

To deal with large and complex problems, people often employ the strategy of divide-and-rule. Based on this method, we propose a general-purpose requirements description model BRM, which synthesizes the concepts of viewpoint and scenario. The model process of BRM consists of five steps: first, to identify the scope of the whole problem domain of the software system, next, to divide the problem domain into some interrelated sub-domains. After that, to list all potential viewpoints and their sequence or overlap relationships of each sub-domain based on the viewpoint identifying methods of viewpoint-oriented requirements engineering [11]. Later on, to look for different scenarios and their sequence or overlap relationships of each viewpoint. Finally, to adopt the scenario describing way of scenario-oriented requirements engineering [12] to establish each scenario model using BDL.

BRM is composed of three kinds of model. One is the scenario behavior model, another is viewpoint behavior model, and the last is system behavior model. The followings are the formal definition of them.

Definition3 (Scenario behavior model): A scenario's behavior model is a 6-tuple:

$$M_s=(B, ;, If, +, ||, /)$$

where,

B is the set of behaviors within the scenario, and each behavior in B has a corresponding behavior expression;

;, If, +, ||, / respectively denotes the relationship of sequence, certainty choice, uncertainty choice, parallel and shielding between behaviors.

The syntax structure of scenario behavior model is defined as **Figure 1**, where, *ABehID* : *Atomic behavior* is a certain atomic behavior expression; *BehaviorOperator* is one of the relation symbol between behaviors, that is ;, If, +, ||, / .

Definition4 (Viewpoint behavior model): A viewpoint's behavior model is a 4-tuple:

$$M_v=(S, \circ, \Diamond, \perp)$$

where,

S is the set of scenarios within the viewpoint, and each scenario in S has a corresponding scenario behavior model;

$\circ$ is a 2-tuple operator, which denotes two scenarios have the sequence relationship in terms of execution;

$\Diamond$ is also a 2-tuple operator, which denotes two scenarios have overlaps in content, that is, they have common behaviors;

$\perp$ denotes two or more scenarios are independent of each other in execution order and in content.

These relation operators can be use to assisting analyze and check requirements model's properties from the aspect of syntax and semantic at the phase of requirements analysis.

The syntax structure of viewpoint behavior model is defined as **Figure 2**, where, *ScenarioOperator* is the scenario's relation symbol $\circ$ or $\Diamond$.

Definition5 (System behavior model): A software system's behavior model is a 4-tuple:

$M=(V, \circ, \Diamond, \perp)$

where,

V is the set of viewpoints related to the system, and each viewpoint in V has a corresponding viewpoint behavior model;

$\circ$ is a 2-tuple operator, which denotes two viewpoints have the sequence relationship in terms of execution;

$\Diamond$ is also a 2-tuple operator, which denotes two viewpoints have overlaps in domain, that is, the sub-domains where they belong to have common elements;

$\perp$ denotes two or more viewpoints are independent of each other in execution order and in domain.

These relation operators can also be use to assisting analyze and check requirements model's properties from the aspect of syntax and semantic at the phase of requirements analysis.

The syntax structure of system behavior model is defined as **Figure 3**, where, *ViewpointOperator* is the viewpoint's relation symbol $\circ$ or $\Diamond$.

Obviously, because the structure and relationship of above models are very clear, people can smoothly transfer the user requirements expressed by natural languages to formal requirements model expressed by BDL based on BRM. Hence, BDL & BRM make a moderate balance between practicability and rigorousness.

```
ScenarioID
SCBEGIN
  [ABEH:    //list of atomic behaviors, it also can be given in BEH directly
    ABehID : atomic behavior
    [, ABehID : atomic behavior]*;;]
  BEH :     //list of behaviors
    BehID = AbehID | atomic behavior | BehID |
      (AbehID | atomic behavior | BehID) BehaviorOperator (AbehID | atomic behavior | BehID)
      [BehaviorOperator (AbehID | atomic behavior | BehID)]*  //at lease one behavior in a scenario
    [, BehID = AbehID | atomic behavior | BehID |
      (AbehID | atomic behavior | BehID) BehaviorOperator (AbehID | atomic behavior | BehID)
      [BehaviorOperator (AbehID | atomic behavior | BehID)]*]*;;
    SBehID =    //scenario behavior expression
      (BehID | BehID BehaviorOperator BehID [BehaviorOperator BehID]*);;
SCEND
```

Figure 1. Syntax of scenario behavior model

```
ViewpointID
VPBEGIN
  [data storage pool ID];; //used to store data input from other viewpoint and data
                           //shared by different scenarios within the viewpoint
  ScenarioID               //at lease one scenario in a viewpoint
  [, ScenarioID]*;;
  SC _ Relationship =      //set of relationship between scenarios
    {[< ScenarioID ScenarioOperator ScenarioID >
      [, < ScenarioID ScenarioOperator ScenarioID >]*]};;
VPEND
```

Figure 2. Syntax of viewpoint behavior model

4. Case Study

On-line Campus Management System (*OCMS*) consists of several subsystems related each other, which used for the daily management of education administrative unit and schools. Its user requirements have modeled using BDL & BRM. In this section, we demonstrate a partial of function requirements model of *OCMS*.

The following is part of the functions of Student Information Management Subsystem:

1) *Student needs to scan his or her IC card at the door-control reader when he or she enters or leaves schoolyard, at the same time, a correlative short message will be automatically sent to the student's parents' mobile phone*;

2) *Teachers can process students' all kinds of information and send student's information to his or her parents by the way of short message and E-mail*;

3) *Administrator distributes IC cards and manages its authorization. Besides, Administrator sets the students attendance rules and the system automatically creates the students attendance reports*;

4) *Parents can query his or her child's all kind of information at school by the way of short message, automatic voice and webpage.*

The logic structure of above functions as **Figure 4**.

Although the above function requirements look very simple, there are many complicated and redundant details. For example, how long and how to does the attendance report is created, how to manage the input, modification, processing, storage, transmission, respondence, etc. of all kinds of students' information among different domain elements. Due to space limitations, we directly give the analysis result of above requirements and only demonstrate a partial of requirements model using BDL & BRM.

The problem domain boundary of above user requirements is clear. The followings are the five sub-domains of it:

Sub-domain 1: student, IC card, IC reader, mainframe, door and swivel of door;

Sub-domain 2: administrator, attendance rules, terminal, mainframe, IC card, IC reader;

Sub-domain 3: teacher, all kinds of student's information at school, terminal, mainframe;

Sub-domain 4: parents, mobile phone, telephone, terminal, mobile phone networks, telephone networks, Internet, all kinds of student's information at school;

Sub-domain 5: mainframe, IC reader, terminal, mobile phone networks, telephone networks, Internet, attendance report, all kinds of student's information at school, list of IC card information.

```
SystemID
SYBEGIN
    ViewpointID        //at lease one viewpoint in a system
    [,ViewpointID]*;;
    VP _ Relationship =      //set of relationship between viewpoints
        {[<ViewpointID ViewpointOperator ViewpointID>
          [,<ViewpointID ViewpointOperator ViewpointID>]*]};;
SYEND
```

Figure 3. Syntax of system behavior model

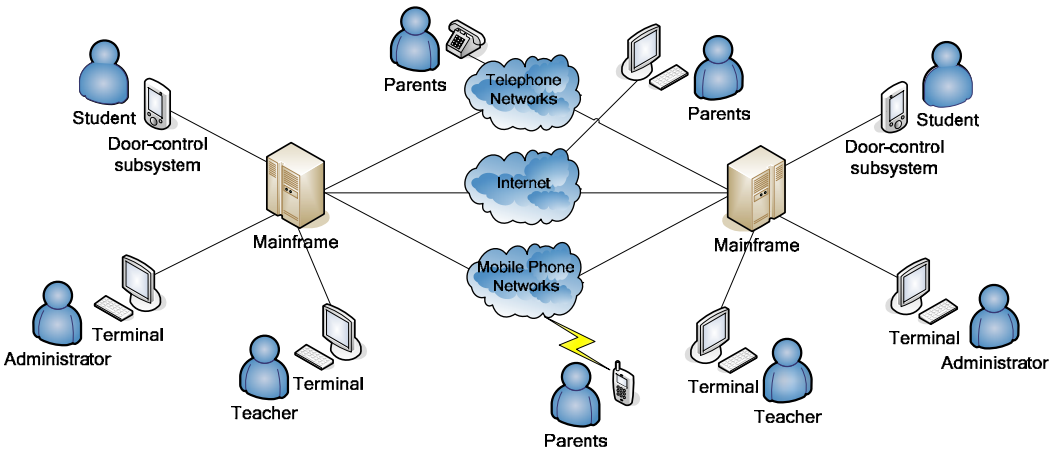


Figure 4. Logic structure of student information management subsystem

Table 1. Relationships of viewpoints belong to Sub-domain 5

Relationships	VP_ICInfo_Manage	VP_AttenRep_Create	VP_Query_Respond	VP_Info_Send	VP_Info_Edit
VP_ICInfo_Manage	\	◊	⊥	⊥	⊥
VP_AttenRep_Create	\	\	∘	⊥	⊥
VP_Query_Respond	\	\	\	◊,∘	◊,∘
VP_Info_Send	\	\	\	\	◊,∘
VP_Info_Edit	\	\	\	\	\

Notes: "\" denotes null.

(1) ICreaderDisp1: display(ICreader, screen)
OUTTo(screen)(*Prompt*="Please scanning card!")
(2) DoorConWait:idle() //waiting user to scan card
(3) ScanC:scancard(person, IC)
(4) ReadC:read(ICreader, IC)
OUTTo(SendCInfo)(*ICNo*, *username*),
(5) SendCInfo:send(ICreader, Mainframe) //send the username to viwepoint VP_ICInfo_Manage
OUTTo(VP_ICInfo_Manage)(*ICNo*, *username*)
(6) RecVerInfo:receive(ICreader, Mainframe) //receive the verification result of IC
INFrom(datapool)(*result*) //the result is store in the viewpoint's data pool
(7) ICReaderDisp2:display(ICreader, screen)
OUTTo(screen)(*username*, *Prompt*="Coming Please!")
(8) AllowOpen:allow(ICreader, swivel)
OUTTo(swivel)(*signal*)
(9) OpenDoor:open(swivel, door) //the action of open the door
(10) CloseDoor:close(swivel, door)

Figure 5. Atomic behavior expressions of the scenario with the right to open the door

These sub-domains related each other through common elements. For example, Sub-domain 1 and Sub-domain 5 has the common element IC reader, which hints some viewpoints of them may have the relationship "◊" defined in Definition 5.

As to Sub-domain 5, we can identify five viewpoints: VP_ICInfo_Manage, VP_AttenRep_Create, VP_Query _Respond, VP_Info_Send, VP_Info_Edit. The relationships of them as **Table 1** using the shape of strictly upper triangular matrix.

As to Sub-domain 1, there is only one viewpoint VP_ScanCard, which have the following relationships with the viewpoints belong to Sub-domain 5:

<VP_ScanCard ◊ VP_ICInfo_Manage>, <VP_Scan Card ∘ VP_Info_Send>, etc.

Now, we give a demonstration of VP_ScanCard's modeling process and its behavior model. The followings are the detailed user requirements of this viewpoint:

When a student wants to enter or leave school, she or he needs to scan her or his IC card at the IC reader firstly. If the IC-holder is authorized to enter or leave the school, the door-control system will display the IC-holder's name on the IC reader's screen and open the door. Otherwise, a warning sound will be played in the IC reader's speaker, and the reason why the person is not permitted to enter or leave will be displayed on the screen.

In this viewpoint, there are two scenarios: one is the IC-holder has the right to enter or leave school SC_ValidScanCard, the other is the opposite SC_InvalidScanCard.

First, we list all atomic behavior expressions belong to SC_ValidScanCard according to above requirements as **Figure 5**.

Then, the scenario behavior model of SC_ValidScanCard can be established as **Figure 6** according to the interrelated relationship of above atomic behavior expressions and Definition 3.

Next, the scenario behavior model of SC_Invalid ScanCard as **Figure 7** can be established similarly.

After that, due to SC_ValidScanCard and SC_ InvalidScanCard have the common elements in domain, the viewpoint behavior model of VP_ScanCard is established as **Figure 8**.

Here, the behavior model of VP_ScanCard is established successfully. Behavior model of other user requirements can be established similarly.


```
SC _ ValidScanCard
SCBEGIN
  ABEH:
    ICreaderDisp1: display(ICreader, screen)
                    OUTTo(screen)(Prompt="Please scanning card!"),
    DoorConWait:idle(),
    ScanC:scancard(person, IC),
    ReadC:read(ICreader, IC)
          OUTTo(SendCInfo)(ICNo, username),
    SendCInfo:send(ICreader, Mainframe)
              OUTTo(VP_ICInfo_Manage)(ICNo, username),
    RecVerInfo:receive(ICreader, Mainframe)
               INFrom(datapool)(result),
    ICReaderDisp2:display(ICreader, screen)
                  OUTTo(screen)(username, Prompt="Coming Please!"),
    AllowOpen:allow(ICreader, swivel)
              OUTTo(swivel)(signal),
    OpenDoor:open(swivel, door),
    CloseDoor:close(swivel, door);;
  BEH:
    BehValidUResp=ICReaderDisp2 AllowOpen,
    BehValidU=
      ICreaderDisp1;
      DoorConWait;
      ScanC;
      ReadC;
      SendCInfo;
      RecVerInfo;
      BehValidUResp;   //if the IC is valid, open the door
      OpenDoor;
      CloseDoor;
      Return(ICreaderDisp1);;
    SBehID=BehValidU;;
SCEND
```

Figure 6. Scenario behavior model with the right to open the door

5. Related Works

The semi-formal and formal requirements modeling language and technique both have achieved prominent outcomes in the past twenty years. As to the behavior based requirements modeling, the importance and validity of it has also recognized by many researchers from academia and industry [13-20].

Ayaz *et al*. propose a behavioral specification language for complex systems—Viewcharts, which extends Statecharts to include behavioral views and their compositions [13]. And they define the syntax of viewcharts as attributed graphs and describe dynamic semantics of viewcharts by object mapping automata [14]. Viewcharts notation allows views to be specified independent of each other, which is similar to BDL. A difference between this work and ours is that Viewcharts does not consider behaviors reside in the internal of system, but only observable behaviors from the external system.

Assem proposes an event-oriented requirements definition approach named Behavioral Pattern Analysis Approach (BPA) [15]. In BPA, Event is the primary object of the world model. And it use the so-called BPA Behavioral Pattern, which is the template that one uses to model and describe an event, takes the place of the use case in the UML. BPA is a more effective alternative to use cases in modeling and understanding the function requirements. However, BPA is special for real-time systems, multi-agent systems and safety-critical systems. Besides, it lacks clear links among Behavioral Patterns and can't be used for modeling complex system and is not convenient for requirements verification. On the contrary, our approach definitely labels the relationships of scenarios, viewpoints, and sub-domains, can effectively


```
SC _ InvalidScanCard
SCBEGIN
  ABEH:
    ICreaderDisp1: display(ICreader, screen)
                    OUTTo(screen)(Prompt="Please scanning card"),
    DoorConWait:idle(),          //waiting user to scan card
    ScanC:scancard(person, IC),
    ReadC:read(ICreader, IC)
          OUTTo(SendCInfo)(ICNo, username),
    SendCInfo:send(ICreader, Mainframe)
              OUTTo(VP_ICInfo_Manage)(ICNo, username),
    //receive the verification result of IC, and the result is store in the viewpoint's data pool
    RecVerInfo:receive(ICreader, Mainframe)
                INFrom(datapool)(result),
    PlayWarnSound:play(ICreader,speaker)
                  OUTTo(speaker)(soundfile),
    ICReaderDisp2:display(ICreader, screen)
                  OUTTo(screen)(Prompt="Overdue IC card!"),
    ICReaderDisp3:display(ICreader,screen)
                  OUTTo(screen)(Prompt="The IC card has reported be lost!"),
    ICReaderDisp4:display(ICreader,screen)
                  OUTTo(screen)(Prompt="Invalid user, unknown reason!");;
  BEH:
    BehInvalidUResp=
      PlayWarnSound□
      If result="overdue"
      Then  ICReaderDisp2
      Else  If result="lost"
            Then  ICReaderDisp3
            Else  ICReaderDisp4
            Fi
       Fi,
    BehInvalidU=
      ICreaderDisp1;
      DoorConWait;
      ScanC;
      ReadC;
      SendCInfo;
      RecVerInfo;
      BehInvalidUResp;    //if the IC is invalid, don't open the door
      Return(ICreaderDisp1);;
    SBehID=BehInvalidU;;
SCEND
```

Figure 7. Scenario behavior model without the right to open the door

```
VP _ ScanCard
VPBEGIN
  datapool;;     //the data pool of this viewpoint
  SC_ValidScanCard,
  SC_InvalidScanCard;;
  SC_Relationship={<SC_ValidScanCard ◊ SC_InvalidScanCard>};;
VPEND
```

Figure 8. Viewpoint behavior model of the scanning card

support the modeling and verification of large and complex system.

Khairuddin *et al*. propose a requirements notation RNSMA and a behavioral approach to specify interactive multimedia applications [16]. RNSMA is based on Petri Net, but its semantics are extended to support reactive systems. In RNSMA, transitions due to events are subdivided into automatic, user and clock. The transitions due to tasks to be done are subdivided into animate, image, sound, text and video. RNSMA uses an extremely simple syntax, which can be read even by novices as a form of pseudo-code. Compared with RNSMA, our work can support general-purpose requirements modeling, not special for stand-alone multimedia applications.

UML is a general-purpose and most famous modeling language for software engineering, which is standardized by OMG [1]. Requirements modeling manner in UML consists of the use case diagram, sequence diagram, state diagram and activity diagram. UML provides standard notation for modeling software analysis and design. But a common and fair criticism of UML is that it is gratuitously large and complex, imprecise semantics, and a dysfunctional diagram interoperability standard (XMI). As another OMG standard, SysML acts as a general-purpose modeling language for systems engineering applications [17]. SysML is based on UML, and it reduces UML's size and software bias while extending its semantic to model requirements and parametric constraints. These capabilities are essential to support requirements engineering and performance analysis.

Besides, there are some researches based on UML and SysML. Luigi *et al*. propose combining problem frames and UML to describe software requirements in order to improve the linguistic support for problem frames and the UML development practice by introducing the problem frames approach [18]. Pietro *et al*. propose the integration of SysML and problem frames by presenting how a set of well known problem frames can be represented by means of SysML [19]. Atle *et al*. propose to extend UML sequence diagrams to model trust-dependent behavior with the aim to support risk analysis [20]. All of these researches are good for the enhancement of behavior modeling.

In addition, there are many kinds of formal languages and techniques for requirements modeling, especially for behavior requirements. Most of them are based on state or event. Some are standardized by different international organization, such as Z [3], E-LOTOS [4]. Others may be very famous in industry, such as B [21], VDM [22]. Although formal languages and techniques have many advantages, it is difficult to put into practices totally. On the contrary, our approach can be easily used to transform natural language requirements to formal requirements model because the syntax of BDL and the structure of BRM are very simple and clear.

6. Conclusions and Future Work

Software requirements modeling from the perspective of behavior can not only supports the description and modeling of function requirements but also supports the analysis and deduction of non-function requirements. As a lightweight formal requirements description language and model, BDL & BRM can help to smoothly transfer the user requirements expressed by natural languages to formal requirements model expressed by BDL. And the formal model BRM is also good for subsequent requirements verification and validation. Hence, BDL & BRM can effectively bridge the gap between practicability and rigorousness of formal requirements modeling language and technique. Several completed case studies also testified this kind of feature of BDL & BRM.

Currently, we have realized the prototype requirements modeling tool and experimented some case studies. Future works will mainly focus on to define all kinds of requirements properties based on BDL&BRM, and to design and implement corresponding automatic analyzing and deducing methods.

7. Acknowledgements

This work is supported by the National High Technology Research and Development Program of China under contract 2007AA01Z185, the Open Research Foundation of Chinese State Key Laboratory of Software Engineering under grant SKLSE20080702, and the SZU R/D Fund 200747.

REFERENCES

[1] Object Management Group, "OMG Unified Modeling Language (OMG UML) Version 2.0," 2005.

[2] J. E. Hopcroft, R. Motwani, and J. D. Ullman, "Introduction to automata theory, languages, and computation," 2nd Edition, Pearson Education, 2000.

[3] "Information technology—Z formal specification notation—Syntax, type system and semantics," ISO/IEC 13568, 2002.

[4] "Information processing systems—Open systems interconnection—Enhancements to LOTOS—A formal description technique based on the temporal ordering of observational behavior," ISO/IEC 15437, 2001.

[5] J. L. Peterson, "Petri net theory and the modeling of systems," Prentice Hall, 1981.

[6] R. Milner, "Communicating and mobile systems: The Pi-calculus," Cambridge University Press, 1999.

[7] J. P. Bowen and M. G. Hinchey, "Ten commandments of formal methods... ten years later," IEEE Computer, Vol. 39, No. 1, pp. 40–48, January 2006.

[8] J. Kong, K. Zhang, J. Dong, and D. Xu, "Specifying behavioral semantics of UML diagrams through graph transformations," Journal of Systems and Software, Vol. 82, No. 2, pp. 292–306, April 2009.

[9] C. Attiogbe, P. Poizat, and G. Salaun, "A formal and tool-equipped approach for the integration of state diagrams and formal datatypes," IEEE Transactions on Software Engineering, Vol. 33, No. 3, pp. 157–170, March 2007.

[10] M. Hinchey, M. Jackson, and P. Cousot, J. P. Bowen, and T. Margeria, "Software engineering and formal methods," Communications of the ACM, Vol. 51, No. 9, pp. 54–59, September 2008.

[11] G. Kotonya and I. Sommerville, "Requirements engineering with viewpoints," Software Engineering Journal, Vol. 11, No. 1, pp. 5–18, January 1996.

[12] A. Sutcliffe, "Scenario-based requirements engineering," in Proceedings of the 11th IEEE International Conference on Requirements Engineering, Monterey, California, pp. 320–329, September 2003.

[13] A. Isazadeh, D. A. Lamb, and G. H. MacEwen, "Viewcharts: A behavioral specification language for complex systems," Proceedings of the 4th International Workshop on Parallel and Distributed Real-Time Systems, Honolulu, Hawaii, pp. 208–215, April 1996.

[14] A. Isazadeh and J. Karimpour, "Viewcharts: Syntax and semantic," Informatica, Vol. 19, No. 3, pp. 345–362, March 2008.

[15] A. E. Ansary, "Requirements definition of real-time system using the Behavioral Patterns Analysis (BPA) approach: The elevator control system," Proceedings of the Second International Conference on Software and Data Technologies, Barcelona, pp. 371–377, July 2007.

[16] K. Hashim and J. Yousoff, "A behavioral requirements specification approach for interactive multimedia applications," Proceedings of the 19th Australian Conference on Software Engineering, Perth, West Australia, pp. 696–699, March 2008.

[17] "OMG Systems Modeling Language (OMG SysML) Version 1.1," Object Management Group, 2008.

[18] L. Lavazza and V. D. Bianco, "Combining problem frames and UML in the description of software requirements," In: B. Luciano, H. Reiko, Ed., Lecture Notes in Computer Science, Vol. 3922, pp. 199–213, 2006.

[19] P. Colombo, V. del Bianco, and L. Lavazza, "Towards the integration of sysml and problem frames," Proceedings of the 3rd International Workshop on Applications and Advances of Problem Frames, Leipzig, pp. 1–8, May 2008.

[20] A. Refsdal and K. Stolen, "Extending UML sequence diagrams to model trust-dependent behavior with the aim to support risk analysis," Science of Computer Programming, Vol. 74, No. 1–2, pp. 34–42, January 2008.

[21] S. Schenider, "The B-method: An introduction," Palgrave, 2001.

[22] C. B. Jones, "Systematic software development using VDM," Prentice Hall, 1990.

J. Software Engineering & Applications, 2010, 3: 255-267
doi:10.4236/jsea.2010.33031 Published Online March 2010 (http://www.SciRP.org/journal/jsea)

Information Content Inclusion Relation and its Use in Database Queries

Junkang Feng[1], Douglas Salt[2]

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

Received October 31st, 2009; revised November 19th, 2009; accepted November 25th, 2009.

ABSTRACT

A database stores data in order to provide the user with information. However, how a database may achieve this is not always clear. The main reason for this seems that we, who are in the database community, have not fully understood and therefore clearly defined the notion of "the information that data in a database carry", in other words, "the information content of data". As a result, databases' capability is limited in terms of answering queries, especially, when users explore information beyond the scope of data stored in a database, the database normally cannot provide it. The underlying reason of the problem is that queries are answered based on a direct match between a query and data (up to aggregations of the data). We observe that this is because the information that data carry is seen as exactly the data per se. To tackle this problem, we propose the notion of information content inclusion relation, and show that it formulates the intuitive notion of the "information content of data" and then show how this notion may be used for the derivation of information from data in a database.

Keywords*: *Information Content, Databases, Knowledge Discovery from Databases, Semantic Theory of Information

1. Introduction

When we query a database, it is said that we are retrieving information from it. This is taken for granted. But, how this happens is not always fully understood. As a result, when a user queries a database [1], the query can only be answered through a "direct match" between the selection criteria within a query and data (up to aggregations of the data) [2]. In a case of querying a database beyond this, the system is unlikely to answer the query. A conventional query is, in essence, concerned with only the *propositional content* of data [3]. We believe that data carries information [4–6]. A piece of data may carry information about another, and moreover it may carry information about a real world situation [7,8]. Therefore, if we can define and formulate the notion of "the information content of data", not only may we obtain insight about the essence of conventional queries, but also we may derive more information beyond "direct match".

However, it would appear that the notion of "information content of data" is elusive. It has been taken as the instance of a database and the information capacity of a data schema as the collection of instances of the schema [9–11]. Another view on the topic of the relationship between information and data is that if it is truthful, meaningful data is semantic information [12]. We argue that such views miss two fundamental points. One is a convincing conception of "information content of data". To equate data with information overlooks the fact that data in a database is merely raw material for bearing and conveying information. Information must be veridical [7], that is, it must relate to a contingent truth [12], while for data there is no such requirement. The other is a framework for approaching the information content of data whereby to reveal information.

That is to say, we define the following research question that we tackle in this paper: how the "information content" of data in a database may be defined with mathematical rigor, and how this notion after have been defined may help retrieve information through reasoning that cannot otherwise be possible through conventional queries.

To answer this research question, we purpose to look at the relationships between the information content of data, database structure and domain knowledge, which may be captured as business rules. These include how tacit domain knowledge may be explicitly expressed and used.

In this paper, we present a novel framework for ap-

proaching the information content of data in a database, which is centered on the notion of *information content inclusion relation*. It helps us understand how a database does its job, *i.e.*, providing information, and helps a database system improve its capability of providing information through inference. The latter is achieved by introducing a variety of information sources such as domain knowledge. With the help of external information sources, queries that deal with a wider range of information than the propositional content of data within a database may be answered. The underlying thought of the framework is based on a concept of *information content* of a signal. Dretske [4] firstly introduced the concept. Then Xu *et al.* [13] extended Dretske's idea and gave a more detailed definition of the information content of a state of affairs. Our thoughts are based on the latter definition.

The next section gives a number of foundational concepts. Then the framework and a prototype of implementation are presented in the third section. The last section concludes the paper.

2. Foundational Concepts

A number of concepts are defined in this section and they are foundational for defining the notion of "information content inclusion relation".

2.1 Information Content

Fred Dretske [4] gave the definition of information content as follow:

"A state of affairs contains information about X to just that extent to which a suitably placed observer could learn something about X by consulting it."

Then he formalized the above as

"Information Content: A signal r carries the information that s is F = The conditional probability of *s*'s being F, given r (and k), is 1(but, given k alone, less than 1)."

Note that k stands for prior knowledge about information source s.

Here is an example: That John is awarded a grade "A" for his Programming course contains the information that he has scored 70% or above for that course.

Dretske's above definition needs to be extended, however, as it does not capture explicitly the crucial role that individual objects, situations and events play in carrying information, and it is these individual things that actually carry information. In the above example, it is the individual event namely "John is awarded a grade "A" for his Programming course" that carries the information "John has scored 70% or above for that course". Dretske's definition is based on probability, and a single event does not have a probability [7], and a type of events has. To extend Dretske's definition and therefore make such a concept accurate, let us define a few very basic notions first.

2.2 Random Variables

Definition 1

Let *s* be a selection process under a set C of conditions, O a possible outcome of *s*, O can therefore be of one of a number values, *i.e.*, the possible outcomes. O is said to be a random variable.

That is to say, a random variable is a variable that can hold one of a number of possible values at a time and which one of the values to be hold is determined randomly. For example, in a database, table Students contains attributes such as ID, Name and DOB. A random variable could be any one attribute or a collection of attributes of the Students table in the sense that for a randomly chosen tuple, the value of its ID cannot be pre-determined and can only be one of all the possible values for ID.

2.3 Random Events

Definition 2

Let *s* be a selection process under a set C of conditions, O a possible outcome of *s*, and such an outcome is called a *state*, and E the power set of all the possible values for O, *i.e.*, all the *states*, X is a random event if E ∋ X and there is a probability of X, *i.e.*, P(X).

For example, to select a student record from table Students randomly in database and the record being concerned with a particular student is a random event.

A random event has to occur within a "probability space", which we define below:

Definition 3

Let *s* be a selection process under a set C of conditions, O a possible outcome of *s*, E the power set of all the possible values for O, *i.e.*, all the *states*, and E ∋ *Xi* for $i = 1,\ldots, n$, P_s is the probability space of the random events X_i for $i = 1,\ldots, n$, if $P_s = \{P(X_1), P(X_2),\ldots, P(X_n)\}$ and $\sum P(X_i) = 1$.

Note that this notion is also useful for explaining what it means by "probability distribution" and the change of "probability distribution", which is necessary and sufficient for information to flow.

2.4 Particulars of Random Events

Furthermore, as mentioned earlier, Xu *et al.* pointed out that even though Dretske's definition was plausible, the role that individual events play in our looking at the information content of a state of affairs was overlooked. To amend this, Xu *et al.* [13] put forward a definition of particulars of a random event as follow:

Definition 4

Let *s* be a selection process under a set C of conditions, X a random event concerning *s*, X_i an instance of *s*, X_i is a particular of X if X_i is in a state Ω, written $\Omega = \text{state}(X_i)$, and X ∋ Ω.

As in the example above, to select a student record from table Students is a random variable, the record happens to

be John's is a random event, and one occurrence of John's record is a particular of the random event.

3. Information Content Inclusion Relations

Having defined the foundational concepts, we can now define the notion of "information content inclusion relation". As this notion formulates the intuitive notion of "the information content of a signal/data", we formulate the latter first.

3.1 Information Content of a State of Affairs

Data in a database may be seen as a type of signals, and as said earlier data may be seen as random events and random variables. A random event is also informally called state of affairs by Dretske in [4].

Definition 5

Let *s* be some selection process or mechanism the result of which is reduction of possibilities, and therefore be an information source, and *k* prior knowledge about *s*[1]

Let *r* be a random event, and r_i a particular of *r* at time t_i and location l_i;

Let *s*'s being *F* be a random event concerning *s*, and s_j some particular of *s*'s being *F* at time t_j and location l_j;

r_i carries the information that there must be some s_j existing at time t_j and location l_j, that is, the state of affairs of *s* is *F* at t_j and l_j, if and only if the conditional probability of *s*'s being *F* given *r* is 1 (and less than 1 given *k* alone).

Definition 6

That a particular r_i carries the information that a particular s_j exists can also be termed that the *information content* of r_i includes s_j, or in other words, s_j is in the *information content* of r_i.

3.2 Information Content Inclusion Relations (IIR)

The term, *information content inclusion relation*, was firstly put forward by Feng in 1998 [14]. We now give an amended definition below:

Definition 7

Let *X* and *Y* be a random event respectively, there exists an *information content inclusion relation*, IIR for short, from *X* to *Y*, if every possible particular of *Y* is in the information content of at least one particular of *X*.

3.3 Types and Sources of IIR

We observe that there are four types of IIR in terms of where a state of affairs takes place, and we list them and some of their sources in the table below:

Information Inclusion Relation - Information content of X includes Y, denoted IIR(X, Y)	Sources
X, Y are random events both in the database world	Syntactic relations between data constructs and data values
X is a database random event. Y is random event in the real world	"Semantic values" [15] of data
X is a real world random event. Y is a database random event	Rules and processes of database design and database operations
X,Y are random events both in the real world	Relations between real world objects and events, business rules

The first two types of IIRs above constitute the information content of data in a database. Furthermore, we observe that for a database to provide information and nothing else, all the four types and all IIRs must be consistent with one another. To elaborate this observation would require much more work, and thus we leave it till another paper later.

3.4 Rules for Inferences on and of IIR

IIR can be formally reasoned about. Modifying those presented in Xu *et al* [13], we present the following inference rules for reasoning about IIR.

"Sum": If $Y = X_1 \cup X_2 \ldots \cup X_n$, then IIR($X_i$, *Y*) for i = 1,…,n.

This rule says that if it is the disjunction of a number of random events, then a random event *X* is in the information content of any of the latter. A trivial case is where *X* and *Y* above are not distinct. The rest of the rules can be interpreted similarly.

"Product": If $X = X_1 \cap X_2 \ldots \cap X_n$, $Y = X_i$ for i = 1, …, n, then IIR(*X*, *Y*).

Transitivity: If IIR(*X*, *Y*), IIR(*Y*, *Z*), then IIR(*X*, Z).

Union: If IIR(*X*, *Y*), IIR(*X*, *Z*), then IIR(*X*, $Y \cap Z$).

Augmentation: If $W = W_1 \cap W_2 \ldots \cap W_n$, *Z* is the product of a subset of $\{W_1, W_2, \ldots, W_n\}$, IIR(*X*, *Y*), then IIR($W \cap X$, $Z \cap Y$).

Decomposition: If IIR(*X*, $Y \cap Z$) then IIR(*X*, *Y*), IIR(*X*, *Z*).

The above set of rules is proven sound and complete. The proofs can be found in [13].

4. Preparing the Information Base for Database Queries

Given a set of IIRs, all IIRs that are logically implied by them and therefore are derivable, which we call the "closure" of the former, constitute the information base for answering queries that are posed to a database.

4.1 The Closure of a Set of IIRs

Definition 7

Let F be a set of IIRs. F closure (denoted F^+) is the set of IIRs logically implied by F. $F \subseteq F^+$. If $F = F^+$, F is called

[1] Note that *k* here goes only as far as what counts as a possibility involved in *s*, and it is not concerned with whether an observer is able to learn and actually learns something about *s* by consulting something else such as *r*.

a complete set of IIRs in the sense that no more IIRs that are logically implied by F can be derived from it by using the IIR inference rules.

The F above are called the original IIR, which are identified by applying the definition of IIR directly to a variety of sources such as the real world, database systems and domain knowledge, and which are not those that are derivable by using the inference rules on known IIR. For example, the *referential integrity* of a relational database is a kind of constraints in a relational database, from which, original IIR can be derived.

To compute F^+ given F, we can compute instead X^+ for all X, where X is a random event, which is normally easier than computing F^+ directly. Once X closure is known, to know if IIR(X, Y) holds given F (*i.e.*, whether it is implied by F) is a matter of verifying if Y is in the X closure or not. If so, IIR(X, Y) holds. Otherwise, as far as the given F goes, IIR(X, Y) does not exist.

4.2 IIR Closure of a Random Event

All random events that are derivable by using the IIR inference rules on a given set of original IIR and therefore are in the information content of the given random event constitute the IIR closure of the random event. For example, "Student ID = B001" is a random event, and "Student Address = 1 High Street" is in its information content. Likewise, "Student Postcode = PA1 2BE" is in that of "Student Address = 1 High Street". Through Transitivity (see Subsection 3.4), "Student Postcode = PA1 2BE" is also in the information content of "Student ID = B001". All such random events as "Student Postcode = PA1 2BE" and "Student Address = 1 High Street" would constitute the IIR closure of "Student ID = B001". Let X denote "Student ID = B001", then we use X^+ to done the IIR closure of X.

Figure 1 shows how the information base for answering queries is identified and formulated by means the foundational concepts, IIR and inference rules for IIR.

5. A System for Querying a Database with IIR

With the idea of IIR and other associated notions just presented, we have created a system for reasoning about the *information content* of data whereby to help derive information in a database by drawing on Wang and Feng [16] and Eessaar [17]. Intuitively, the system works like this.

Let us re-iterate that to select a student from table Students is seen as a random event, and the term "particulars of a random event" is used to describe a single occurrence of a random event. For example, student John's record happens to be selected from table Students, and this particular occurrence of John's record being selected is a "particular" of the random event that the record happens to be John's. A random variable may be seen as an aggregation of random events. In a table, an attribute can be

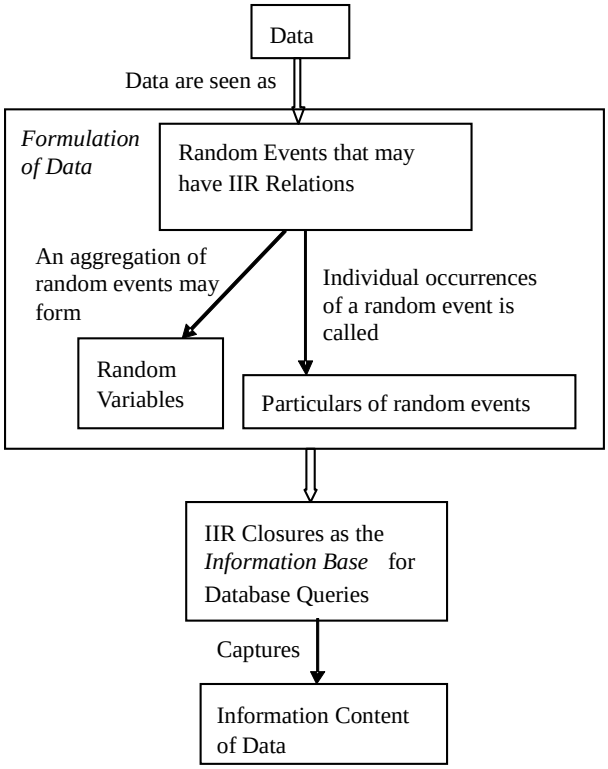


Figure 1. The information base for database queries

seen as a random variable because it normally contains many random events in it. For example, Student Name is a random variable, which contains Student Name being John and Student Name being Herman, among others. The IIR closure of Student ID being B001, for example, contains Student Name being "John", Student Major being "history" and Class Name being "BD445". If a user queries about the class name about John, the query can be answered by searching in this IIR closure of Student ID being B001. That is, once IIR closures are known, queries can be checked against these closures. This way some information that cannot be found by conventional queries may be discovered.

Figure 2 illustrates the structure of our experimental system. It consists of three main parts. The upper part is where users pose queries to the system. The middle part is the Datalog implementation of information content reasoning. The lower part shows a variety of sources of original IIR, namely domain knowledge and the syntactic and semantic properties of the database that are inherent to it.

The form of the queries is the conventional SQL. Most programming efforts were made on computing the IIR closures. The core algorithm is based on the IIR rules. Original IIRs were then added into the unit. This is one of the most difficult tasks in the programming required for the construction of the system as when more original IIR

were discovered more computation capability has to be added into the program such that the closures can continually increase accordingly. The output of the unit is simple however, which are IIR closures. User queries, then, are checked against these closures. Thus, more information can be discovered through queries.

The process of discovering original IIRs could be hard. There is a variety of sources out there that could potentially contain huge amount of original IIRs [13]. The two main sources though are domain knowledge and the properties of the database *per se*. The latter can be further divided into those of semantic and syntactic levels respectively. Hereinto, the syntactic level includes plenty of constraints such as data dependencies, integrity rules and the cardinality ratio between tables.

We now wish to demonstrate how the experimental system was created using IIR. The previous version of the system was coded in Oracle's PL/SQL [18]. It is now coded in the deductive database language, Datalog (Datalog Learning System, Universidad Complutense de Madrid, System v.1.6.2). First, we show how our IIR inference rules may be implemented by using Datalog in order to make use of the deduction power of it.

5.1 Datalog Implementation of IIR Inference Rules

It will now be shown that the above inference rules may be coded as *rules* (with example facts) in the Datalog language.

5.1.1 Conventions

1) Random Events

Any lower case literal is considered a constant in Datalog. We shall conflate these to random events or products of random events. In general, a Datalog constant, $x \equiv X$, where x is a Datalog constant, and X is a random event, which in the case of a database could be an attribute being a particular value extant in a database.

2) Information Content Inclusion Relation (IIR)

IIRs may be expressed as a relationship between either random events (Datalog constants), or Datalog variables, where the Datalog variable, when evaluated will contain a Datalog constant, and therefore, by extension a representation of a random event. We shall adopt the convention of using the predicate iir to indicate or derive an IIR between Datalog constants. Hence these will have the form:

iir(a,b).

iir(X,Y).

iir(a,X).

iir(X,a).

where a and b are Datalog constants, and X and Y are Datalog variables.

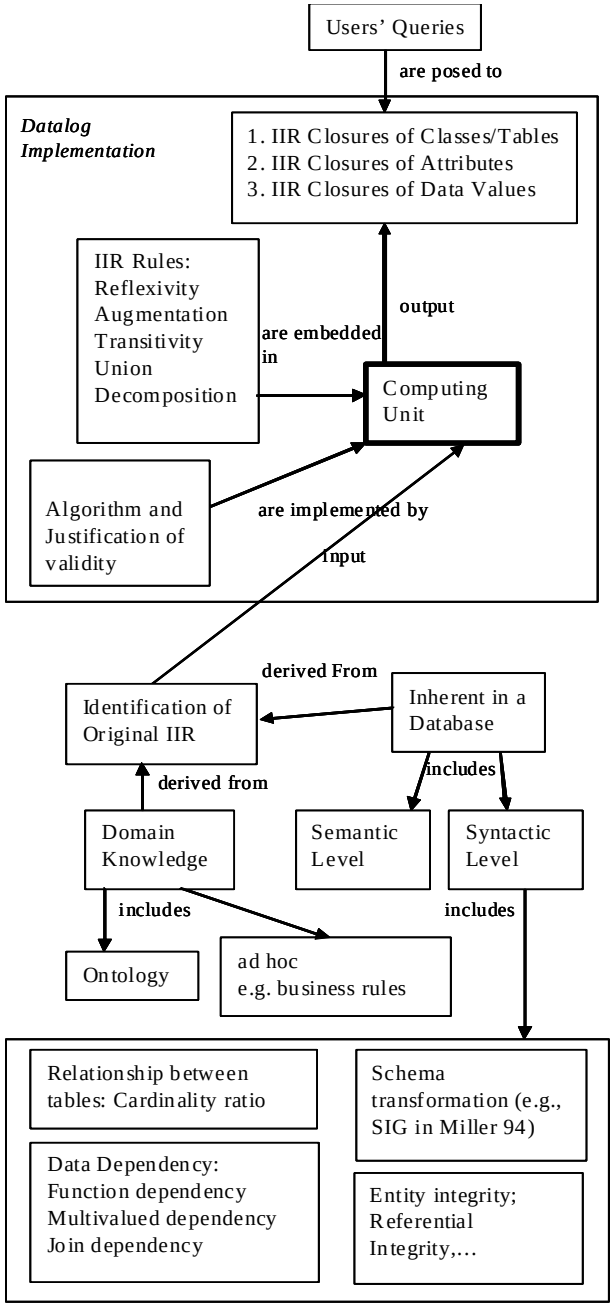


Figure 2. A system for reasoning about information content of data in a database

3) Products of Random Events

If we have IIR($A \cap C \cap D, B$), then $A \cap C \cap D$ represents the product of random events A, C and D containing the information content of random event B. This product may be represented in Datalog in the following manner:

product(p*ACD*,a,c,d).

where we have adopted the following conventions, a, c, d and p*ACD* are Datalog constants. We assume $a \equiv A$, $c \equiv C$,

and $d \equiv D$, and the single Datalog fact product(*pACD*, *a*, *c*, *d*)is equivalent to

$$pACD = A \cup C \cup D.$$

4) Sums of Random Events

If we have $L = P \cup Q \cup A$, then random event *L* represents the sum of random events *P*, *Q* and *A*. That is, we use L to refer to a random event where at least one of *P*, *Q* and *A* is extant. We choose to represent the sum in Datalog in the following manner:

```
sum(l,p).
sum(l,q).
sum(l,a).
```

where we have adopted the following conventions l, p, q and a are Datalog constants, and we assume that $l \equiv L$, $p \equiv P$, $q \equiv Q$ and $a \equiv A$. The three facts above may be considered as $L = P \cup Q \cup A$. The reason we have adopted this convention is that it allows the construction of sums containing two or more than two random events, and allows their eventual evaluation as binary relations, between any two random events. As Datalog is just the evaluation of a series of Horn clauses, then such structures are evaluated in conjunction for validity. This is exactly the effect we desire as random events that may happen, may be considered as unions of the state space for those random events, and thus may be treated as conjunctions happening across all random event closures. This also makes the coding in Datalog considerably simpler and more elegant.

5.1.2 IIR Inference Rules

1) "*Sum*" *Rule*

In Subsection 3.4, the "Sum" rule was given: If $Y = X_1 \cup X_2 \ldots \cup X_n$, then IIR($X_i$, Y) for $i = 1,\ldots,$ n. With the convention above, to code the "Sum" in Datalog, we use the rule:

```
iir(X, Y):-sum(Y, X).
```

This indicates that the information content of the sum Y is contained in any of the sum's members, denoted X. So given

```
sum(l, p).
sum(l, q).
sum(l, a).
```

if we run the query 'iir(X,l)?', we get the response:

```
{
  iir(a,l),
  iir(p,l),
  iir(q,l)
}
Info: 3 tuples computed.
```

This indicates iir(a,l), iir(p,l) and iir(q,l), respectively.

2) "*Product*" *Rule*

In Subsection 3.4, the "Product" rule was given: If $X = X_1 \cap X_2 \ldots \cap X_n$, $Y = X_i$ for $i = 1, \ldots,$ n, then IIR(*X*,*Y*). With the convention given earlier for products, in

```
product(pEG,e,g).
```

e, and g are Datalog constants, pEG represents the product of random events E and G, $pEG = E \cap G$

The "Product" rule may now be represented by the Datalog rule as follows:

```
iir(P,X):-product(P,X,A).
iir(P,X):-product(P,A,X).
```

This depicts, that any product *P*, has an IIR with any member of that product. The variable *A* represents a place holder, indicating to Datalog that for any matching predicates, then this variable is not to be returned in the query. In answer to the query, iir(pEG,X), asking what is in the information content of product of random event, *E* and *G*, Datalog returns the following:

```
{
  iir(pEG,e),
  iir(pEG,g)
}
Info: 2 tuples computed.
```

These indicate IIR($E \cap G$,*E*) and IIR($E \cap G$,*G*) respectively.

In general, to represent the product rule for a product consisting of *n* random events, then an additional *n* rules are required to show the product rule for a set of random events.

3) *Transitivity*

Assuming that we have IIR(*C*,*A*), IIR(*A*,*B*) which may be represented by the following Datalog facts:

```
iir(c,a).
iir(a,b).
```

where we have adopted the following conventions, c, a, and b are Datalog constants, X, Y and Z are Datalog variables. We assume $c \equiv C$, $a \equiv A$, and $b \equiv B$. The rule required in Datalog to represent transitivity may now be coded in Datalog as the following:

```
iir(X,Y):-iir(X,Z),iir(Z,Y).
```

and iir(X,Y)represents an IIR between two random events X and Y. This rule states that if any random event contains in its information content a second random event, which in turns contains in its own information content a third random event, then the first has the third in its information content. In answer to the query, iir(c,X), asking what is in the information content of random event, *C*, Datalog returns the following:

```
{
  iir(c,a),
  iir(c,b)
}
Info: 2 tuples computed.
```

These indicate IIR(*C*,*A*) and IIR(*C*,*B*) respectively, and

the latter is arrived at due to Transitivity.

4) *Union*

In Subsection 3.4, the Union was given: If IIR(*X*, *Y*), IIR(*X*, *Z*), then IIR(*X*, *Y*∩*Z*).

We now need to represent the relationship between these random events in Datalog, which may be done with the following facts:

```
iir(a,b).
iir(a,c).
product(pCB,c,b).
```

c, b, and a are Datalog constants. We assume c ≡ *C*, b ≡ *B*, and a ≡ *A*, and pCB represent the products of the random events *C* and *B*, such that *pCB*=*C*∩*B*.

We now need to create a Datalog rule which will link the product of random events *C* and *B* to random event *A*.

```
iir(X,P):-product(P,Y,Z),iir(X,Y),product(P,Y,Z),iir(X
,Z).
```

This will return all products of random events that have contain a random event with an existing IIR with the first argument.

In answer to the query, iir(a,X), asking which random events and products of random events are in the information content of random event *A*, Datalog returns the following:

```
{
  iir(a,b),
  iir(a,c),
  iir(a,pCB)
}
Info: 3 tuples computed.
```

These indicate IIR(*A*,*B*), IIR(*A*,*C*) and IIR(*A*,*B*∩*C*) respectively, and IIR(*A*,*B*∩*C*) is arrived at due to the Union rule.

5) *Augmentation*

Augmentation is a little more involved. Let us assume*W*=*X*∩*Y*∩*Z*, M = X∩Y, and IIR(*A*,*B*). We code these in Datalog as the following facts:

```
iir(a,b).
product(m,x,y).
product(w,x,y,z).
```

where w, x, y, z, a and b are Datalog constants.

We assume w ≡ *W*, x ≡ *X*, y ≡ *Y*, z ≡ *Z*, a ≡ *A*, b ≡ *B*, and iir(a,b)represents *IIR*(*A*,*B*), product(m,x,y) represents M = X∩Y, product(w,x,y,z) represents *W*=*X*∩*Y*∩*Z*. Then according to Augmentation, if *IIR*(*A*,*B*), M is a subset of W, then IIR(A∩W, B∩M). In general, to implement Augmentation we must use the following Datalog rules:

```
iir(P,W1∩Y):-iir(X,Y),product(W,W1,W2,W3),produ
ct(P,X,W).
iir(P,W1∩Y):-iir(X,Y),product(W,W1,W2,W3),produ
ct(P,W,X).
```

For W2 and W3 we would have a similar pair of Datalog rules.

If there are further products containing differing numbers of random events, then augmentation rules must be created for these as well. In general there will be two additional rules for each defined product of *n* members.

The two rules above effectively gives IIR from the product *M* to a product between random events *W* and *Y*. We need some further rules to show these as binary relations, to allow the further uncovering of available IIR. This is allowable as we have no other iir predicates with three arguments, so only those relationships, arising from the representations of random events being involved in augmentation will be evaluated. So to derive binary relationships the additional rules required are:

```
iir(P,W):-iir(P,W∩Y).
iir(P,Y):-iir(P,W∩Y).
```

This states, that the first argument, *i.e.* a random event, contains the information content, of another random event, if that latter random event is in a product derived from the augmentation rules above. There are two instances of the rule to allow both parts of the product to be uncovered. In answer to the query, iir(m,X), asking which random events are contained in the information content of random event *M*, Datalog returns the following:

```
{
  iir(m,b),
  iir(m,w)
}
Info: 2 tuples computed.
```

Indicating IIR(*M*,*B*) and IIR(*M*,*W*), respectively. Note that the product *W* will be further decomposed by the product and decomposition rules.

6) *Decomposition*

According to Decomposition, if IIR(D,E∩G), then IIR(D, E) and IIR(D, G). This may now be coded in Datalog as follows.

```
iir(d,pEG).
product(pEG,e,g).
```

where d, e, g and pEG are Datalog constants. We assume d ≡ *D*, e ≡ *E*, and g ≡ *G*, product(pEG,e,g) is equivalent to *pEG*=*E*∩*G*. To decompose this product we must use the Datalog rules:

```
iir(X,Y):-product(P,Y,A),iir(X,P).
iir(X,Y):-product(P,A,Y),iir(X,P).
```

This states, that the first argument, *i.e.*, a random event, contains the information content, of another random event, if that latter random event is in a product which is in the information content of the random event, represented by the first argument. There are two instances of the rule to allow for unordered evaluation. In answer to the query, iir(d,X), asking which random events are contained in the information content of random event *D*, Datalog returns

the following:

```
{
  iir(d,e),
  iir(d,g),
  iir(d,pEG)
}
Info: 3 tuples computed.
```

These indicate IIR(*D*,*A*), IIR(*D*,*G*) and IIR(*D*,*E*∩*G*), respectively. Note that the first two are created due to Decomposition.

We will now consider two examples herein. Firstly we shall consider a notional group of IIR and determine whether we can elaborate the closure, *i.e.*, all the pertinent (*i.e.*, logically implied) IIR arising from a set of specified IIR. Secondly we will consider a more real world example of a student database.

5.2 An Example of IIR Closures

Example 1

For our first example, we assume that the following IIR are given:

F={*IIR*(*A*∩*B*,*C*), *IIR*(*C*,*A*),
IIR(*B*∩*C*,*D*),
IIR(*A*∩*C*∩*D*,*B*),
IIR(*D*,*E*∩*G*),
IIR(*B*∩*E*,*C*),
IIR(*C*∩*G*,*B*∩*D*),
IIR(*C*∩*E*,*A*∩*G*) }

In addition, we assume following random events (Note that some of them are products/sums of some others):

W=*X*∩*Y*∩*Z*
M=*A*∩*W*,
L=*P* ∪ *Q* ∪ *A*,
T=*B*∩*D*∩*W*

In which as said before IIR(*X*,*Y*) is a simplified version of *I*(*X*)∋*Y*. Each upper-case letter stands for a random event in a notional database, extant at given spatial and temporal coordinates. That is, each random event is an entity in a database containing a specific set of attribute values. Additionally we adopt the convention that any intersection of random events, such as *A*∩*B* implies that both random events must have, or should have occurred concurrently, which results in a product of random events. Lastly the union of random events indicates that at least one of any of the random events in the union takes place, which results in a sum of random events.

Supposing we wanted to know the IIR closures of all combinations of the database entities (*i.e.*, random events) based on the above given IIR. It has been proved by Xu *et al*, 2008 that for the above IIR, the IIR inference rules given in Subsection 3.4 are sound and complete to derive all IIR that are logically implied by a given set of IIR.

We are now in a position to code the example, specified above. Here is the listing of the code.

1) Example Code

```
% Purpose of this program is to try and
% generate the IIR closure for
% specific set of random events given
% the original IIR

% facts
% =====
% Here is the IIR we wish to represent
% in Datalog

% 1. IIR(A∩B,C)
% 2. IIR(C,A)
% 3. IIR(B∩C,D)
% 4. IIR(A∩C∩D,B)
% 5. IIR(D,E∩G)
% 6. IIR(B∩E,C)
% 7. IIR(C∩G,B∩D),
% 8. IIR(C∩E,A∩G)
% 9. W = X∩Y∩Z
% 10. M = W∩A
% 11. L = P ∪ Q ∪ A

% To find the closure BDW+

%12. T = B∩D∩W

% The number of Datalog constants we
% employ are:
% t, a, b, c, d, e, g, l, m, n, w, x,
% y, z, pAB,
% pBC, pACD, pEG, pBE, pCG, pBD, pCE,
% pAG
% 23 constants in total

% 1. IIR(A∩B,C)
product(pAB,a,b).
iir(pAB,c).

% 2. IIR(C,A)
iir(c,a).

% 3. IIR(B∩C,D)
product(pBC,b,c).
iir(pBC,d).

% 4. IIR(A∩C∩D,B)
product(pACD,a,c,d).
iir(pACD,b).

% 5. IIR(D,E∩G)
product(pEG,e,g).
iir(d,pEG).
% 6. IIR(B∩E,C)
product(pBE,b,e).
```



```
iir(pBE,c).

% 7. IIR(C∩G,B∩D),
product(pCG,c,g).
product(pBD,b,d).
iir(pCG,pBD).

% 8. IIR(C∩E,A∩G)
product(pCE,c,e).
product(pAG,a,g).
iir(pCE,pAG).

% 9. W = X∩Y∩Z
product(w,x,y,z).
% 10. M = W∩A
product(m,w,a).

% 11. L = P ∪ Q ∪ A
sum(l,p).
sum(l,q).
sum(l,a).

% The final rule expresses the product
% BDW+. This allows us
% to find the closure for these three
% random events.

%12. T = B∩D∩W
product(t,b,d,w).

% rules
% =====

% product

% for a product of 2 members

iir(P,X):-product(P,X,A).
iir(P,X):-product(P,A,X).

% for a product of 3 members

iir(P,X):-product(P,X,A,B).
iir(P,X):-product(P,A,X,B).
iir(P,X):-product(P,A,B,X).

% Note: variables A and B are place
% holders in the above
% predicates - we are not interested
% in their content.

% We need no further product rules as
% there are no products
% containing more than 3 random
% events.

% transitivity

iir(X,Z):-iir(X,Y),iir(Y,Z).

% union

iir(X,P):-product(P,Y,Z),iir(X,Y),product(P,Y,Z),iir(X
,Z).

% augmentation

% We have products of 2 and 3 members
% so need two sets
% of augmentation rules.
% each of these requiring 2 (n +2)
% rules where n is the number in
% the product and two sets allows
% any ordering.

iir(P,W):-product(P,X,W),iir(X,Y),product(W,W1,W2,
W3).
iir(P,W):-product(P,W,X),iir(X,Y),product(W,W1,W2,
W3).
iir(P,W1):-product(P,X,W),iir(X,Y),product(W,W1,W
2,W3).
iir(P,W1):-product(P,W,X),iir(X,Y),product(W,W1,W
2,W3).
iir(P,W2):-product(P,X,W),iir(X,Y),product(W,W1,W
2,W3).
iir(P,W2):-product(P,W,X),iir(X,Y),product(W,W1,W
2,W3).
iir(P,W3):-product(P,X,W),iir(X,Y),product(W,W1,W
2,W3).
iir(P,W3):-product(P,W,X),iir(X,Y),product(W,W1,W
2,W3).

iir(P,Y):-iir(X,Y),product(P,W,X),product(W,W1,W2,
W3).
iir(P,Y):-iir(X,Y),product(P,X,W),product(W,W1,W2,
W3).

iir(P,W):-product(P,X,W),iir(X,Y),product(W,W1,W2
).
iir(P,W):-product(P,W,X),iir(X,Y),product(W,W1,W2
).
iir(P,W1):-product(P,X,W),iir(X,Y),product(W,W1,W
2).
iir(P,W1):-product(P,W,X),iir(X,Y),product(W,W1,W
2).
iir(P,W2):-product(P,X,W),iir(X,Y),product(W,W1,W
2).
iir(P,W2):-product(P,W,X),iir(X,Y),product(W,W1,W
2).

iir(P,Y):-iir(X,Y),product(P,W,X),product(W,W1,W2).
```



```
iir(P,Y):-iir(X,Y),product(P,X,W),product(W,W1,W2).


% sum

% rules (Note the variables A and B are
% place holders).
% this is a sum of 3 members.

iir(X,Y):-sum(Y,X).
```

We will now attempt to generate the closure for a specific product of $B \cap D \cap W$. We have represented this product by assigning this product to random event *T*. This generates the following set of tuples, in response to the query **iir(t,X)**:

```
{
  iir(t,a),
  iir(t,b),
  iir(t,c),
  iir(t,d),
  iir(t,e),
  iir(t,g),
  iir(t,l),
  iir(t,m),
  iir(t,pAB),
  iir(t,pAG),
  iir(t,pBC),
  iir(t,pBD),
  iir(t,pBE),
  iir(t,pCE),
  iir(t,pCG),
  iir(t,pEG),
  iir(t,w),
  iir(t,x),
  iir(t,y),
  iir(t,z)
}
Info: 20 tuples computed.
```

That is, all defined random events and their various products are in the information content of the product $B \cap D \cap W$, as expected, except for the additional members of the sum *L*.

5.3 What We Learnt from This Example

This example shows that we can use Datalog to implement all the inference rules that we developed for carrying out deduction on IIR. Moreover, this example also demonstrates that IIR closures can be computed by using Datalog. In the section that follows, we give the algorithm for computing IIR closures.

5.4 An Algorithm for Computing IIR Closures

We now give an algorithm for uncovering logically implied IIR as pseudo logic below.

Select random events

Create a product of the random events (as these events may be considered to have occurred simultaneously).

```
LOOP until
        OR any iteration gives the same product as before
        OR all available random events are included
        OR no further random events can be obtained for the product
        OR any remaining IIR do not contain any subset of the current product
                IF any random event of the product has an IIR transitive relation with further random events
                        Use the union rule to add these further random events to the product of random events
                END-IF
                IF any single random event and any sub-product of the product may be used in the IIR augmentation rule
                        Use the augmentation rule to add each member of the product to the product of random events.
                END-IF
                IF any random event of the product belongs to a union of random events
                        Use the sum rule and the union rule to add the sum to the product of random events
                END-IF
END-LOOP
```

The result product of random events is the closure of the set of original random events that was selected at the beginning of the algorithm.

Note the algorithm implicitly uses the decomposition rule, and product rule, when utilizing sub-products of the resultant random event product string.

5.5 An Example of Querying a Database Using IIR Closures

Example 2

The next example is taken from Wu and Feng [17], but we use our Datalog system to complete the job. This example is concerned with how to derive IIR upon an example of a real-world database. We show how our Datalog system accomplishes this task. Functional dependencies between attributes constitute a basis for IIR between values of attributes. The reasoning behind this is that an instance, *i.e.*, a tuple of an entity (represented by a *relation*) such as *student*, *class* and *enrolment* can be seen as a collection of particulars of some random events, and

moreover the random events may be involved in certain IIR. Some IIR are captured by functional dependencies between attributes. For example, attribute *student id* functionally determines attributes *surname*, *major*, *level* (or *year*) and *age*. Attribute value "*student id* being 100" *is a* random event, so is "*surname* being Smith", and moreover, the latter is in the information content of the former, which can be denoted as IIR("*student id* being 100", "*surname* being Smith"). This IIR is underpinned by the aforementioned functional dependency.

We now show how this example is coded up in Datalog as follows:

```
% Facts

student(100,smith,history,gr,25).
student(150,parks,geology,so,21).
student(200,baker,finance,gr,24).
student(250,glass,history,sn,19).
student(300,baker,geology,sn,20).
student(350,rosso,finance,jr,18).
student(400,bryan,geology,sr,22).

class(ba200,tth9,sc110).
class(bd445,mwf3,sc213).
class(bf410,mwf8,sc213).
class(cs150,mwf3,ea304).
class(cs250,mwf1,eb210).

enrollment(100,bd445).
enrollment(150,ba200).
enrollment(200,bd445).
enrollment(200,cs250).
enrollment(300,cs150).
enrollment(400,ba200).
enrollment(400,bf410).
enrollment(400,cs250).
enrollment(450,ba200).

businessRule(history,swimming).
businessRule(geology,diving).
businessRule(finance,basketball).

% rules
iir(X,Y):-student(A,B,X,C,D),businessRule(X,Y).

% functional dependencies

iir(X,Y):-student(X,B,C,D,E),enrollment(X,Y).
iir(X,Y,Z):-enrollment(A,X),class(X,Y,Z).

result(A,B,C,D,E,F,G,H,I):-
student(A,B,C,D,E),
iir(A,F),
iir(F,G,H),
iir(C,I).
```

If the program is run with the following query:

result(A,B,C,D,E,F,G,H,I).

It gives the following results:

```
{
result(100,smith,history,gr,25,bd445,mwf3,sc213,swimming),
result(150,parks,geology,so,21,ba200,tth9,sc110,diving),
result(200,baker,finance,gr,24,bd445,mwf3,sc213,basketball),
result(200,baker,finance,gr,24,cs250,mwf1,eb210,basketball),
result(300,baker,geology,sn,20,cs150,mwf3,ea304,diving),
result(400,bryan,geology,sr,22,ba200,tth9,sc110,diving),
result(400,bryan,geology,sr,22,bf410,mwf8,sc213,diving),
result(400,bryan,geology,sr,22,cs250,mwf1,eb210,diving)
}
Info: 8 tuples computed.
```

These are IIR closures arrived at of "Student ID being 100", "Student ID being 150", etc. respectively, which constitute the "information base" (see **Figure 1**) for queries. If a query is looking for "all the students that like diving", by checking the IIR closures, we get students Parks, Baker and Bryan.

5.6 What We Learnt from This Example

This example demonstrates that our approach works on real world situations. We code tuples in a database as Datalog "facts", and identify part of original IIR from functional dependencies between attributes of a relation, which are then used in coding Datalog "rules". Then IIR closures are computed, which serve as the information base for answering queries. Our Datalog system works exactly as expected.

6. Contributions of This Work

We observe that this work makes the following contributions to the field of databases. Fist of all, a justifiable approach to defining the notion of the "information content" of data with mathematical rigor was developed. This approach appears superior to intuitive approaches that we have seen thus far in the literature that is based on equating data with information.

Secondly, we have shown that reasoning about information content of data (rather than data *per se*) is possible, and this can be achieved by identifying a set of sound inference rules, and then these rules are implemented with a logic based system, *i.e.*, Datalog.

Thirdly, we also have demonstrated that information content based reasoning can go beyond "direct match" that conventional query answering employs. The former reveals information that the latter cannot find.

Therefore, in summary, we have constructed and tested an innovative approach to a fundamental problem in the field of databases, namely the information that data in a database carry. This may be seen constituting some significant value in further developing database theory and we also have shown that this is also applicable to real world problems.

7. Conclusions

In this paper, we have proposed a novel approach to the information content of data in a database. We gave a set of basic concepts and described an experimental system that makes use of this notion. A number of examples were used to test our system. With information sources outside a database imported into the system, the information content of a random event (data values) within the database expanded dramatically. Users could make the most of the information content of data by posing queries. Thus, more information can be discovered than conventional queries. The increase of random events' closures is based on the boost in original IIR and the inference capability using IIR. Identification of original IIR rules could be hard due to wide range of sources outside database. However, once original IIR have been identified and then integrated into the computing unit of our system, the system provides a powerful engine for users to query a database. Our experiment shows that with the IIR inference capability hidden information within database can be discovered with the increase of original IIR derived from database itself and external sources.

With IIR rules, we discussed the relation of information content inclusion between random events. Such a relation at a higher level, *i.e.*, that between random variables requires more work. How the relations on different levels are connected also deserves further investigation. The process of identifying original IIR was done manually, for which a semi-automated technique making use of meta-data to suit the need of a user is desirable and looks feasible. Moreover, how to approach and inference about the information content of data that are stored in independent and yet inter-operating databases should be investigated.

In summary, our work thus far seems to have shown that the information that a database can potentially provide is definable by using the notion of *information content inclusion relation* (IIR). Furthermore, the inference rules for formally reasoning about such a relation enables the development of a seemingly elegant way, by means of *IIR closure*, of identifying the information content of data in a database, which serves as a basis for answering queries.

8. Acknowledgments

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. Connolly and C. Begg, "Database systems: A practical approach to design, implementation, and manage- ment," Pearson Education, 2004.

[2] Y. Feng, "Database foundation," Hua Zhong Institute of Technology Press, 1986.

[3] J. Mingers, "Information and meaning: Foundations for an intersubjective account," Information Systems Journal, Vol. 5, No. 4, pp. 285–306, 1995.

[4] F. Dretske, "Knowledge and the flow of information," CSLI Publications, Stanford, 1999.

[5] S. Wang, C. Lin, and J. Feng, "A hermeneutic approach to the notion of information in IS," Proceedings of the 7th WSEAS International Conference on Simulation, Modeling and Optimization, pp. 400–404, 2007.

[6] J. Feng and M. Crowe, "The notion of 'Classes of a Path' in ER Schemas," Proceedings of 3rd East European Conference on Advances in Databases and Information Systems, Springer, Berlin, 1999.

[7] J. Barwise and J. Seligman, "Information flow: The logic of distributed systems," Cambridge University Press, 1997.

[8] H. Xu and J. Feng, "Towards a definition of the 'information bearing capability' of a conceptual data schema," Systems Theory and Practice in the Knowledge Age, Kluwer Academic/Plenum Publishers, New York, 2002.

[9] R. Hull, "Relative information capacity of simple relational database schemata," SIAM Journal of Computing, Vol. 15, No. 3, pp. 856–886, 1984.

[10] R. J. Miller, Y. E. Ioannidis, and R. Ramakrishnan, "The use of information capacity in schema integration and translation," Proceedings of the 19th International Conference on Very Large Data Bases, Dublin, Ireland, pp. 120–133, 1993.

[11] R. J. Miller, Y. E. Ioannidis, and R. Ramakrishnan, "Schema equivalence in heterogeneous Systems: Bridging theory and practice," Information Systems, Vol. 19, No. 1, pp. 3–31, 1994.

[12] L. Floridi, "Is semantic information meaningful data?" Philosophy and Phenomenological Research, Vol. 70, No. 2, pp. 351–370, 2005.

[13] 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, 2009.

[14] J. Feng, "The 'information content' problem of a conceptual data schema," Systemist, Vol. 20, No. 4, pp. 221-233, November 1998.

[15] A. Shimojima, "On the efficacy of representation," Ph.D. Thesis, Indiana University, 1996.

[16] Y. Wang and J. Feng J, "FCA assisted IF channel construction towards formulating conceptual data modeling," WSEAS Transactions on Systems, Vol. 6, No. 6, pp. 1159–1167, June 2007.

[17] E. Eessaar, "Guidelines about usage of the complex data types in a database," WSEAS Transactions on Information Science and Applications, Vol. 3, No. 4, pp. 712–719, April 2006.

[18] S. Urman, R. Hardman, and M. McLaughlin, "Oracle database 10G PL/SQL programming," McGraw-Hill/Osborne, 2004.

[19] X. Wu and J. Feng, "A framework and implementation of information content reasoning in a database," WSEAS Transactions on Information Science and Applications, Vol. 6, No. 4, pp. 579–588, April 2009.

J. Software Engineering & Applications, 2010, 3: 268-272
doi:10.4236/jsea.2010.33032 Published Online March 2010 (http://www.SciRP.org/journal/jsea)

A Study on Development of Balanced Scorecard for Management Evaluation Using Multiple Attribute Decision Making

Kwang Mo Yang[1], Young Wook Cho[2], Seung Hee Choi[3], Jae Hyun Park[4], Kyoung Sik Kang[5]

[1]Department of Industrial Engineering, Yuhan University, Puchoen, Korea; [2]Department of Technology & Systems Management, Induk Institute of Technology, Seoul, Korea; [3]Department of Industrial & Manufacturing Engineering, Pennsylvania State University, Pennsylvania, USA.; [4]Department of Qualification Trend Analysis, Human Resource Development Service, Seoul, Korea; [5]Department of Industrial & Management Engineering, Myongji University, Yongin, Korea.
Email: kmyang@yuhan.ac.kr

Received September 22nd, 2009; revised October 12th, 2009; accepted October 20th, 2009.

ABSTRACT

Recently, most businesses have introduced a system for improving their responsibility to the customers in terms of job improvement. For example, small-quantity batch production increases cost but improve efficiency of management. Companies have been introduced the balanced scorecard to evaluate their management as part of improvement, while they suffer from many trials and errors. Many businesses still have difficulty in introducing balance scorecard concept in their process, but we suggest a method to successfully introduce the balance scorecard. This study aims to suggest a new performance measurement model reflecting relative importance of the key performance indicators for each factor. Our model is applied to several companies in real-world to validate the new model. Also, our study proposes a methodology for an adequate performance measurement using multiple attribute decision making.

Keywords*: *Balanced Scorecard, Multiple Attribute Decision Making, Management Evaluation

1. Introduction

A large number of small and medium enterprises have realized it is necessary to employ a management evaluation system for increasing competitiveness, renovating their business system, and decreasing the cost. Unfortunately, the efforts and investment on the system do not seem to lead to the output. Consequently, it is necessary to develop a new methodology for efficient implementation and maintenance of a management evaluation system for reflecting both the department objective and the entire business goal. Balanced Scorecard (BSC) is a deliberately selected balanced set of measures derived from the vision and strategies that represent a tool for leaders to use in communicating strategies to the organization and motivating change [1]. Multiple attribute decision making (MADM) is one of the decision making methods to choose the alternative under multiple attributes [2]. If the BSC measuring achievement is applied to MAMD, a business could consider each attribute based on not the each department, but the vision and strategy of the entire business. Thus, this paper will suggest the method using BSC enabling to evaluate a management for insuring productivity in the real MAMD problem including more alternatives and attribute.

2. Related Work

2.1 Balanced Scorecard (BSC)

The current business environment is an era of mega-competition absolutely requiring a great measurement process and an excellent management method of administration performance. Measurement is a key factor in management. Kaplan and Norton [1] emphasized that the importance of performance measure by saying "You cannot control what you cannot measure." Balanced Scorecard is a deliberately selected balanced set of measures derived from the vision and strategies that represent a tool for leaders to use in communicating strategies to the organization and motivating change [1]. The concept of performance measure is accepted by private companies, first, and then, it has been spread to public institutions and non-profit organizations. The performance measurement system was traditionally limited to appearance emphasizing and growth-oriented aspects and financial measurement factors. However, the performance measure-

ment of an organization based on financial factors has showed a limitation as a means for delivering information on the quality and performance of administration. Many researchers have studied performance measurement based on financial factors to overcome the limitation. Currently, many companies have noted the BSC proposed by Kaplan and Norton [1] and have gradually applied and operated the BSC. For example, the research on the administration performance measurement method and management has been studied, actively, and there were remarkable development. However, the applied area is still limited to human resource organization. In this study, we estimate the weight reflecting the adequate importance of the BSC for lower Key Performance Indicators (KPI) (**Figure 1**) by using a Multiple Attribute Decision Making (MADM) based on the analysis of administration environments of company. In other words, we consider the weight reflecting practical features and suggest a desirable performance measurement model based on the weight. This study aims to suggest a new performance measurement model reflecting relative importance for the KPI for each aspect and apply the new performance measurement model to real business environment to validate the effect of the new model, identify any limitation, and suggest a methodology for proper performance measurement.

2.2 Multiple Attribute Decision Making (MADM)

Multiple attribute decision making (MADM) is one of the decision making methods to choose the alternative under multiple attributes [3]. An MADM problem could occur when we understand the management situation. Since a number of conflict factors are caused by the limited resources, MADM allows a decision maker to determine the factor among the variables with multi-attribute or the optimal environment to operation situation. To solve an MADM problem with a numerical approach, Barron and Schmidt [4] attempted to solve a problem with distance or fuzzy index. Dyer and Sarin [5], French [6], Haimes and Chankong *et al*. [7] suggested the interactive approach to improve the method using multi-objective liner programming. However, it was hard to keep the consistency and to guarantee the optimal solution. Analytic Hierarchy Process (AHP) [8,9] and Elimination Et Choice Translating Reality(ELECTRE) [10] became more complicated because the more attributes, the more coefficient by geometric progression. Cho [3] described the method to determine the optimal plant in the MADM problem having mixed attributes, such as nominal-the-better type, smaller-the-better type, and larger-the-better type. Although the method can not only decide the optimal plant but also solve the MADM having mixed attributes, it is possible only if each attribute is independent. In this paper, we put the priority on the management variables having the high mean value and the low difference of the weights of a certain factor by experts to understand management situation with Process Capability Index. Thus, we will suggest the method using BSC enabling to evaluate a management for insuring productivity in a real MADM problem including more alternatives and attributes.

3. Management Evaluation Formula

Management evaluation method is based on balanced scorecard with multiple attribute. It is consider the subjective and objective attributes (**Figure 2**). The objective attributes are the element calculated with the target data and real data observed. The subjective attributes are the sub variables under the basic four aspects in the BSC.

The *v* is defined as decision making matrix, having each

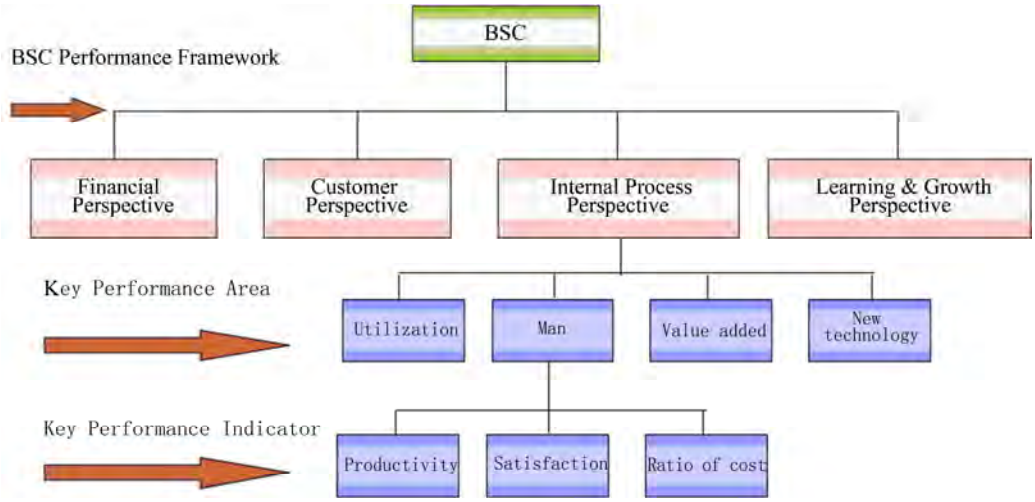
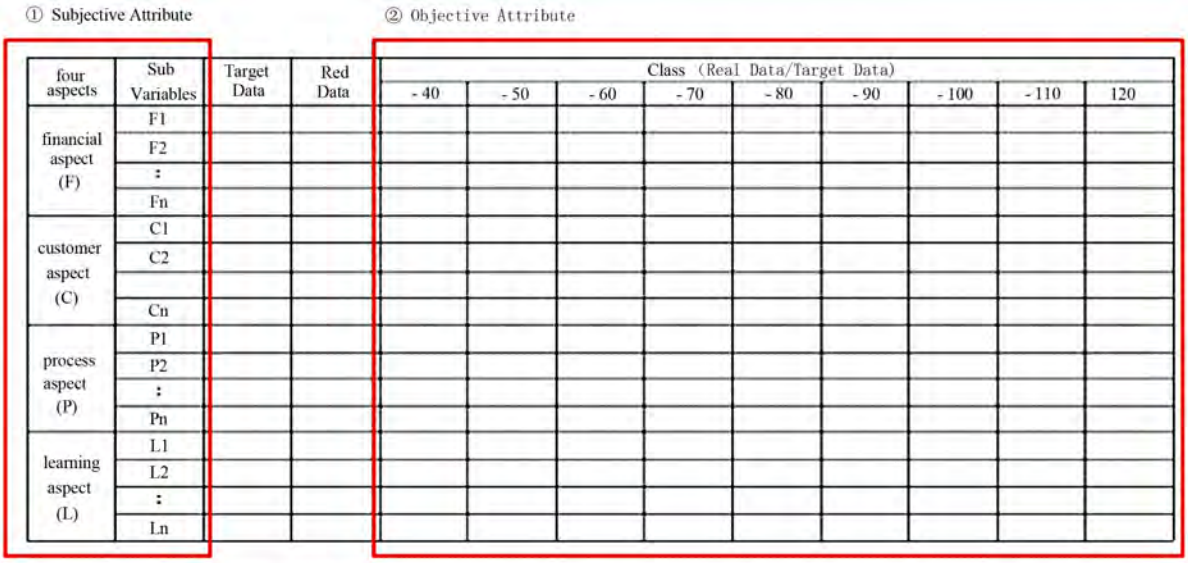


Figure 1. Example of KPI and four aspects of the BSC



four aspects	Sub Variables	Target Data	Red Data	Class (Real Data/Target Data) - 40	- 50	- 60	- 70	- 80	- 90	- 100	- 110	120
financial aspect (F)	F1											
	F2											
	:											
	Fn											
customer aspect (C)	C1											
	C2											
	Cn											
process aspect (P)	P1											
	P2											
	:											
	Pn											
learning aspect (L)	L1											
	L2											
	:											
	Ln											

Figure 2. BSC checklist form (per department)

department of m and reconsideration attribute of l connected with this as following:

$$v = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_i \\ \vdots \\ A_m \end{array} \begin{array}{c} \begin{array}{cccccc} X_1 & X_2 & \cdots & X_j & \cdots & X_l \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1j} & \cdots & x_{1l} \\ x_{21} & x_{22} & \cdots & x_{2j} & \cdots & x_{2l} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{i1} & x_{i2} & \cdots & x_{ij} & \cdots & x_{il} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mj} & \cdots & x_{ml} \end{bmatrix} \end{array}$$

where, A_i = ith department, X_j = jth attribute, x_{ij} = evaluation value X_j of for attribute in department A_i.

3.1 Subjective Attribute Formula

3.1.1 Weighted Decision to Each Subjective Attribute

It is very difficult to assign the weight to each attribute in an MADM problem. Since last selection crystallization can change according to the weight given, the weight should be assigned, deliberately. In this paper, we decide the weight based on the suggestion of experts, and the method determining the weight could be used for process capacity index. Process capability is the process characteristic ability that reflects how identical product can be produce according to manufacturing process established in product design process, which means uniformity of the product. To estimate characteristic ability, various statistical methods have suggested. Evaluating process capacity by variables of process and specification limit of product is known as process capacity analysis, and the process capacity analysis can be expressed in terms of process capability index (C_p). The process capability index is developed based on the concept of 6σ and applied first to industry field.

$$C_p = \frac{USL - LSL}{6\sigma} = \frac{T}{6\sigma} \quad (1)$$

For a single specification, the limit is defined as following.

For an upper specification, the limit is:

$$C_p = \frac{USL - \mu}{3\sigma} \quad (2)$$

For a lower specification, the limit is:

$$C_p = \frac{\mu - LSL}{3\sigma} \quad (3)$$

In this paper, we will use the process capability index for a lower specification limit only. The weight for each attribute is assigned by the data that several experts decide to each attribute. The evaluation data of experts for each attribute is determined by experts scoring from 9 to 1. The mean of data (μ) that experts decide can be calculated. The lower specification limit is 1 that experts decided absolute minimum, and the standard deviation that each expert decides is σ which is following.

$$\hat{\sigma} = s = \sqrt{\frac{\Sigma(b_{jp} - \bar{b})^2}{n-1}} \quad (p = 1, 2, ..., n) \quad (4)$$

where, b_{jp} is the mean that expert P decided data for each attribute j. And then, in this paper, we put the priority order on the attribute X_j having the high mean value and the low difference of the weights by the decision of experts. The values decided by experts are calculated by

Equation (4), normalized by Equation (5), and represented as NC_p(Normalized Capability Index).

$$NP_{pj} = \frac{C_{pj}}{(C_{p1} + C_{p2} + ...C_{pi})} \tag{5}$$

NC_p is defined the weight for each attribute, and the notation is replaced to w, where, w is under a certain criterion, such as following:

$$w = w_1, w_2, ...w_l, \qquad \sum_{j=1}^{l} w_j = 1$$
$$\text{here,} \quad w_j = C_{pi} / \sum_{j=1}^{l} C_{pi} \tag{6}$$

3.2 Objective Attribute Formula

3.2.1 Normalization of Evaluation Value Matrix for Objective Attribute

Evaluation value x_{ij} for attribute X_j in each department A_i is considered as profit attribute or cost attribute by normalized step, and the quantitative values of attributes are also normalized in the same intervals. For example, the profit attribute, high preference as evaluation value, is normalized as following:

$$r_{ij} = x_{ij} / (x_{1j} + x_{2j} + ... + xi_j + ... + x_{mj})$$
$$(i = 1, 2, ..., m; \quad j = 1, 2, ..., l) \tag{7}$$

Otherwise, cost attribute, low preference as evaluation value, is normalized as following:

$$r_{ij} = (1/x_{ij}) / [(1/x_{1j}) + (1/x_{2j}) + ... + (1/xi_j) + ... + (1/x_{mj})]$$
$$(i = 1, 2, ..., m; \quad j = 1, 2, ..., l)$$
$$\text{here}, 0 \le r_{ij} \le 1 \tag{8}$$

And then, we can make matrix R as following based on the normalized values.

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1l} \\ r_{21} & r_{22} & \cdots & r_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{ml} \end{bmatrix}$$

This paper presents the process of calculation only for the financial aspect factor, and the processes for the rest aspects are regarded identical. If $PR(F)_i$ is preference rate for department i, $PR(F)_i$ is weighted mean of attribute for process i.

$$PR(F)_i = \sum_{j=1}^{l} Fw_j \times NF(i)_j \tag{9}$$

where,

$$\sum_{i=1}^{m} PR(F)_i = 1$$

$NF(i)_j$ is the normalized data of department in financial attribute j. According to the result calculated by Equation (9) for each department, most high preference rate had department select and then if free department is department that had optimum the priority order as following:

$$\max PR(F)_i = \max(PR(F)_1, PR(F)_2, ..., PR(F)_m) \tag{10}$$

In this model, we assume each factor is independent each other. Similarly, the data for customer aspect, internal process aspect, and learning/growing aspect can be estimated. The proposed model has evaluated in a department performance in management environment. The result is applied to the simulated operation of CLV (Customer Lifetime Value) [11]. The primary evaluation criterion is the BSC which consists of financial aspect, customer aspect, internal process aspect, and learning/growing aspect. Also, in this paper, it will be able to be applied to MADM for deciding weigh of each variable. Therefore, we can summary the step to evaluate data for each department as following:

Step 1) The variables to assign weight are divided into financial aspect, customer aspect, internal process aspect, and learning/growing aspect.

Step 2) The data rank of each decided variable is determined by Group Consensus.

Step 3) Find sub-variables in the higher level variables. The sub-variables can be changed by the condition of the enterprise.

Step 4) Assign the weight for each factor by using MADM.

Step 5) Decide total evaluation data for the department by using Equation (11).

$$Total\ Evaluation\ Data = aPR(Fi) + \beta PR(Ci) + \gamma PR(Pi) + \delta PR(Li) \tag{11}$$

here

$0 < \alpha < 1, 0 < \beta < 1, 0 < \gamma < 1, 0 < \delta < 1$

$\alpha + \beta + \gamma + \delta = 1,$

$F_i > 0, P_i > 0, L_i > 0, C_i > 0$

where

α: financial aspest weight

β: customer aspect weight

γ: internal process aspect weight

δ: learning/growing aspect weight

$PR(F_i)$: preference rate of financial aspect

$PR(C_i)$: preference rate of customer aspect

$PR(P_i)$: preference rate of internal process aspect

$PR(L_i)$: preference rate of learning/growing aspect

The result in **Figure 3** is drawn by applying the suggested balanced scorecard to a real company. In **Figure 3**, we compare the evaluation data for each aspect and show the total evaluation data.

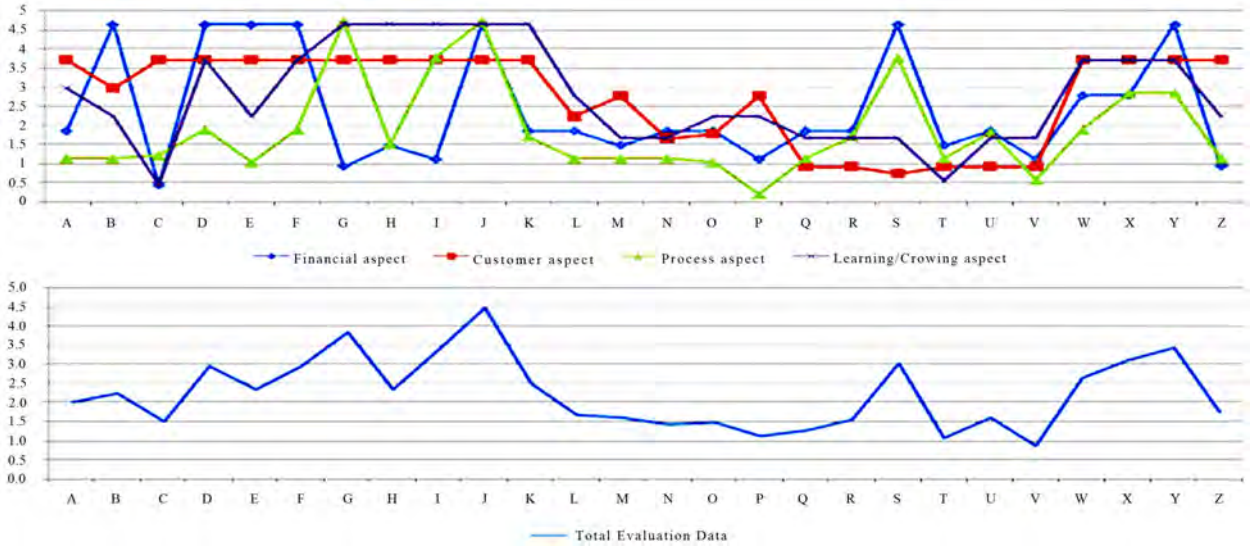


Figure 3. Result of management evaluation data for each department in a company

4. Conclusions

Among the methods to solve multiple attribute decision making with balanced scorecard for management evaluation for a department, numerical approaches offer the optimal solution, but fail to reflect the opinions of decision makers and CEO. Besides the current interactive approaches making up for the weak point fail to keep the consistency and to insure the optimal solution since it is hard to consider the entire alternative and attribute, simultaneously. Moreover, the increase in the number of attributes grows the amount of information due to pair wise comparison between the alternatives. In this paper, we propose the balanced scorecard method to assign the attribute weight by an expert group in the multiple attribute decision making including more alternatives and attributes. Also, we suggest the management evaluation method that assigns more weight on the attribute having the high mean weight by experts and the low difference or consensus in the evaluation. The proposed method could contribute to developing a good approach to reflecting both the optimal solution and the strategy of the entire business.

REFERENCES

[1] R. S. Kaplan and D. P. Norton, "Transforming the balanced scorecard from performance measurement to strategy," Accounting Horizons, Vol. 15, No. 2, pp. 87–104, 2001.

[2] K.-M. Yang, Y.-W. Wook, and J.-H. Park, "A study on evaluation method for process safety using multiple attribute decision making," 38th International Conference on CIE, Las Vegas, USA, pp. 2274–2278, 3–6 August 2008.

[3] Y.-W. Cho, "Selecting the optimal preferred facilities with multiple characteristics using Loss Function," Korea Safety Management & Science, Vol. 2, No. 2, pp. 127–135, 2002.

[4] H. Barron and C. P. Schmidt, "Sensitivity analysis of additive multi-attribute value models," Operations Research, Vol. 36, pp. 122–127, 1988.

[5] J. S. Dyer and R. K. Sarin, "Measurable multi-attribute value functions," Operations Research, Vol. 27, No. 4, pp. 810–822, 1979.

[6] S. French, "Decision theory: An introduction to the mathematics of rationality," Ellis Horwood Series in Mathematics and its Applications, pp. 448, 1986.

[7] Y. Y. Haimes and V. Changkong, "Decision making with multiple objectives," Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, New York, No. 242, pp. 388–399, 1985.

[8] L. C. Lawrence and C. Dong, "On the efficacy of modeling multi-attribute decision problems using AHP and Sinarchy," Eurpean Journal of Operational Research, Vol. 132, No. 1, pp. 39–49, 2001.

[9] T. L. Saaty, "A scaling method for priorities in hierarchical structures," Journal of Mathematical Psychology, Vol. 15, No. 3, pp. 234–281, 1977.

[10] P. T. Harker and L. G. Vargas, "The theory of ratio scale estimation: Saaty's analytic hierarchy process," Management Science, Vol. 33, No. 11, pp. 1383–1403, 1987.

[11] K.-M. Yang, S.-H. Choi, J.-H. Park, and K.-S. Kang, "Development of correlation weight customer lifetime value using analytic hierarchy process," 36th International Conference on CIE, Taipei, Taiwan, pp. 5461–5465, 20–23 June 2006.

J. Software Engineering & Applications, 2010, 3: 273-279
doi:10.4236/jsea.2010.33033 Published Online March 2010 (http://www.SciRP.org/journal/jsea)

Exploiting Distributed Cognition to Make Tacit Knowledge Explicating

Mingrui He[1], Yongjian Li[2]

[1]School of Management and Economics of University of Electronic Science and Technology of China, Chengdu, China; [2]School of Management and Economics of Southwest Jiaotong University, Chengdu, China.
Email: mingrui0208@21cn.com; swtjlyj@sina.com.cn

Received September 1st, 2009; revised October 5th, 2009; accepted October 12th, 2009.

ABSTRACT

Distributed cognition is a new development trend of cognitivism, and is also a new research field of knowledge management. The study discusses that tacit knowledge explicating activity is a distributed cognitive activity, whose success depends on interaction of each of these factors in distributed cognitive system and none of the factor could be neglected. Further, the study exploits distributed cognition to explore how to design these factors in the system so that tacit knowledge explicating can be accomplished successfully.

***Keywords*:** *Tacit Knowledge, Distributed Cognition, Tacit Knowledge Explicating*

1. Introduction

In today's dynamic global economy, knowledge is viewed as a key strategic and competitive resource by organizations, and effective management of individual knowledge within the work place has become critical to business [1,2]. Growing interest in the management of knowledge within organizations has focused on the control of tacit knowledge, which can be retained within the firm as a source of possible competitive advantage [3,4]. The knowledge in employee's head (tacit knowledge) is accounting for 42% of organization total knowledge, by surveying the knowledge composition of Delphi Group [5]. Further, OECD's a report named *The Knowledge-Based Economy* has indicated that the best value to organization is tacit knowledge of individual. So how to exploit and manage tacit knowledge always is crucial to knowledge management.

With expanding of knowledge management study, more and more scholars realize that we should pay more attention to not only technique but also personal factors [6]. To grasp the essence and regulation of human cognition is indispensable to the study of knowledge management [7]. The breakthrough of modern cognitive psychology, especially the development of distributed cognition, provides new angle of view to study tacit knowledge explicating.

Firstly, this article attempts to bring some clarification to tacit knowledge. We give an overview of historical beginnings of tacit knowledge concepts. Before we discuss how to transfer tacit knowledge, it's necessary to understand what means the tacit knowledge discussed in this article. Secondly, the article will inventory the basic tenets of the concept of distributed cognition, then review the current studies of tacit knowledge explicating and discuss how to disclose the cognition activity is the most essential question. In the following section, we discuss how to disclose the cognition activity based on distributed cognition. In particular, we discuss that the change of each factor in function system would cause the change of the whole system, and each factor in the system is very important for the success of tacit knowledge explicating. If we want to make the cognitive activity successful, to design each factor in the system is necessary.

2. Tacit Knowledge

In 1958, Michael Polanyi [8] put forward the term (tacit knowledge) in his book named *Personal Knowledge*, he proposes his famous epigram "we know more than we can tell": Humans can undertake a range of activities, and thus in a key sense know how to do them, without necessarily being able to provide a complete or coherent account of their actions, their reasons for undertaking them or to explain to others how to undertake them, let alone to explain the laws of physics, biology and so on that underline them.

The term has been paid close attention by many scholars from the earliest times. Robert J. Sternberg and his colleague defined the term from the view of psychology

[9–11]. It is viewed as knowledge that generally is acquired with little support from other people or resource, as procedural in nature, and it has direct relevance to individual's goals. P. F. Drucker defined the term from the view of management [12]. Tacit knowledge can't be explained by language, only be confirmed by demonstration. The only way to study them is apperception and exercise. They root in experience and skills. And Nonaka believed that, tacit knowledge can have both technical and cognitive dimension, and it is high personalized and high situated, it includes individual thinking model, belief and mental model etc. Those models and beliefs are so deeply rooted that we are quite hard to perceive them. But when we are looking around the world, we always receive their huge impact [4]. Tacit knowledge, which is deeply rooted in action and context, can be acquired without awareness and is typically not articulated or communicated [13].

The notion of tacit knowledge is intuitively appealing and seems to be something that we all instinctively understand as the knowledge that people have in their heads, rather than knowledge that is written down and recorded [14]. However, as Day [15] notes, the "folk-psychology" notion of tacit knowledge is simplistic and leads to the expectation that tacit knowledge can easily be transferred simply by having the knowledge holder reflection and articulate the knowledge. In fact, the real tacit knowledge remains ambiguous, with researchers applying the term with a variety of meanings and characterizations [16]. So before discussing tacit knowledge explicating, the article attempts to bring up some clarification to tacit knowledge firstly.

The nature of tacit knowledge in a business context can be viewed as a continuum with structured, codified, or explicit knowledge at one extreme and unstructured, uncodified, or tacit knowledge at the other [17]. Actually, we discuss the tacit knowledge at the extreme of knowledge continuum is complete tacit knowledge, which means that people absolutely cannot perceive them, let alone explain or articulate them, such as mental models. Between one extreme of knowledge continuum and the other extreme of knowledge continuum, there is a kind of tacit knowledge which cannot be structured or codified, but people can perceive them. For instance, skilled baker can bake delicious bread, he knows he can do that, and others also know he can do that, however he cannot articulate how to do that. We discuss the tacit knowledge which he owns is a special kind of tacit knowledge. The kind of tacit knowledge usually is in form of individual skill. But only after people can perceive the skill, the skill can be fallen the kind tacit knowledge. Tacit knowledge discussed in the article is this kind of tacit knowledge. It's no meaning to discuss tacit knowledge explicating if people absolutely cannot perceive them. In fact, for business organizations this kind of tacit knowledge has more meanings.

So we describe this kind of tacit knowledge as follows, and tacit knowledge mentioned in the following text is this kind of tacit knowledge.

Tacit knowledge is difficult to be partially or totally coded by language or words in a particular situation. The definition reveals the main features of tacit knowledge. One is difficult to code the knowledge in a particular situation which means that maybe others can code it or maybe one can code it in another particular situation. Tacit knowledge is high personalized and situated, and its cost of transfer is so high. It is formed automatically by subconscious. Its forming and utilizing aren't controlled by willingness of subjective and are manifested by inspiration, skill, habit and belief, and so on. However tacit knowledge is not mysterious experience (Polanyi), it is just can not be partial or total coded by one in a particular situation.

3. The Concept of Distributed Cognition

Distributed cognition, which takes cognitive overview into consideration, is a new development trend of cognitivism and a new cognitive paradigm. In 1884, Dewey wrote that organisms do not deviate from environment. It's impossible to look mental activity as individual activity without any relations [18]. One's cognition should be built on interactive relationship of human and environment [19]. Hutchins explicitly defined distributed cognition as a new basic paradigm to rethink cognitive phenomena in all fields [20]. The chief theory and methodology of distributed cognition is that it emphasizes that analytical unit of design, individual in social or in some situation instead of individual who be thought that he plays cognitive activity only in his head, and functional relations among different factors in cognitive process form functional system [21].

Having learnt these methods from cognition science, anthropology, sociology and social psychology, distributed cognition holds that to know cognitive phenomena should be from functional system point of view, which is composed of individual, other individual and artifact, and so on. Those cognitive phenomena which cannot be known only from individual point of view are pinpointed in distributed cognition. It's particularly important that distributed cognition stresses interaction among individuals and technique tools in a specific cognitive activity [22]. So distributed cognition is a system made up of cognitive subjects and environment, a new analytical unit including all things in the cognitive activity [23], and an information processing of representation to inner and external [24].

Since it was born, distributed cognition has strong vitality. It not only has learnt many advantages of traditional cognition, but also has different features from traditional cognition. Firstly, distributed cognition takes all factors into consideration in cognitive activity, puts forward a new analytical unit which is built on functional relations

among different factors participating cognitive processing together, and forms functional systems which show different representation status among different media and at the same time harmonize these media. Secondly, distributed cognition emphasizes the distribution across individuals, artifacts and internal and external representations in terms of a common language of 'representational states' and 'media', and holds that cognition can be distributed not only within individual but also in media, cultures, social and time. Thirdly, distributed cognition also stresses the influence of social substance situation to cognition process. Finally, distributed cognition notes that communication, sharing, factors (human and artifacts, etc.) depends on each other, and artifacts play important role in distributed cognition. When artifacts are used by people, cognitive residue phenomenon will appear. As long as artifacts are applied to help cognitive action, the ability being trained in the action would be remained, even the artifacts have gone, the ability is still here and can support high level thinking.

4. Tacit Knowledge Explicating is a Cognitive Activity with Distributed Cognition

4.1 The Current Studies of Tacit Knowledge Explicating

The idea about knowledge transfer was firstly formulated by Teece in 1977 [25], he thought that technique transfer can help industry accumulate valuable knowledge and impel technology diffusion, the result can reduce technology gap among different areas. With further development of knowledge-based economy, the study of tacit knowledge explicating has been put on the agenda. Tacit knowledge has been studied that it can be explicated by deep talks including analogy, story and metaphor [4,26,27], can be attained and transferred by learning and informally communication among people working as technical innovation [28], and can be exploited and applied by a new applied information technology. Tacit knowledge in inter-web of organization can be distinguished by information retrieval system based on software proxy technology [29]. Cognition mapping is also a useful implement to transfer tacit knowledge [30]. Further, Zhang [30] made differential dynamic model of organization tacit knowledge and analyzed these primary controlling parameters to influent tacit knowledge diffusing. Gao [31] made a model transfer of tacit knowledge based on ontology. Liang [32] thought informal relationship network is a primary way to transfer tacit knowledge. Tang [33] thought knowledge has biological activity, and made knowledge fermentation model borrowing biology fermentation process. Of course, SECI model is the most influential in the field of tacit knowledge explicating, which was put forward by Nonaka and Takeuchi. Socialization, externalization, combination and internalization form a circle of knowledge transformation and creating.

The current studies of tacit knowledge explicating focus mainly on the methods of explicating, transfer model and technique, especially SECI model pushes up greatly the development of knowledge management theory and practice, and becomes one of important foundations of knowledge management theory [34]. However, it is a pity that SECI doesn't take cognitive psychology into account [35] when it provides a suit of analysis paradigm [36]. In recent years, cognitive psychology has been paid more and more attention to by scholars when they are studying tacit knowledge. The inner mechanism of tacit knowledge and implicit cognition has been discussed. Implicit cognition provides empirical evidence for tacit knowledge from psychology, and tacit knowledge provides theoretical basement from epistemology [37]. And cognitive structure of tacit knowledge is also discussed, which is comprised of implicit system mechanism, ingredients transferred mechanism and motivation mechanism [38]. However, most of the research works focus on the individual cognition of tacit and few discuss on the cognitive activity of tacit knowledge explicating. From the most fundamental terms, tacit knowledge explicating is one kind of cognitive activity. The success of tacit knowledge explicating depends on individual, artifacts, environment, cultures, etc. in the cognitive activity. How to reveal the cognition activity is the most essential question. Only after having discussed the activity which is like black box, we can made further research on how to increase the efficiency of tacit knowledge explicating.

4.2 Tacit Knowledge Explicating As Distributed Cognition

Tacit knowledge explicating activity can fall into two forms.

One is that individual with tacit knowledge can explicate the knowledge by his own explicit knowledge and artifacts, and it can be shown as **Figure 1**. In this form of activity, cognition is distributed within individual, among artifacts, in culture, in environment, through time and so on.

Another is that when individual with tacit knowledge communicate with others, his tacit knowledge can be explicated by others' explicit knowledge and artifacts, and it can be shown as **Figure 2**. In this form of activity, cognition is distributed within individual, among individuals, among artifacts, in culture, in environment, through time and so on. The second form is a general form for tacit knowledge explicating, so we discuss the second form in this article.

When cognition is considered as a distributed system, it opens up the process of tacit knowledge explicating to inspection. This is important for tacit knowledge explicating because inspection permits people to examine the

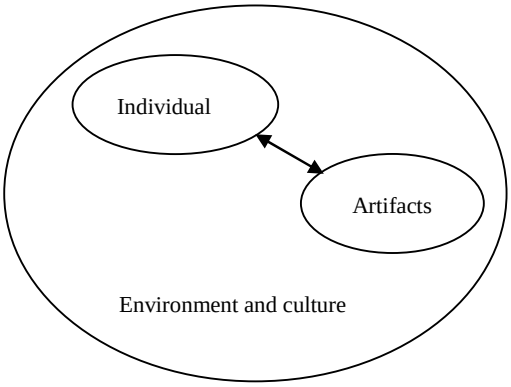


Figure 1. Form 1

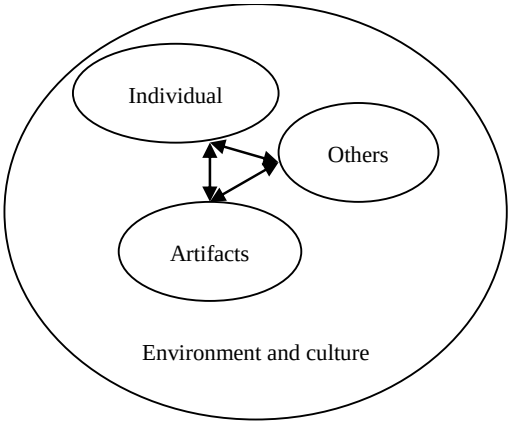


Figure 2. Form 2

variables involved in the activity and more importantly, distributed cognition, as a perspective with which to think, has the potential to enhance the likelihood of careful selections of tools by people for the activity.

In the activity, cognition is distributed:

1) Within individual: Cognition structure of individual is psychological basis to make tacit knowledge explicit;

2) Among individuals: For instance, storytelling is often thought as an effective way to make tacit knowledge explicit;

3) Among artifacts: Artifact is a core term in distributed cognition, which means tool, thinking and method and so on. When artifacts are applied, people's intelligence can be extended and people can be more intelligent and more effective. For instance, computers can make us simulate real world better, and symbols can make us express our thoughts and ideas better. Human inner cognitive ability and external artifacts together can greatly improve cognitive level [21];

4) In culture: Different individuals in the activity maybe are from different cultures, which causes they have different cognitive styles. Culture can be experienced by communicating face-to-face, and influents cognitive process indirectly. The human mind is more than the sum of localized (interiorized) cognition; our thoughts, capabilities, and actions are continuously shaped by, and co-evolve with, elements of the external world and the cultural contexts in which we operate [39];

5) In environment: Interaction of internal and external representations is influenced by environment. Any activity cannot be divorced from environment, including tacit knowledge explicating;

6) Through time: The products of earlier events can transform the nature of later events [20].

5. Analyzing Tacit Knowledge Explicating Activity Based on Distributed Cognition

When cognition is seen as distributed system, it has provided a very good method and angle of view to study the essence of tacit knowledge explicating which is like black box. During the cognitive activity, not only the individual with tacit knowledge should be paid attention to, but also other variables in this cognitive activity should be paid attention to [40], for instance, the cognitive state of others, the actions of others when they accept tacit knowledge, the characters of artifacts, culture state and environment state, and so on. The whole tacit knowledge explicating is a dynamic activity in which dynamic exchange is between individuals, between artifacts, between individual and artifact, etc. Of course, dynamic exchange is also in particular culture and environment. In the whole activity, the source of tacit knowledge is often considered as teacher, and the focal-point of tacit knowledge is often considered as learner.

The whole tacit knowledge explicating activity has cognitive distribution, which distribute within individual, among individuals, among artifacts, in culture, in environment and so on. Further, individuals, artifacts, culture and environment constitute a functional system. According to the functional system, the representation, transmission, and progress can be analyzed as follows by four analysis phases [41,42]. The method pays attention to collect raw data not only from different metaphysical and material levels, but also from the changes of representation state in the activity. So it can show the crucial moment which is maybe unclear and maybe neglected by traditional analysis, and can definitely reveal that the problem is caused by mutual influence of various factors.

In the first phase: Individual takes inner representation and forms some knowledge (for instance, individual skill) in his working memory system. He cannot express this knowledge (tacit knowledge) by any language or words, only could teach others by demonstrating over and over or using suitable artifacts. On one hand, learner undergoes instructions himself from the knowledge holder, on the other hand, he selects suitable artifacts to help himself learn it better, for instance, he can use video or audio to record situation. The records must be very detailed. To omit any subtle corner, color and odor can cause some significant information missing (be missed). In this phase,

inner representation, interaction among artifacts and social interaction between different inner representations are primary.

In the second phase: Learner always has different cognitive structures and knowledge structures from the knowledge holder. By observing the holder and using artifacts, he could translate all information representation by video or audio records into printing press representation, including language and other words. In this phase, the interaction of inner representation state and technique tool representation is primary.

In the third phase: Learner and his (her) partners would find significant behaviors and events from the printing press representation attained in the second phase. So the continuous behavior flow is divided into lots of significant blocks.

In the fourth phase: to explain these significant behaviors and events confirmed in the third phase, then translate into corresponding theory. So the result of explicating would be attained.

When we analyze tacit knowledge explicating by the method, we need to descript all aspects of behaviors and interactions, which are so trivial and apt to take for granted, and cannot neglect the important function of environment, culture and artifacts in the cognitive activity. Different from those methods which only care the main element causing problem, the method definitively reveal a problem caused by various factors [43]. By the method to analyze the tacit knowledge explicating activity, we can see that the cognitive activity cannot be successful without any one of the factors in the functional system.

6. Discussions

Distributed cognition theory holds that cognition distributes in functional system, which is made up of individuals, artifacts, environment and culture. And tacit knowledge explicating activity is unexceptional, too. In a functional system, the change of any of these factors would cause the change of system. For example, the change of sharing tacit knowledge culture would cause the failure of tacit knowledge explicating, the usage of wrong technique tools would cause the failure, and the inappropriate information representation transform of others also would cause the failure, etc. The success of tacit knowledge explicating activity depends on interaction of each factor in a functional system. How to translate temporary unstable cognitive distributed system into stable distributed cognitive resource is an inspiration for us. For instance, to build up long-term resource pool and to cleanse and analyze the result of explicating can help us translate distributed cognition into stable distributed cognitive resource and help to explicate tacit knowledge effectively.

Factors in distributed system must depend on each other to accomplish one task, so none of the factors could be neglected. Communication is a necessary condition of distributed cognition, and shared information is pooled information, which can make someone who has the best resource apply the information for other's benefits [21]. In tacit knowledge explicating activity, sharing culture has important influence on the success of the activity, too. Only in sharing culture, each factor can communicate with each other effectively and the specific person can apply the useful information to cause the success of the explicating activity. So how to build up sharing culture in tacit knowledge explicating activity is a very important task.

The function of artifacts in the system is not only as tool, but also as a teacher sometimes. Artifacts expand and support human's intelligence, even they are more effective in some special task. When artifacts are applied, cognitive residue phenomenon will appear. When individual must finish some task without these artifacts, cognitive residue can provide efficient service for individual. For instance, the recorders can record those micro motions and effects of learners in detail. Even without the knowledge holder instruction and demonstration, the records could make learners go on to study. The cognitive residue phenomenon would support learners to understand the skill effectively. So in tacit knowledge explicating activity, how to choose and design suitable artifacts should also not be neglected. Such as, how to apply information bank and how to apply symbols, etc.

Individual is at the center of distributed cognitive system as cognitive subject [44]. The success of explicating activity is bound up with the cognitive structure of individual. If the specific person hasn't the corresponding professional knowledge, he couldn't succeed to explicate the knowledge. So for the purpose of sharing tacit knowledge effectively, the receiver with corresponding cognitive structure is an important factor, too.

Distributed cognition theory and study not only promote the development of cognitive theory, but also provide a good angle of view to study management problem, for instance, the study in team management [7]. At the same time, it has significance to study tacit knowledge explicating. However, many problems still need to be discussed more. For instance, how to design and choose suitable artifacts in different situation of tacit knowledge explicating, and how to discriminate suitable receivers to join the explicating activity, and how to build up suitable culture and environment in organization to push the success of tacit knowledge explicating activity, and so on.

REFERENCES

[1] W. M. Cohen and D. A. Leventhal, "Absorptive capacity: A new perspective on learning and innovation," Administrative Science Quarterly, Vol. 35, No. 1, pp. 128–52, 1990.

[2] M. Ipe, "Knowledge sharing in organizations: A conceptual framework," Human Resource Development Review, Vol. 2, No. 4, pp. 337–59, 2003.

[3] R. W. Coff, D. C. Coff, and R. Eastvold, "The knowledge-leveraging paradox: How to achieve scale without making knowledge imitable," Academy of Management Review, Vol. 31, No. 2, pp. 452–465, 2006.

[4] I. Nonaka, "A dynamic theory of organizational knowledge creation," Organizational Science, Vol. 5, No. 1, pp. 14–37, 1994.

[5] Frappaolo, "Carl defining knowledge management: Four basic functions," Computerworld, 1998.

[6] R. L. Wang, X. M. Guo, and X. Q. Zheng, "The dimension and strategies of knowledge management," China Soft Science, Vol. 6, pp. 43–47, 2001.

[7] X. Wu and Z. M. Wu, "A study on team knowledge management based on shared mental model," R & D Management, Vol. 18, No. 3, pp. 9–15, 2006.

[8] M. Polanyi, "Personal knowledge:Towards a post-critical philosophy," University of Chicago Press, Chicago, 1958.

[9] R. J. Sternberg, "Successful intelligence," Plume, New York, 1997.

[10] R. J. Sternberg, G. B. Forsythe, J. Hedlund, J. A. Horvath, R. K. Wagner, W. M. Williams, S. A. Snook, and E. L. Geigorenko, "Practical intelligence in everyday life," Cambridge University Press, New York, 2000.

[11] R. J. Sternberg and J. Hedlund, "Practice intelligence, g, and work psychology," Human Performance, Vol. 15, No. 1–2, pp. 143–160, 2002.

[12] P. F. Drucker, "The new productivity challenge," Harvard Business Review, 1991.

[13] C. T. Matthew and R. J. Sternberg, "Developing experience-based (tacit) knowledge through reflection," Learning and Individual Differences, Vol. 19, No. 4, pp. 530–540, December 2009.

[14] M. D. Koenig, "Knowledge Management," International Encyclopedia of Information and Library Science, 2nd Edition, In: J. Feather & P. Sturges, Ed., Routledge, New York, pp. 351–359, 2003.

[15] R. E. Day, "Clearing up 'implicit knowledge': Implications for knowledge management, information science, psychology, and social epistemology," Journal of the American Society for Information Science and Technology, Vol. 56, No. 6, pp. 630–635, 2005.

[16] H. Taylor, "Tacit knowledge: Conceptualizations and operationalizations," International Journal of Knowledge Management, Vol. 3, No. 3, pp. 60–73, 2007.

[17] T. Gautschi, "The knowledge continuum," Design News-Academic Research Library, Vol. 54, No. 12, pp. 170, 1999.

[18] G. M. Zhou and X. L. Fu, "Distributed cognition: A new cognition perspective," Advance in Psychological Science, Vol. 10, No. 2, pp. 147–153, 2002.

[19] J. J. Gibson, "The ecological approach to visual perception," Houghton Mifflin, Boston, 1979.

[20] E. Hutchins, "Cognition in the wild," The MIT Press, 1995.

[21] P. Bell and W. Winn, "Distributed cognitions, by nature by design," East China Normal University Press, Shanghai, 2002.

[22] J. F. Ren and K. D. Li, "Distributed cognition theory and its application in CSCL system design," Audio-Visual Education Research, Vol. 136, No. 8, pp. 3–6, 2004.

[23] "Distributed cognitions: Psychological and educational considerations," In: G. Salomon, Ed., Cambridge University Press, 1993.

[24] J. Chuah, J. Zhang, and T. R. Johnson, "Distributed cognition of a navigation instrument display task," In: M. Hahn and S. C. Stoness, Ed., Proceedings of the Twenty-First Annual Conference of the Cognitive Science Society, Lawrence-Erbaumn, Mahwah, New Jersey, 1999.

[25] D. J. Teece, "Technology transfer by multinational Firms: The resource cost of transferring technological know-how," The Economic Journal, Vol. 87, pp. 242–261, June 1977.

[26] W. Swap, D. Leonard, M. Shields, and L. Abrams, "Using mentoring and storytelling to transfer knowledge in the workplace," Journal of Management Information System, Vol. 18, No. 1, pp. 95–114, 2001.

[27] S. J. Hitt, "Tacit knowledge contained in internet/ web-based discussion group message," The Union Institute, pp. 36–37, 2001.

[28] K. U. Koskinen and H. Vanharanta, "The role of tacit knowledge in innovation process of small technology companies," International Journal of Production Economics, Vol. 80, pp. 57–64, 2002.

[29] S. Barnes, "Knowledge management system: Theory and practice," China Machine Press, Beijing, 2004.

[30] S. T. Zhang, T. Li, and X. M. Duan, "Tacit knowledge transferred model research in organization," Science Research Management, Vol. 25, No. 4, pp. 28–32, 2004.

[31] X. Y. Gao, "A model transformation of tacit knowledge based on ontology," Information Studies: Theory & Application, Vol. 30, No. 1, pp. 41–45, 2007.

[32] Q. H. Liang and X. H. He, "Spatial clustering: The mechanism and path of tacit knowledge transferring and sharing," Management World, Vol. 3, pp. 146–147, 2006.

[33] J. S. Tang and J. S. He, "The knowledge fermenting models of organizational learning and individual learning," Scientific Management Research, Vol. 23, No. 1, pp. 86–88, 2005.

[34] C. Peng and H. B. Hu, "The mechanism of knowledge creation in knowledge alliance: BaS-C-SECI model," R&D Management, Vol. 20, No. 1, pp. 118–122, 2008.

[35] Z. C. Gao and S. K. Tang, "The analysis of mechanism of enterprise knowledge creating based on cognitive psychology," Journal of Information, Vol. 8, pp. 87–91, 2008.

[36] J. X. Chu and S. K. Tang, "A Q-SECI model based on the insight learning and its application," Science Research Management, Vol. 28, No. 4, pp. 95–99, 2007.

[37] J. Z. Liu, "The inner mechanism of implicit cognition and tacit knowledge," Studies in Dialectics of Nature, Vol. 15, No. 6, pp. 11–14, 1999.

[38] Z. Li and K. J. Zhang, "Tacit knowledge of cognitive structure," Social Sciences Journal of Hunan University, Vol. 4, pp. 38–41, 2007.

[39] B. Cronin, "Bowling alone together: Academic writing as distributed cognition," Journal of the American Society for Information Science and Technology, Vol. 55, No. 6, pp. 557–560, 2004.

[40] N. H. Schwartz, "Exploiting the use of technology to teach: The value of distributed cognition," Journal of Research on Technology in Education, Vol. 40, No. 3, pp. 389–404, 2008.

[41] F. Decortis, S. Noirfalise, and B. Saudelli, "Distributed cognition as framework for cooperative work [DB/OL]," 2005. http://www-sv.cict.fr/cotcos/pjs/TheoreticalApproaches/DistributedCog/DistCognition-paperDecortis.Htm.

[42] Y. Roger and J. Ellis, "Distributed cognition: An alternative framework for analyzing and explaining collaborative working," Journal of Information Tech- nology, Vol. 9, No. 2, pp. 119–128, 1994.

[43] Y. H. Yu, "Knowledge management and organizational innovation," Fudan University Press, Shanghai, 2001.

[44] D. N. Perkins, "Person-plus: A distributed view of thinking and learning," In: G. Salomon, Ed., Distributed Cognitions: Psycho1ogica1 and Educational Considerations," Cambridge University Press, 1993.

J. Software Engineering & Applications, 2010, 3: 280-286
doi:10.4236/jsea.2010.33034 Published Online March 2010 (http://www.SciRP.org/journal/jsea)

Scientific Research

Deriving Software Acquisition Process from Maturity Models—An Experience Report

Hussain Alfaraj, Shaowen Qin

School of Computer Science, Engineering and Mathematics, Flinders University, Adelaide, Australia.
Email: hussain.alfaraj@csem.flinders.edu.au

Received October 26th, 2009; revised December 15th, 2009; accepted December 20th, 2009.

ABSTRACT

The establishment of an existing practice scenario was an essential component in providing a basis for further research in the area of COTS software acquisition within the organisation. This report details the identification of means of describing the existing practice of software acquisition within an organisation and identification of models that could be used to present this view. The chosen best practices descriptions for the idealized model were maturity models, including SA-CMM, CMMI-ACQ, and ISO/IEC 12207. *This report describes these models briefly and then describes the process of identifying the requirements for idealizing these maturity models into process frameworks that could be identified to actually business process models from a real organisation in order to identify gaps and optimizations within the organisation's realization of the best practices model. It also identified the next steps in identification of the theoretical best practice framework, which will involve translation of the model to YAWL Petri nets and simulation of the process in order to identify potential modelling flaws or issues with framework efficiency. Implications of the currently ongoing research include the identification and correspondence of specific tasks and activities from ITIL and CoBiT frameworks with the generic key process areas of software acquisition frameworks and identification of sufficiently detailed structural framework models for each level in order to identify appropriate frameworks for application even in cases where these frameworks were not explicitly identified by the organisation or the researcher.*

Keywords: *BPM, Workflow, Software Acquisition, Simulation, CoBiT, ITIL*

1. Introduction

The researcher's current area of focus is on the derivation of the software acquisition model in use within organisations for commercial off the shelf (COTS) software acquisition using BPM tools. Two different areas of investigation were chosen for this analysis, including determination of an idealized model based on current best practices and the description of the actual practice within the organisation under study. While many organisations do attempt to undertake the development of processes under the best practices frameworks described below, many organisations do not succeed in this goal either due to deliberate divergence from the best practice in order to accommodate organisational realities or because of inability to reach the best practices condition for another reason [1].

Existing best practice models for software acquisition are built on commonly accepted standards that either represent software acquisition as a standalone process or integrate the acquisition process into the software lifecycle. The diversity of these models means that an organisation is likely to be able to choose an appropriate model for its needs, but none of the best practices models is likely to be fully adequate. Valuable information can be gained by comparing the process as enacted within the organisation with the template provided by the best practices model. This research required building theoretical models for three commonly used software lifecycle standards, the Capability Maturity Model for Software Acquisition (CMMI-ACQ), the Software Acquisition Capability Maturity Model (SA-CMM), and the ISO/IEEE 12207 software lifecycle standard (which integrates software acquisition into the lifecycle model as compared to the other two models, which describe it separately).

This report describes these models in brief and then focuses on the process of describing the idealized template for the oldest model, SA-CMM, in order to demonstrate the engagement of analytical tools. A discussion is also provided regarding the next steps in identification of organisational optimization and matches to this process. The importance of this report is that it provides a scientific understanding of the models as they relate to the software acquisition process. Organizations that have likely used

other methods in the past that may not have taken fully into account both the managerial issues and the tasks related to the best outcomes of software acquisition will be provided not only with a model for the process, but an understanding of the important elements of the model and how to integrate them into real-world application.

2. Best Practices and Standards

Three best practices models or standards were identified for inclusion based on the completeness of the standard and the level of actual use within these organisations. SA-CMM, the oldest model that is examined, is still in active use within some organisations, while others have enacted its successor, CMMI-ACQ [2]. Both of these models were developed by the Carnegie Mellon University Software Engineering Institute to support the development of processes in the organisation. The third model ISO/IEC 12207, describes an overall view of the software lifecycle that includes the acquisition process. Each of these models presents a foundation that is important to understand in order to examine the develop a model of the software acquisition process that integrates both management issues and key tasks and procedures that are to be followed for the opportunity for the best outcomes for an organization. The examination not only examines these theories, but also helps to bridge theory with real-world needs and concerns.

2.1 SA-CMM

The SA-CMM best practices standard (currently version 1.03, released 2002), was developed to provide a capability maturity model that could be used in the context of software acquisition [3]. Although it was developed for use by the United States Department of Defence, it has been widely used in educational and industrial contexts as well. Companies have found that the SA-CMM best practices standard provided a straight-forward process by which to understand the important issues related to software acquisition [3]. The SA-CMM model, which is shown in **Table 1**, is based on a five-layer model in which each level delimitates a different level of maturity. The maturity level is made up of key process areas, which include goals, institutionalization features (commitment to perform, ability to perform, measurement and analysis, and verification), as well as activities [3]. The SA-CMM model is a staged model, indicating that the results are cumulative—for example, it is necessary to meet the requirements of Level 2 in order to achieve Level 3. Without completing one stage successfully before moving to the next, important issues are not entirely handled, and the result is likely to not be the best outcome for an organization [3].

The goal of the SA-CMM best practices model is to provide a description of the software acquisition process that can be adapted to any organisational context, and as such it is a very wide-ranging [4]. However, this characteristic makes the SA-CMM difficult to implement within an organisation, as specific requirements for this implementation in terms of tools or identified techniques are not complete. Another significant gap in the SA-CMM is that it does not specify requirements, which is considered to be essential for determination of the appropriate fit between software and organisation [5]. As such, although the SA-CMM is used in many organisational contexts it may not be appropriate or optimal for all organisations.

Table 1. SA-CMM model

Level	Focus	Key Process Areas
1. Initial	Competent People and Heroics	
2. Repeatable	Basic Project Management	- Transition to Support - Evaluation - Contract Tracking and Oversight - Project Management - Requirements Development and Management - Solicitation - Software Acquisition Planning
3. Defined	Process Standardization	- Training Program Management - Acquisition Risk Management - Contract Performance Management - Project Performance Management - User Requirements - Process Definition and Maintenance
4. Quantitative	Quantitative Management	- Quantitative Acquisitions Management - Quantitative Process Management
5. Optimizing	Continuous Process Improvement	- Acquisition Innovation Management - Continuous Process Improvement

2.2 CMMI-ACQ

The CMMI-ACQ best practices description (current release 1.02, September 2007) is the successor to the SA-CMM model (although the SA-CMM model is still in use in many organisations). [6] This model is built on the older SA-CMM description, but provides a considerable improvement over this model. It was generated from the CMMI Architecture and Framework, an existing model that describes various aspects of lifecycle development of software [7]. This model is also intended to be a generic model for organisational process description and improvement, applicable to any organisation [7].

Unlike the SA-CMM model, there are three levels of inclusion for CMMI model components, including required, expected, and informative components [7]. There are 22 identified process areas within the CMMI-ACQ model, and like SA-CMM, the CMMI-ACQ model can be used as a staged model; however, unlike the SA-CMM model, the CMMI-ACQ model can also be used in a con-

tinuous representation, which allows for transition between stages depending on existing capabilities [7]. What this means for an organization is that it does not have to feel that each stage must be completed, even if it does not entirely apply to the organization or its needs, before moving to the next. An organization has some control over the model in the ability to make changes or adjustments to specific issues or concerns in relation to how it operates and the needs that it considers to be important.

Under the staged model, in which, as with SA-CMM, transitions occur from one model to the next sequentially and determined by achieving core competencies from previous stages, there are five different stages that the organisation can move through (Initial, Managed, Defined, Quantitatively Managed, and Optimizing); these five stages roughly correspond to the five stages of the SA-CMM model [7]. However, the continuous model allows for a sixth stage, Incomplete (in effect a Level 0). This extra level in the model can be important for an organization because the assumption that a level has been completed is not possible. An organization can indeed be incomplete with regards to its acquisition efforts. The difference between these two models is that the standard presents a capability model in the continuous representation (which is intended to describe the individual capabilities of the organisation at whatever level they have occurred), while the staged representation is intended as a maturity model, representing the overall maturity level of the process within the organisation [7]. This means that the staged and continuous representations do have different approaches to tasks and requirements for competence attainment, but the models are largely consistent with each other.

While the CMMI-ACQ description did rectify some of the challenges of SA-CMM including development of a requirements determination activity area and introduction of specific predefined tasks, it does retain some challenges as well. These include lack of guidance offered on the priority of process areas within the continuous representation [8]. This lack of prioritization means it can be challenging for implementers of the process to determine appropriate priorities or task ordering; while decision support models have been identified that can rectify this challenge to some extent it remains one of the highest barriers to organisational implementation of this process. In addition, for an organization that may be beginning the process of truly thinking about best practices related to software acquisition or having a scientific theory to guide software acquisition, the use of a broad model that is more oriented toward ideas as opposed to specific tasks can be challenging. An organization can become so concerned about ideas and concepts than it ignores the tasks that need to be completed in order to successfully complete software acquisition and ensure that it meets the needs of the organization once it has been completed.

2.3 ISO/IEC 12207

The third standard that was undertaken during this analysis process was ISO/IEC 12207, which implements the software acquisition process as part of a full description of the software lifecycle. ISO/IEC 12207 was the first standard to describe the full software lifecycle [9]. The ISO/IEC 12207 family of standards is one of the most commonly used standards for the definition of the software lifecycle process as a whole, including the software acquisition process, which is embedded in the standard [10]. Starting from the acquisition process, the ISO/IEC 12207 standard describes the full software lifecycle, including aspects such as human resources management and infrastructure life cycle management. This means that specific tasks that need to be completed, as opposed to only focusing on issues and concepts, are part of the model. In essence, this model allows for a bridge to be created between purely issue-related software acquisition and the tasks and duties that are part of the software acquisition process [9]. However, only the software acquisition process model was considered to be relevant for this research process.

Unlike SA-CMM and CMMI-ACQ, the ISO/IEC 12207 model is not a capability or maturity model *per se*, but is instead a lifecycle model. Also unlike the previous two models discussed, it clearly defines operations, activities, tasks, and provides a complementary Supply Process that outlines the operations and activities required of the supplier of the software [9]. This view of the process from the supplier's viewpoint increases the potential that requirements of the organisation acquiring the software are met, because the Supply Process specifically addresses the requirement to meet purchaser requirements [9]. Also unlike the CMMI-ACQ and SA-CMM model, the ISO/IEC 12207 standard is not a staged model that is based on the stage of organisational maturity, but is instead a single process designed for use at all levels of maturity [9]. This does allow the organisation to build competence in these processes over time, but does not provide a means of determining the organisation's maturity. However, there are sub-processes offered that allow for identification of greater detail in the process if required. The model is much more detailed in this regard, which is important for an organization that may be undertaking software acquisition efforts in a scientific manner for the first time. The specific details that are listed provide a better guide for what needs to be performed in order to achieve a successful outcome.

Thus, while the ISO/IEC 12207 model is not appropriate for determining the maturity or capability level of an organisation it does allow for the development of specific skills and competencies related to software acquisition by spelling out the required process for effective software acquisition (and through the complementary Supply Pro-

cess, the effective supply of software to organisations). As such, this can be seen to be a standard that would be put to a different use from the organisational requirement that led to the use of the capability maturity models described by the SA-CMM or CMMI-ACQ frameworks.

3. SA-CMM Conversion Process

The description above provides an overview of the best practices frameworks. However, there is still the question of how these textual descriptions can be defined as a framework that can be directly compared to process models derived from organisational studies. There were a number of issues identified in this experience. First, the textual descriptions offered little information regarding the specific tasks and activities. Second, the varying maturity levels within this capability maturity model offer different activities and processes, making the model complex to describe as a single model framework. In order to overcome these challenges, the specific tasks and activities were kept generic in order to comply with the framework of the discussion, and the individual models were described in separate frameworks. It is understood that since the SA-CMM model is a cumulative framework, that organisations would be able to compare their observed processes serially against the models in order to determine which capability level met the requirements most effectively.

3.1 Description of the Framework

The SA-CMM framework presented is the Level 2 approach to software acquisition. However, a similar process was followed to describe the process at all five levels of the organisation. (This process was also performed for all levels within the CMMI-ACQ framework as well as the Acquisition process within ISO/IEC 12207). The process was as follows:

1) Identify key process areas that were described within the model;

2) Create a definition of the key process area that textually described the inputs, process, and outcomes of the process area;

3) Identify inputs;

4) Identify outputs;

5) Identify people;

6) Identify cost.

The key process areas identified at Level 2 (Repeatable) included Software Acquisition Planning, Solicitation, Requirements Development, Project Management, Contract Tracking and Oversight, Evaluation, and Transition to Support [3]. As can be seen in **Figure 1**, the key process areas are a mixture of competencies and process areas, including technical, project planning and management, and legal aspects of the acquisition process, which indicates that the process will engage different individuals and organisational competencies within the region.

The process flow indicated by the Level 2 (Repeatable) description with the SA-CMM model was exceptionally simple. This process was a linear process with little room for deviation from the existing model or translation from one end of the model to another.

The model above describes the process of software acquisition at maturity Level 2 within the SA-CMM. The steps involved in software acquisition in the model are standard in terms of key process areas, but actual tasks and activities vary depending on the organisation. The issues of cost and resource allocation are strongly dependent on individual implementation and are not determined in the standard; as such, they will need to be determined on observation of individual implementations of the standard. However, the key process areas must all be identified and effectively engaged in if the firm wishes to move beyond the Level 2 (Repeatable) level of implementation, just as it was necessary for all the key process areas at Level 1 to be met effectively in order to move to Level 2 [7]. Because of this, a maturing software organisation will move into an effective implementation of all identified key process areas before moving forward to the Level 3 (Defined) model. As discussed above, there is a considerable challenge within this model, as tasks and activities are not actually clearly defined and there is no way to identify a generic standard for tasks and activities; as such, there will be representative tasks and activities identified in order to attempt to create a framework that can be used to describe a generic situation that meets the demands of a Level 2 organisation.

3.2 Identification of Tasks and Specific Activities

The model above has an obvious weakness, in that it does not describe specific tasks and activities, but instead focuses on the identification of process and management related functions. Once again, the disconnect that seems to exist between purely theoretical concerns as opposed to concerns related to real-world software acquisition and implementation are noticeable. In practice, the organisation will need to implement the best practices framework with actual tasks and activities, which may be refined from organisational needs or may be identified from additional best practices frameworks. The best practices frameworks that have been identified as ideal for use in construction of the framework models include the Information Technology Infrastructure Library (ITIL) best practice guide, published by Great Britain's Office of Government Commerce, and the Control Objectives for Information and Related Technologies (CoBiT) framework, which provides IT governance best practices [11–13].

The ITIL framework, which is shown in **Figure 2**, is commonly used in conjunction with CoBiT to incorporate governance and best practices, and there is a specific guideline intended to facilitate this co-incorporation [11]. The applicable ITIL volume is the Software Asset Management volume, which addresses software management at all stages of the lifecycle [14]. These additional best practices frameworks mimic the addition of supplemental frameworks, policies and standards within an actual organisation in order to determine appropriate tasks and activities. Additionally, researchers have identified these frameworks as being commonly implemented within the organisational environment, meaning that it is likely that an organisational study will reveal a similar process of identification of actual tasks and activities that would take place. As such, this is considered to be an appropriate supplement for the structure demonstrated above.

The combining of the ITIL and CoBiT models can actually be performed with relative ease. **Figure 3** provides a model for combining ITIL and CoBiT. Combining the models overcomes the problem of the lack of specific tasks and activities that is found in the SA-CMM Level 2 process flow. The combined ITIL and CoBiT model brings together management functions and company activities. In essence, what has occurred in the combined model is that software acquisition as moved from being purely managerial in nature to something that involves employees at all levels of an organization.

4. Refinement of the Framework

This report describes only the first iteration of a process that is expected to have several stages of refinement. One potential way in which the identified models can be refined and further clarified is the use of simulation to identify potential difficulties and challenges within the framework. Business process simulation is commonly used in organisational environments for such tools as business process re-engineering and new process implementation, because it allows for identification of flaws within the proposed process Model [15]. The use of simulation within this context allowed for the identification of areas that could be problematic if implemented in actual practice. These areas will then be analysed in order to determine whether this is a specification error or whether it represents an actual area of implementation difficulty or inefficiency within the best practices framework. In the first case, the model will be refined to account for the identified difficulty, while in the second case further research will be performed with this area of weakness as a focal point.

The identified simulation and modelling tool that will be used for this process is YAWL (Yet Another Workflow Language), which is an open source workflow modelling and simulation tool [16]. This tool has been used extensively in academic BPM research as it is extensible and has a greater level of flexibility than most commercial offerings, which are primarily intended for analysis and are not aimed toward researchers. This process will require translation from the current BPML to YAWL's Petri nets structure, but the use of simulation in order to identify potential difficulties in the framework models that will be used for comparison of actual cases will provide signifi-

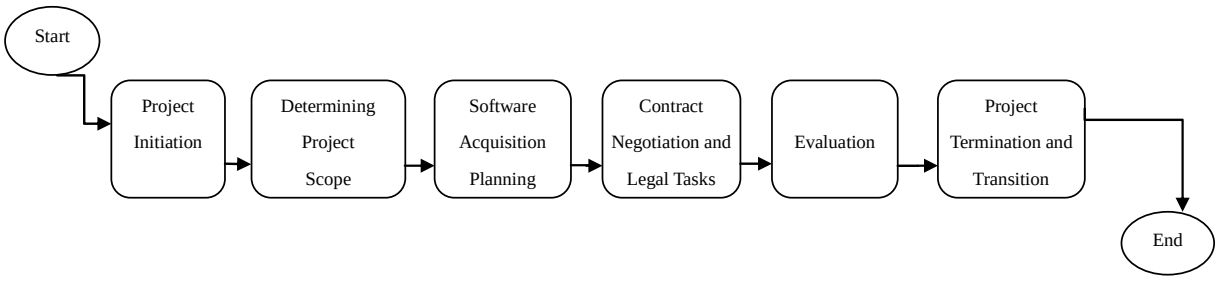


Figure 1. SA-CMM Level 2 process flow

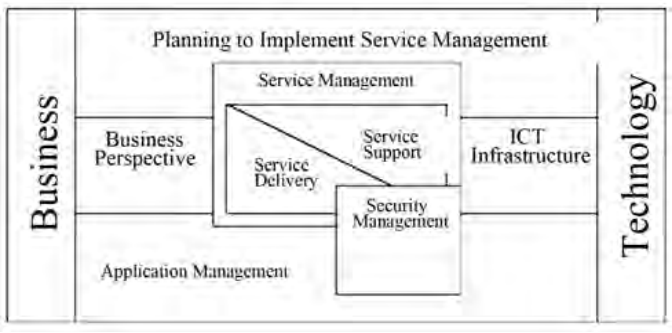


Figure 2. ITIL framework

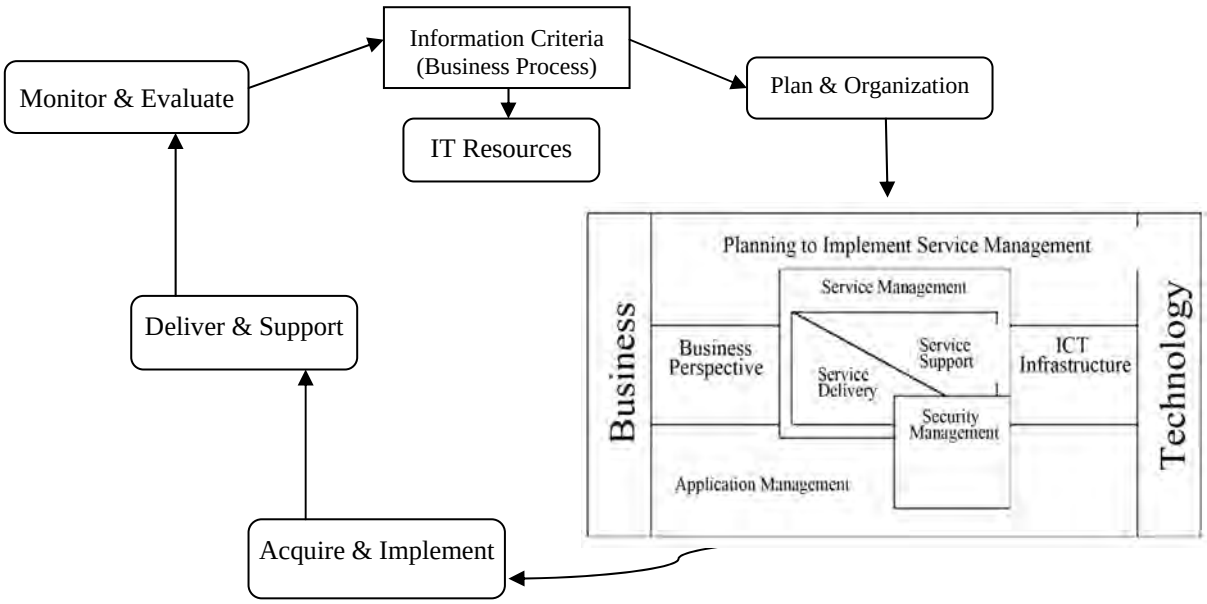


Figure 3. Combined ITIL/CoBiT model

cant benefits to the current research and as such this is considered to be an acceptable requirement.

5. Conclusions

The use of best practices frameworks and standards is common within organisations, but the specific needs of the organisation drives the choice of framework or standard. The three most common best practices standards provide different advantages and disadvantages to the organisation, and can be used in different ways to improve the organisation. The organisation that wants to engage in a specific process that is consistent across maturity levels and stages and includes specific tasks is likely to choose the ISO/IEC 12207 standard, while an organisation intent on developing capabilities in software acquisition will choose the CMMI-ACQ model. An organisation intending to develop maturity in the software process could use either CMMI-ACQ or the SA-CMM model. This discussion has demonstrated the process by which the textual descriptions were converted to idealized process frameworks that could be used to actualize process models identified from organisational studies. However, some issues have remained unresolved form this experiment, including the ability to identify specific activities and structures. This is not especially a problem with the ISO/IEC 12207 standard, which spells out specific organisational activities and a specific process, but does remain a challenge with SA-CMM and CMMI-ACQ, which are more flexible in terms of identification of the process activities and requirements (and in fact in some cases do not have specific requirements in this regard at all). In this discussion, it was suggested that the implementation of standards including ITIL and CoBiT could be used to identify the specific tasks and processes that are missing from these structures. However, the use of organisational studies will be required in order to determine how the organisations themselves have resolved this issue —have these organisations used these best practices IT governance frameworks, or have they merged software capability and maturity models with uniquely identified models or practices? This is one of the outstanding issues that the researchers hope to resolve through the current research process.

It should be noted that the majority of organisations that use the SA-CMM framework for description of software acquisition processes currently operate at Level 2, which was the model described within this framework [17]. Thus, the description of the Level 2 framework was engaged in first in order to be able to describe the widest potential organisational pool. However, this process was followed for other frameworks and organisational levels as well. The researchers hope to use this generic description as a means of creating a template on which organisations can be matched following the identification of actual processes within the organisation in order to provide identification of processes from the research perspective, even in organisations that do not use the frameworks or models explicitly. This could yield information both for implementation of the models within other contexts (for example, providing information regarding optimizations in generic frameworks that can be identified from actual organisational studies) as well as provide information for organisations in terms of process improvement and increasing efficiency. Thus, the identification not only of key process areas, but also association of tasks and activities as described by ITIL and CoBiT to these key

process areas, is expected to be key to the eventual utility of this research.

REFERENCES

[1] M. Biro, C. Deak, J. Ivanyos, and R. Messnarz, "From compliance to business success: Improving outsourcing service controls by adopting external regulatory requirements," Software Process Improvement and Practice, Vol. 11, pp. 239–249, 2006.

[2] L. Anderson, M. Fisher, and J. Gross, "Case study: IRS business system modernization process improvement," Carnegie Mellon University, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2004.

[3] J. Cooper and M. Fisher, "Software Acquisition Capability Maturity Model (SA-CMM) Version 1.03," Technical Report, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, USA, 2002.

[4] J. White, "Managing information in the public sector," M. E. Sharpe, London, UK, 2007.

[5] J. A. Mykkanen, M. P. Tuomainen, "An evaluation and selection framework for interoperability standards," Information and Software Technology, Vol. 50, pp. 176–197, 2008.

[6] F. Navarrete, P. Botella, X. Franch, "Reconciling agility and discipline in COTS selection processes," Proceedings of the 6th International IEEE Conference on Commercial-off-the-Shelf [COTS]-Based Software Systems, pp. 1–11. IEEE, 2007.

[7] "CMMI product team: CMMI for acquisition, version 1.2: CMMI-ACQ, version 1.2," Technical Report, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, USA, 2007.

[8] S. J. Huang, W. M. Han, "Selection priority of process areas based on CMMI continuous representation," Information and Management, Vol. 43, pp. 297–307, 2006.

[9] J. Moore, T. Doran, A. Kark, "Systems and software engineering—software lifecycle processes," Software & Systems Engineering Standards Committee of the IEEE Computer Society, Institute of Electrical and Electronic Engineers, 2rd Edition, Piscataway, 2008.

[10] Y. Hwang, J. G. Park, "Approaches and requirements to develop and improve the standard processes for a research and development organisation," Systems Engineering, Vol. 9, No. 1, pp. 35–44, 2006.

[11] D. Nichols, "Governing ITIL with COBIT, DITY Newsletter," itSM Solutions LLC, Lexington, USA, 2007.

[12] "ITGI/ISACA: COBIT 4.1," IT Governance Institute, USA, 2007.

[13] "ITGI/OCG: Aligning COBIT, ITIL, and ISO 1799 for business benefit: Management summary," Office of Government Commerce, Norfolk, 2007.

[14] "Office of government commerce: Software asset management," Stationery Office, London, 2003.

[15] I. Lee, "Selected readings on information technology and business systems management," Idea Group Inc, London, 2008.

[16] "YAWL: Yet another workflow language," http://www.yawl-system.com/.

[17] C. Meyers and P. Oberndorf, "Managing software acquisition: Open systems and COTS products," Addison-Wesley, Sydney, 2001.

J. Software Engineering & Applications, 2010, 3: 287-302
doi:10.4236/jsea.2010.33035 Published Online March 2010 (http://www.SciRP.org/journal/jsea)

A Novel Training System of Lathe Works on Virtual Operating Platform

Hui-Chin Chang

Department of Mechanical Engineering, De Lin Institute of Technology, Taipei, Taiwan, China.
Email: chang.hcjang@gmail.com

Received October 19th, 2009; revised November 11th, 2009; accepted November 20th, 2009.

ABSTRACT

In recent years virtual reality technology has been extensively applied to the areas relating to manufacturing, such as factory layout planning, manufacturing planning, operation training, system testing, and process control, etc. Most of the studies made in the past focused on the simulation and monitoring of the entire manufacturing system, or the simulation of working schedule implementation. There was no complete research result on the most basic processing unit of manufacturing system—the operation training for the lathe works. However, these skills of operating methods are the basic skills and particularly emphasized in the practical operation during instruction. As observed from the past experience, after workers had learned the operating process of lathe works, they could achieve very good results in the written examination of the basic knowledge about the operation of different works. However, when they faced the actual operation in front of machine, they were always at a loss. The reason behind this was that when the workers had to face the possible collision and damage during actual operation of machine, since they did not have performed many times of simulated computer rehearsal designed for them to get familiar with the entire operating process, fear and nervous psychology were naturally derived from them. In view of this, the paper uses EON Studio software to integrate virtual reality technology with the application of 3D solid model to simulate a virtual operation of the various operating steps and virtual machining of lathe works during practical operation of lathe machine. The simulation enables users to learn in the simulated environment without scruple. After the accumulation of learning experience, it can be applied in the actual environment to accomplish the mission of operation.

Keywords*: *Virtual Reality, Lathe Machine

1. Introduction

Computer graphics technology was originally developed from the traditional 2D cartographic technology, which was then developed to be 2.5D, and then the 3D solid object and animation production. In recent years, virtual reality (VR) technology gradually becomes mature. Man and computer have been brought to a communication interface of "going into the environment". The scenes appeared in computer are no longer the single stiff images, but the continuous, vivid and animated images. VR is an operation environment composed of "intelligent objects" with different particular attributes. Currently, VR has been extensively applied to medical science, education, military, entertainment, engineering, machines, marketing, etc.

The so-called "virtual reality" (VR) technology mainly uses computer to simulate a real or virtual environment, enabling users to have a feeling of being in the environment. The environment not only gives a three-dimensional and layered look, but also lets users learn in the simulated environment. After they have accumulated their learning experience, they can apply it in the real environment to accomplish their missions.

Presently, VR technology has been adopted in many areas. For example, in the area of medical science, M. Tavakoli *et al*. [1] and Chen E. *et al*. [2] used haptic interface for the computer-integrated endoscopic surgery system. Through force feedback device, user could interact with the virtual scene in computer to perform more effective training of surgical and medical operation. The intravenous injection simulation system of Shoaw [3] concretely provided a training course that met the requirements for the learning of intravenous injection technique by nursing students. The system decreased the happening of accidents and the sliding of syringe during intravenous injection, and raised the quality of nursing and clinical services. The palpation simulation system of

M. Dinsmore *et al*. [4] was applied by doctors to the simulation of looking for tumors from patients. It could train doctors to diagnose the tumor of subcutaneous tissue accurately through palpation by fingertips.

As to the engineering and mechanical domain, Korves and Loftus [5] and Sly [6] imported VR technology to the outlay and planning of manufacturing system for factories. The system could more intuitively and efficiently use the computer digitalized virtual prototype. Through the simulated prototype, before the design of a product and the actual prototype or manufacturing system appears, the designer was able to experience and feel the performance of the future product or the status of the manufacturing system. Immediately, the designer could discover the mistakes and defects that were not considered in the process of product design, and then make amendments accordingly. Hence, more perspective decisions could be made, and more excellent implementation projects could be implemented to guarantee the quality and quantity of product.

Dewar *et al*. [7] indicated that in the process of product assembly, since it always involved such problems as product design and assembly, it had to rely on professional knowledge and the actual assembly by experienced experts so as to formulate standard assembly procedures. Therefore, in order to decrease effectively the time spent on the assembly procedures of product, VR technology is up to now a technology most frequently applied. As to the studies in this aspect, they mainly included two directions: one was undergone directly by using VR technology to assist assembly training [8–10]; and the other was the application of VR, together with the related technologies like CAD/CAM, etc., to preserve many significant concepts, procedures or experience in the assembly process by digitalized (visualized) or formalized ways (steps) [11–13]. There was one thing worthy of mentioning that the visual assembly design environment (VADE) system developed by Jayaram *et al*. [14] emphasized the integration of VADE and CAD system. On the one hand, VADE information was acquired from CAD system; and on the other, the design performed by VADE or assembly information could be transmitted back to CAD system for further application. At the same time, after VADE system was added with detection of collision and simulation of physical nature, the application area of VR technology was tremendously enlarged.

Regarding VR instruction and training, the related application and research are very extensive. There were studies on the benefits of haptic feedback in virtual reality environment in terms of the shortening of completion time and the improvement of perceptual motor capabilities of human operator [15,16]. Wu Y. L. *et al*. [17] established a virtual network laboratory. Students can do physical experiment in the virtual laboratory on the internet. The laboratory can protect students from encountering possible danger when doing physical experiments. Eder Arroyo *et al*. [18] established a virtual control and operation system of apparatuses, providing staff with the training on control and operation procedures of the apparatuses and equipments in factory, and assisting staff in becoming competent for their jobs within a short period of time. Lei Li *et al*. [19] proposed an immersive virtual reality system called ERT-VR, in which the instructors assigned a specific training scenario to the trainees by using the scenario creator. Trainees took on the role of the characters in the training scenario, and controlled their actions and ultimately the scenario outcome.

Lathe machine is one of the working machines with the widest range of usage in machinery factories, and lathe works are also the basic skill for their workers. Lathe works use cutter to machining raw material to form the shapes of facing, external turning, internal turning, knurling, grooving, drilling, taper turning, contour turning, threading, etc. Therefore, this paper firstly uses VR application software to integrate the 3D models of lathe machines. After that, through step-by-step use of function nodes provided by EON studio software, the operation features for the driving function of each operating handwheel of lathe machine are constructed. At the same time through the connection with external program, it is available to simulate the virtual machining actions. In this way, users are able to learn the operation of lathe machine in the simulated environment. After they have accumulated their learning experience, they can apply it in the real environment, thus decreasing the damage of mechanical operation, and saving the time for education and training.

2. Development of Virtual Lathe Machine

This paper mainly applies the combination of VR technology and 3D solid model to simulate the virtual operation of the practical operating process of lathe machine, and virtual machining operation of the lathe works. The simulation reduces the worker' fear aroused when facing the large-sized machine in the learning process of lathe works. It not only improves the learning effects of workers, but also decreases the damage of machine caused by the wrong operation of workers for their unfamiliarity or nervousness. The main task of this section includes:

1) Construction of 3D models of virtual lathe machine;

2) Development of the virtual operation platform of lathe machine.

2.1 Construction of 3D Models of Virtual Lathe Machine

The paper adopts Pro/Engineer software, which has the characteristics of 3D solid model, single database, feature-based design and parametric design, as the operation tool for establishing the 3D model of the various parts of lathe machine and for setting the relationship among their mutual positions. Therefore, besides the main fixed bed

the 3D model required to be constructed still needs to build up the components of transmitting mechanism, such as the axle, variable-speed mechanism, auto-feed mechanism, machining feed bench, tool post, power clutch, and tailstock assembly and dead center, etc. As shown in **Figure 1**.

2.2 Principles for Establishment of Virtual Operation Platform of Lathe Machine

To the entire lathe machine, the overall operation functions are: 1) Operation for the action of tool change of tool post; 2) Automatic and manual operation for longitudinal/transverse feed control; 3) Control of CW and CCW rotation of axle, operation of braking, and simulation of its axle rotation inertia. Here using (1) Operation for the action of tool changes of tool post function as example to explain its construction process.

2.2.1 Principles for Action Control of Tool Change of Tool Post

1) Requirements of system

- When carriage clamping lever is locked, there is a position limit of locking;
- Only when carriage clamping lever is at loosening position, the tool change of tool post can be implemented.
- Tool post should be able to rotate in CW and CCW direction, and there is no position limit.

The tool change of tool post mechanism is shown in **Figure 2**.

2) Principles of construction

a) Initial condition is carriage clamping lever to be in sensing status, and transmit its orientation into orientation judgment program;

b) When carriage clamping lever is active (mouse left button or mouse right button be clicked), then carriage clamping lever rotate in CW or CCW direction, and transmit action signal into orientation judgment program;

c) To judge whether carriage clamping lever is situated at the position of locked limit by means of orientation judgment program;

d) If carriage clamping lever has been at the locked position, it is required to disable the sensing of both the right button of carriage clamping lever and tool post rotation;

e) If carriage clamping lever has been deviated from the locked position, it is required to active the sensing of both the right button of carriage clamping lever and tool post rotation.

The flow chart of tool change of tool post control as shown in **Figure 3**.

3) Software construction techniques

a) Use the right (left) button sensing of "ClickSensor" function node (symbol of "tool post open" ("tool post lock") icon depicted in **Figure 4**) to detect whether the mouse right (left) button be clicked on the object of carriage clamping lever or not;

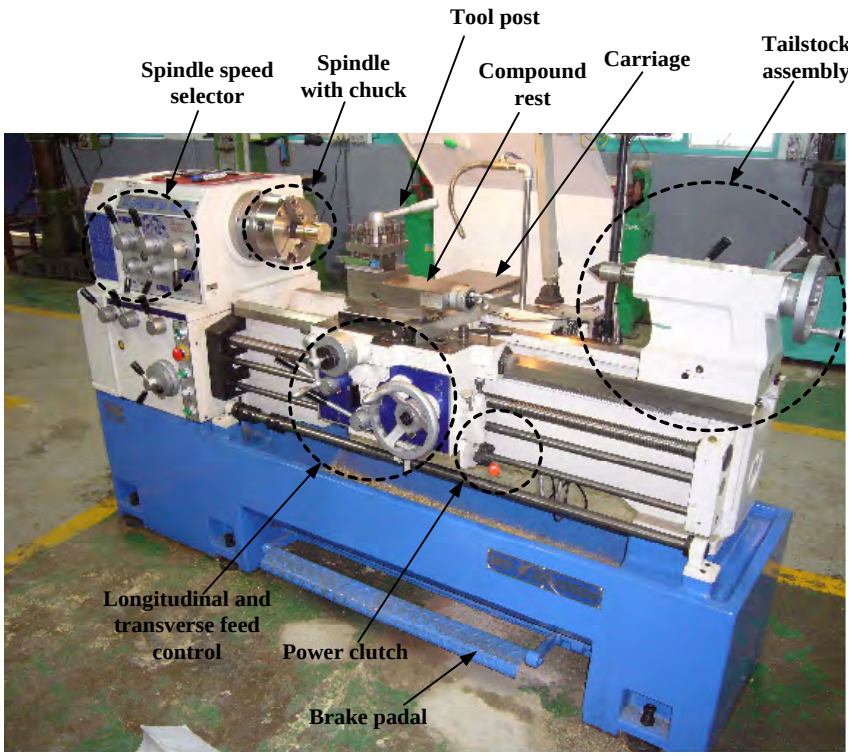


Figure 1. Lathe machine (WH 1000G-Win Ho Technology Industrial Co., Ltd. Manufacturing)

b) Use the "Place" function node (symbol of "tool post open act" ("tool post lock act") icon depicted in **Figure 4**) to control the object of carriage clamping lever action. *i.e.* when the mouse right (left) button be clicked on the object of carriage clamping lever, the "ClickSensor" function node will receive the "OnButtonDownTrue" signal, and then it will send out the "SetRun" signal (as the **Table 1** listed) to drive the object of carriage clamping lever action;

c) Use the "Frame" function node (symbol of "tool post handle-1" icon depicted in **Figure 4**) to transmit the object of carriage clamping lever's orientation ("World Orientation" as the **Table 1** listed) into orientation judgment program (symbol of "Script" icon depicted in **Figure 4**);

d) To judge whether carriage clamping lever is situated at the position of locked limit by means of orientation judgment program (as the **Figure 5** depicted);

e) If carriage clamping lever has been at the locked position, the orientation judgment program will send out the "statussign" and "statussign1" (as the **Table 1** listed) signals to "Place" function node (symbol of "tool post left turn" and "tool post right turn" icons depicted in **Figure 4**). At this time, the value of "statussign" and "statussign1" signals both are "0", in other words, it will disable the action of "Place" function node, and then the object of tool post can not rotate;

f) In contrast, if carriage clamping lever has been at the loosened position, the orientation judgment program will send out the signal value "1" to the "Place" function node, that is say, it will active the "Place" function node, and then the object of tool post can rotate in CW and CCW direction.

The control process and script function of orientation judgment program of tool change of tool post as shown in **Figures 4–5**, and their interactive relationship between each function nodes are listed in **Table 1**.

2.2.2 Requirements for Automatic and Manual Operation of Longitudinal/Transverse Feed Control

Requirements of system:

- The longitudinal/transverse feed selection lever should possess the functions of upper/lower limit position, and central neutral position.
- When the longitudinal/transverse feed selection lever is situated at the central neutral position, it can be used for longitudinal and transverse manual feed operation.
- During the CCW (CW) rotation of axle, and when the longitudinal/transverse feed selection lever is situated

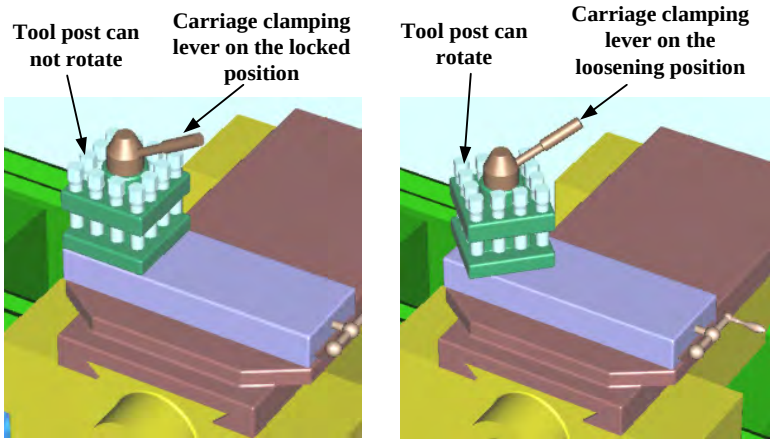


Figure 2. Tool change of tool post mechanis

Table 1. The software control skill of tool change of tool post

Output node	Output signal	Receive node	Receive signal	comment
Tool post handle 「Frame」	WorldOrientation	Script 「Script」	posvalue	Recording and judging the carriage clamping lever orientation
Tool post left 「ClickSensor」	OnButtonDownTrue	Script 「Script」	Mousein1	
Tool post right 「ClickSensor」	OnButtonDownTrue	Script 「Script」	Mousein	
Script 「Script」	statussign	Tool post left turn 「Place」	SetRun	
Script 「Script」	Statussign1	Tool post right turn 「Place」	SetRun	

at the lower limit position, then after the automatic feed control lever is pulled down and geared, the apron shall implement automatic feed movement towards (staying away from) the chuck direction.

- During the CCW (CW) rotation of axle, and when the longitudinal/transverse feed selection lever is situated at the upper limit position, then after the automatic feed control lever is pulled down and geared, the carriage shall implement automatic feed movement staying away from (towards) the worker direction.
- When implementing longitudinal or transverse automatic feed process, if the automatic feed control lever is pulled back to manual position, the apron or carriage shall immediately stop the feeding movement.

The longitudinal and transverse feed control mechanism is shown in **Figure 6**.

2.2.3 Principles of Control for CW and CCW Rotation and Inertia Action of Axle

Requirements of system:

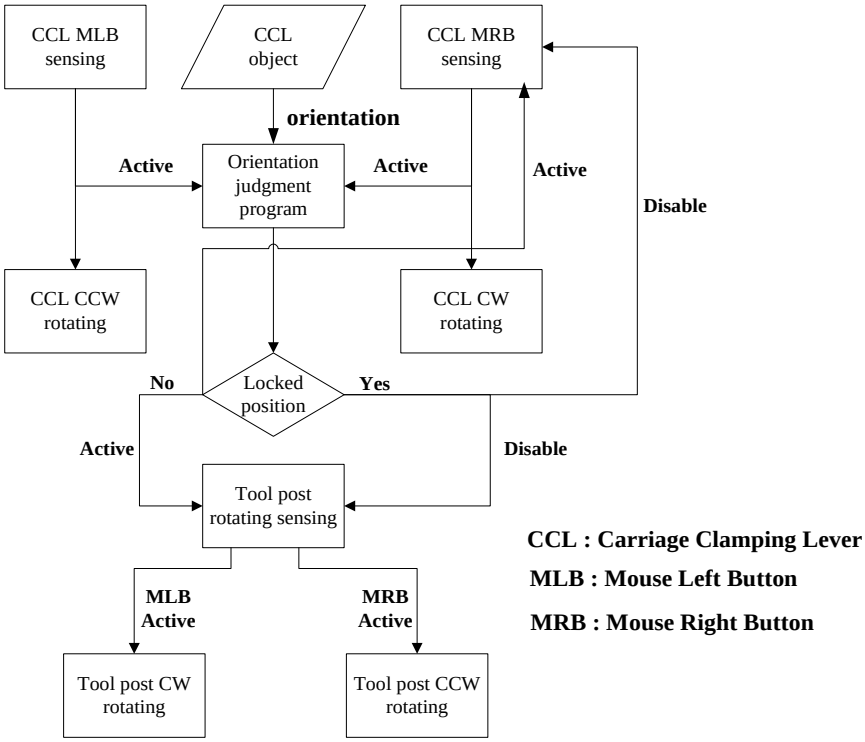


Figure 3. Flow chart of tool change of tool post control

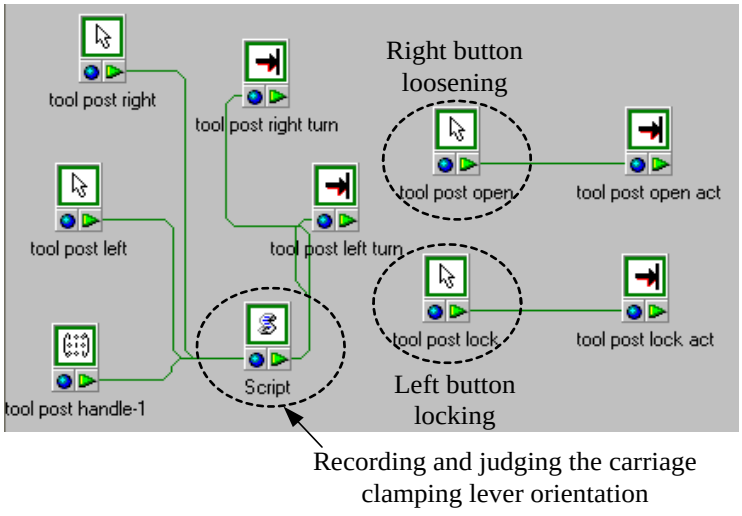


Figure 4. Control process of tool change of tool post

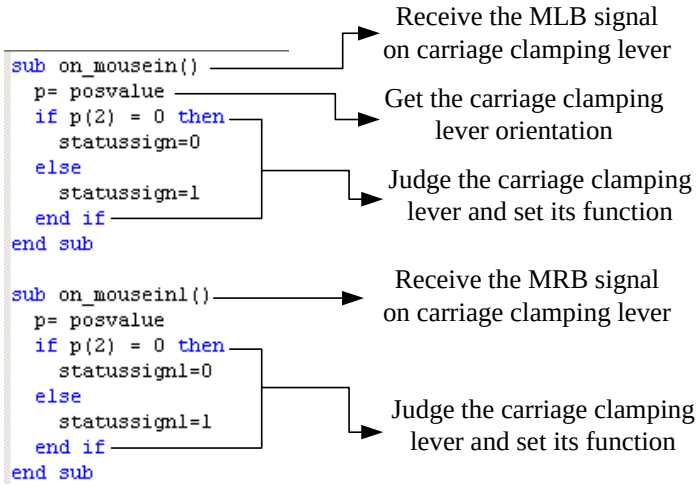


Figure 5. Script function of orientation judgment program

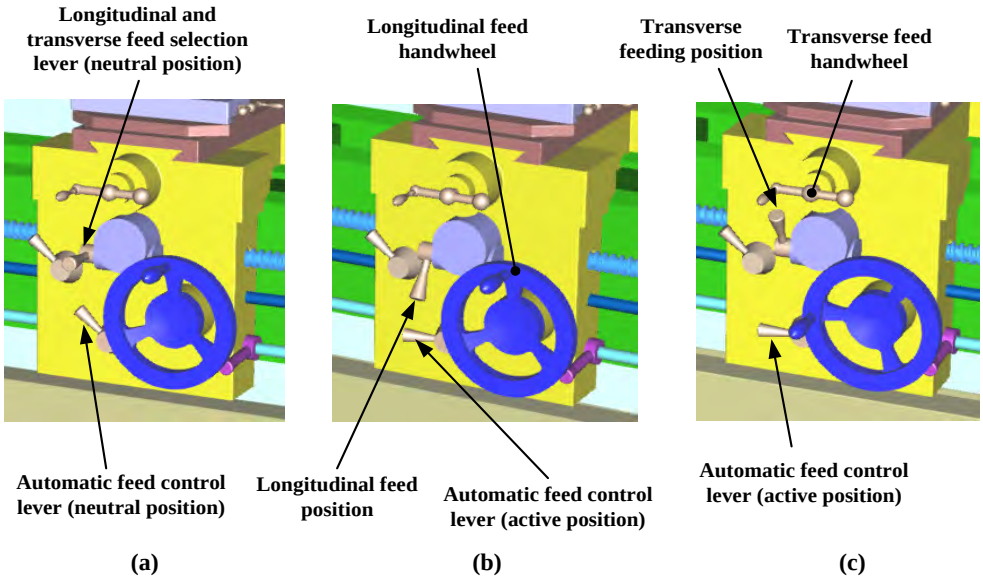


Figure 6. Longitudinal and transverse feed control mechanism

- Power clutch has to possess the function of being at upper/lower limit position;
- When power clutch is at the upper (lower) limit position, the axle rotates in CW (CCW) direction at specified speed;

After the axle is turned on, when power clutch resumes to the neutral position, the axle gradually stops rotating at a constantly decreasing speed;

- When the brake pedal is stepped down, the axle has to be able to stop rotating immediately. Meanwhile, the brake pedal should be able to resume to the original position actively.

The axle rotating control with power clutch mechanism is shown in **Figure 7**.

3. Principles for Establishment of Virtual Machining Platform

To the entire lathe works, the overall operation functions are:

1) External straight turning;
2) Internal straight turning;
3) Facing;
4) Necking;
5) External threading;
6) Taper turning.

3.1 Simulated Theorem of External Straight Turning

We utilize an assembly operation of outer-annular hollow

part and solid mandrel workpiece to simulate unmachined raw material and at the same time, we adjust the scaled rotation origin "O" of outer-annular hollow part to its left-sided shaft center shown in **Figure 8**. While simulating cutting operation, if the collision occurs between the cutter and the outer-annular hollow part, it triggers the outer-annular hollow part to induce scaled operation and the scale proportion, *i.e.*, $SCALE_{external}$ along the X direction can be expressed as the follows.

$$SCALE_{external} = \frac{L - tv}{L - (t-1)v} \quad t = 1, 2, 3 \qquad (1)$$

L denotes the original length of outer-annular hollow part, v is the cutting speed along the X direction, and t is the time to be a unit of second.

3.2 Simulated Theorem of Internal Straight Turning

We utilize an assembly operation of outer-annular hollow part and solid mandrel workpiece to simulate unmachined raw material and at the same time, we adjust the scaled rotation origin "O" of outer-annular hollow part to itsleft-sided shaft center shown in **Figure 9**. While simulating cutting operation, if the collision occurs between the

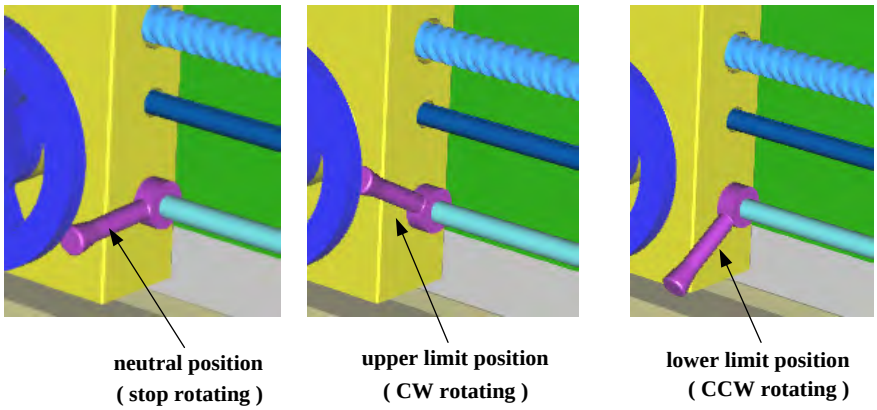


Figure 7. Axle rotating control with power clutch mechanism

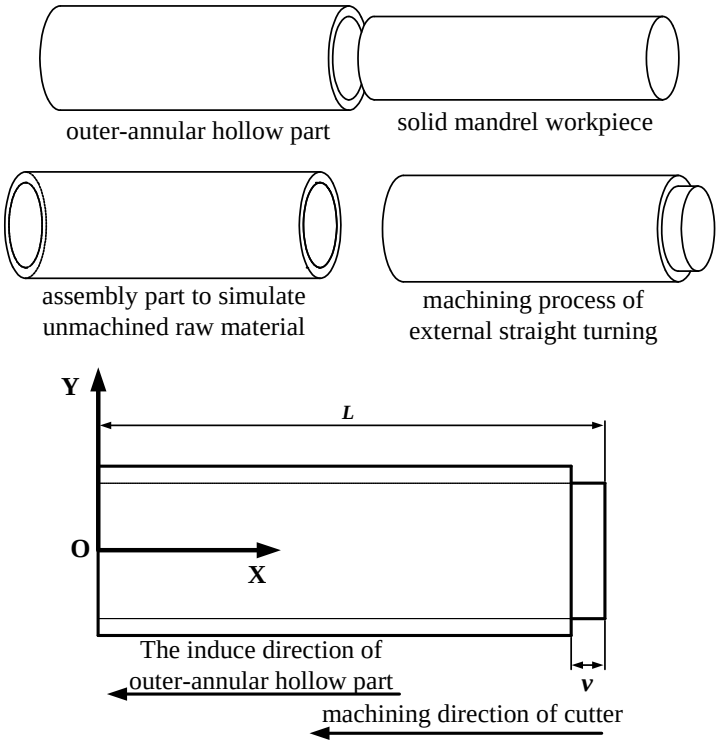


Figure 8. Simulated theorem of external straight turning

cutter and the solid mandrel workpiece, it triggers the solid mandrel workpiece to induce scaled operation and the scale proportion, *i.e.*, $SCALE_{inner}$ along the X direction can be expressed as the follows.

$$SCALE_{inner} = \frac{L - tv}{L - (t-1)v} \qquad t = 1,\ 2,\ 3 \qquad (2)$$

L denotes the original length of outer-annular hollow part, v is the cutting speed along the X direction, and t is the time to be a unit of second.

3.3 Simulated Theorem of Facing

We utilize an assembly operation of finished part and solid cutting-ring workpiece to simulate unmachined raw material and at the same time, we adjust the scaled rotation origin "O" of solid cutting-ring workpiece to its left-sided shaft center shown in **Figure 10**. While simulating cutting operation, if the collision occurs between the cutter and solid cutting-ring workpiece, it triggers the solid cutting-ring workpiece to induce scaled operation and the scale proportion, *i.e.*, $SCALE_{face}$ along the YZ plane di- rection can be expressed as the follows.

$$SCALE_{face} = \frac{D - 2tv}{D[1 + 2(t-1)v]} \qquad t = 1, 2, 3 \qquad (3)$$

D denotes the original diameter of solid cutting-ring workpiece, v is the cutting speed along the Z direction, and t is the time to be a unit of second.

3.4 Simulated Theorem of Necking

As the facing simulation, we also utilize an assembly operation of finished part and solid cutting-ring workpiece to simulate unmachined raw material and at the same time, we adjust the scaled rotation origin "O" of solid cutting-ring workpiece to its left-sided shaft center shown in **Figure 11**. While simulating cutting operation, if the collision occurs between the cutter and solid cutting-ring workpiece, it triggers the solid cutting-ring workpiece to induce scaled operation and the scale proportion, *i.e.*, $SCALE_{groove}$ along the YZ plane direction can be expressed as the follows.

$$SCALE_{groove} = \frac{D - 2tv}{D[1 + 2(t-1)v]} \qquad t = 1, 2, 3 \qquad (4)$$

D denotes the original diameter of solid cutting-ring workpiece, v is the cutting speed along the Z direction, and t is the time to be a unit of second.

3.5 Simulated Theorem of External Threading

As the external straight turning simulation, we also utilize an assembly operation of cutting workpiece and finish

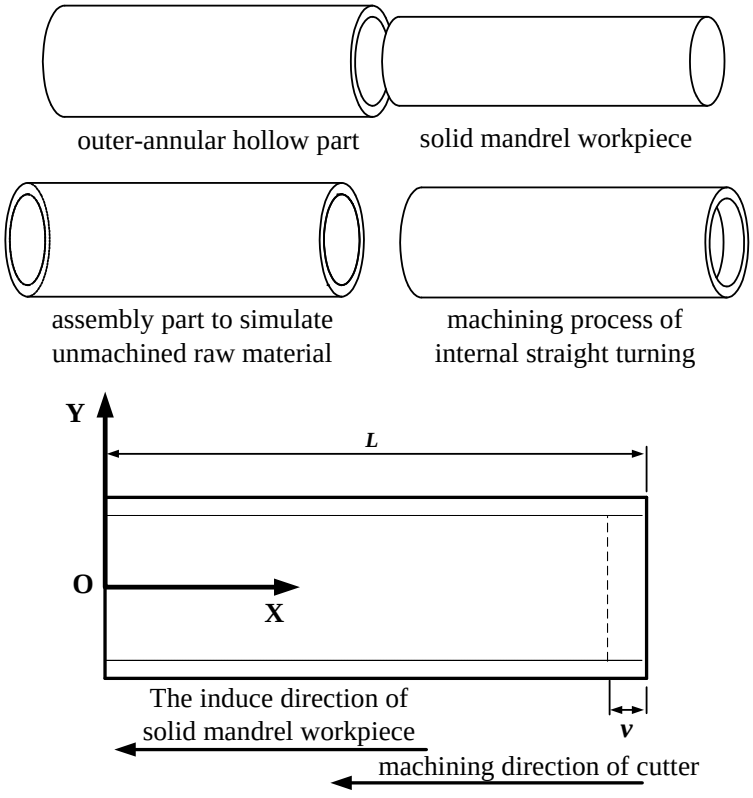


Figure 9. Simulated theorem of internal straight turning

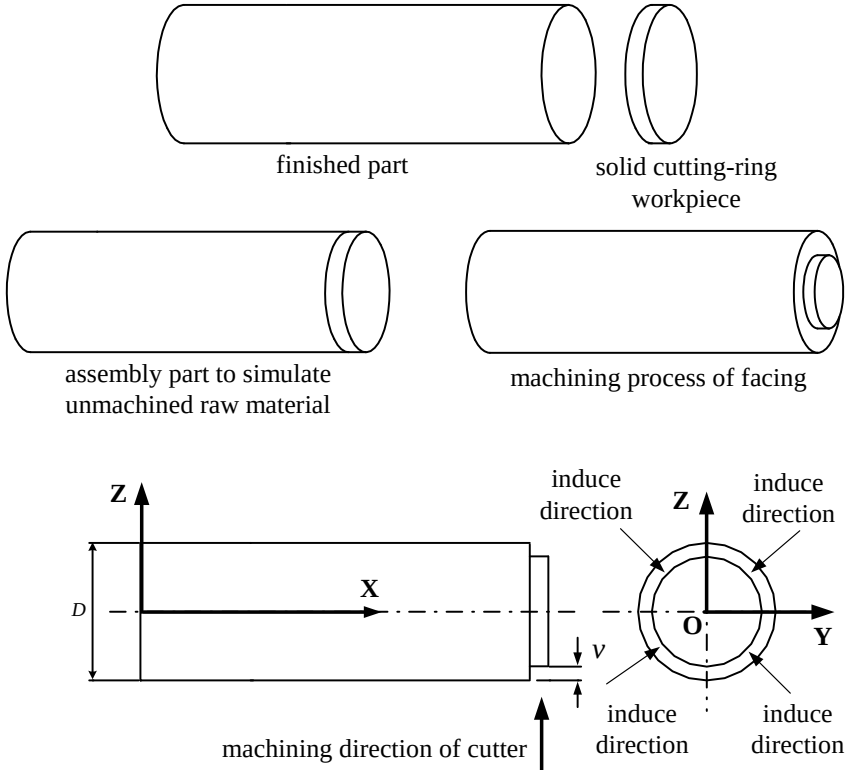


Figure 10. Simulated theorem of facing

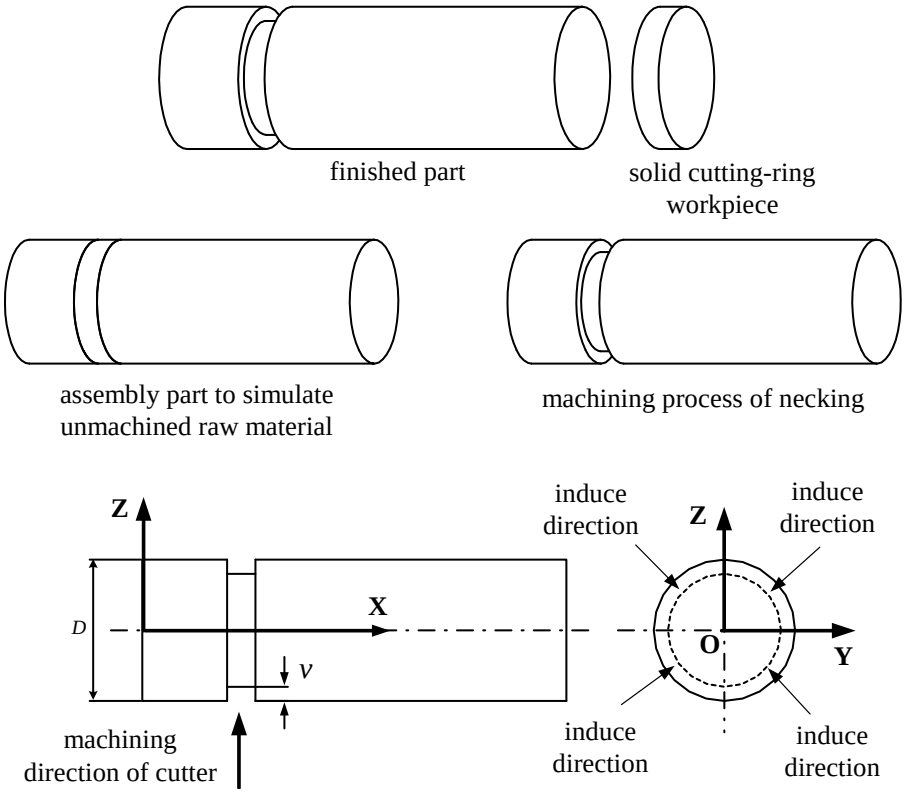


Figure 11. Simulated theorem of necking

part to simulate unmachined raw material and at the same time, we adjust the scaled rotation origin "O" of cutting workpiece to its left-sided shaft center shown in **Figure 12**. While simulating cutting operation, if the collision occurs between the cutter and the cutting workpiece, it triggers the cutting workpiece to induce scaled operation and the scale proportion, *i.e.*, $SCALE_{thread}$ along the X direction can be expressed as the follows.

$$SCALE_{thread} = \frac{L - tv}{L - (t-1)v} \quad t = 1, 2, 3 \qquad (5)$$

L denotes the original length of outer-annular hollow part, v is the cutting speed along the X direction, and t is the time to be a unit of second.

3.6 Simulated Theorem of External Taper Turning

As the external straight turning simulation, we also utilize an assembly operation of outer-annular hollow part and-finish part to simulate unmachined raw material and at the same time, we adjust the scaled rotation origin "O" of outer-annular hollow part to its left-sided shaft center shown in **Figure 13**. While simulating cutting operation, if the collision occurs between the cutter and the outer-annular hollow part, it triggers the outer-annular hollow part to induce scaled operation and the scale proportion, *i.e.*, $SCALE_{taper}$ along the X direction can be expressed as the follows.

$$SCALE_{taper} = \frac{L - tv\cos(\tan^{-1}(\frac{D-d}{2L}))}{L - (t-1)v\cos(\tan^{-1}(\frac{D-d}{2L}))} \quad t = 1, 2, 3 \qquad (6)$$

D denotes large diameter, d is small diameter, L is the length of taper, v is the cutting speed along the X direction, and t is the time to be a unit of second.

3.7 Constructed Technique of Virtual Machining System

This paper utilizes the reducible feature and mutual collision detection functions of function node, and then by means of programmable function to carry out the induced
proportion, to achieve the virtual machining simulation. Here uses the external straight turning as example to show the constructed technique of virtual cutting system.

1) Requirements of system

- When "S" key has been pushed down, the cutter shall implement automatic feed movement towards the part;

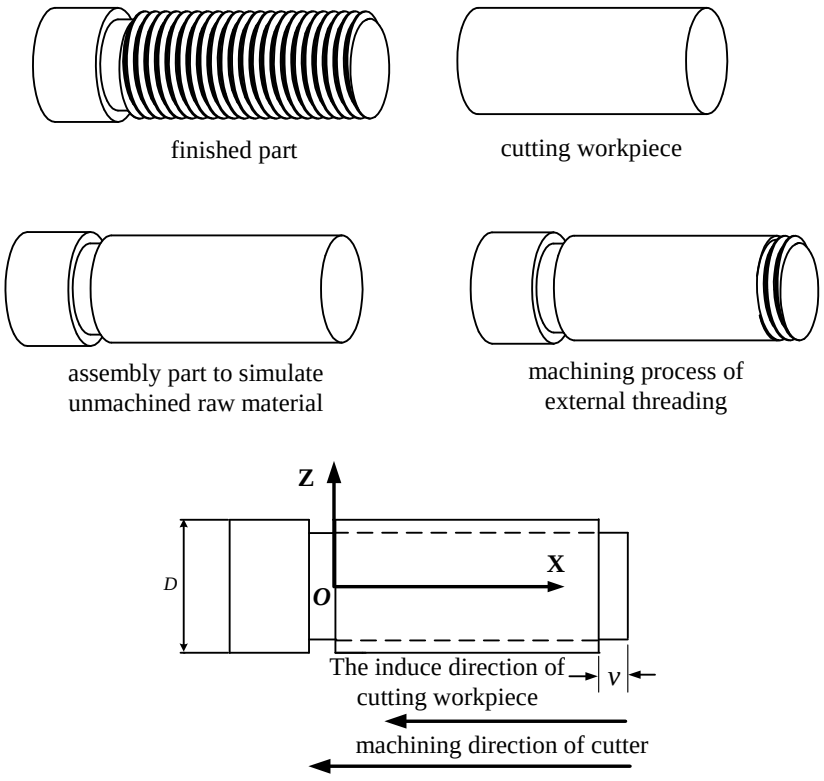


Figure 12. Simulated theorem of external threading

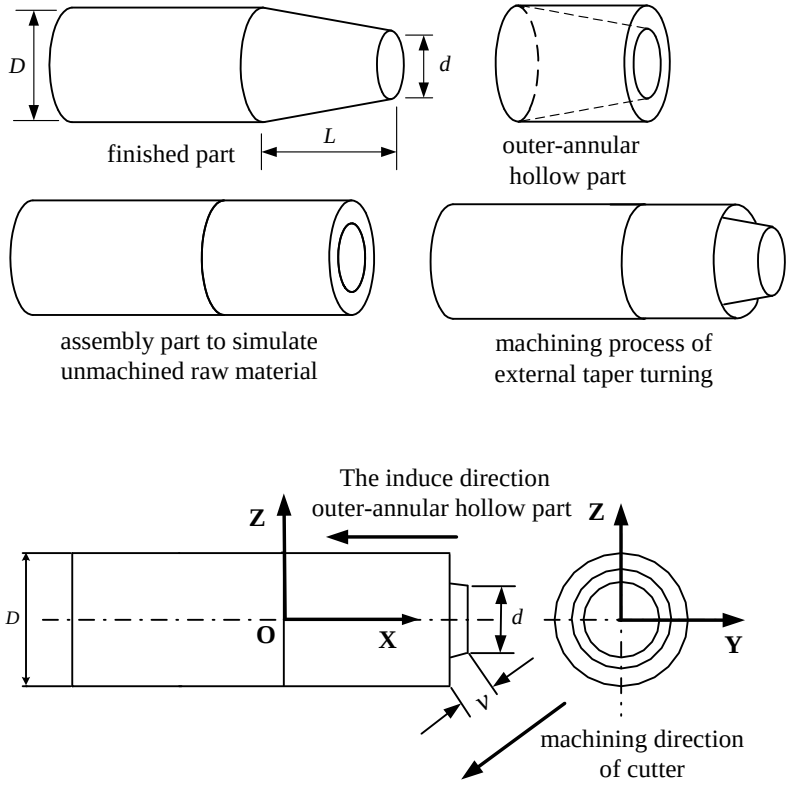


Figure 13. Simulated theorem of external taper turning

● When the cutter collided with the outer-annular hollow part, the length of outer-annular hollow part must be induced, and the induced speed is the same as cutter moving feed.

2) Constructed technique of software

- Use the "ClickSensor" function node to detect the condition that whether "S" key has been pushed down or not;
- Use the "Place" function node to control the cutter moving feed.
- Use "Script" function node to receive, record, judge and output the induced proportion of length of outer-annular hollow part, so as to control the induced speed is the same as cutter moving feed.

The model tree, routes, and parameters of script node are shown in **Figure 14**.

4. Practical Operation Cases

This paper makes a comparison between the situations before and after the operating process of the overall transmitting control function and machining function of lathe works, and takes it as an implementation example of virtual lathe machine operating and machining for lathe works.

Figure 15 shows the virtual lathe machine platform for this paper. **Figure 16** presents the results after the forward and backward movements of apron caused in the longitudinal feed operating process. **Figure 17** shows the results after the forward and backward movements of carriage caused in the transverse feed operating process. **Figure 18** shows the results after the forward and backward movements of compound rest caused in compound rest operating process. **Figure 19** presents the initial and results during external straight turning process. **Figure 20** shows the initial and results during internal straight turning process. **Figure 21** shows the initial and results during facing process. **Figure 22** shows the initial and results during necking process. **Figure 23** shows the initial and results during external threading process. **Figure 24** shows the initial and results during external taper turning process.

5. Conclusions and Results

The paper integrates virtual reality technology with the application of 3D solid model to complete a virtual operation platform based on the transmitting principles of lathe machine during practical operation. At the same time, the paper has completed the virtual machining for various lathe works. Users are able to learn in the simulated environment without scruple, increasing the effects of training. After the accumulation of learning experience, it can be applied by users in the actual environment to accomplish the mission of operation.

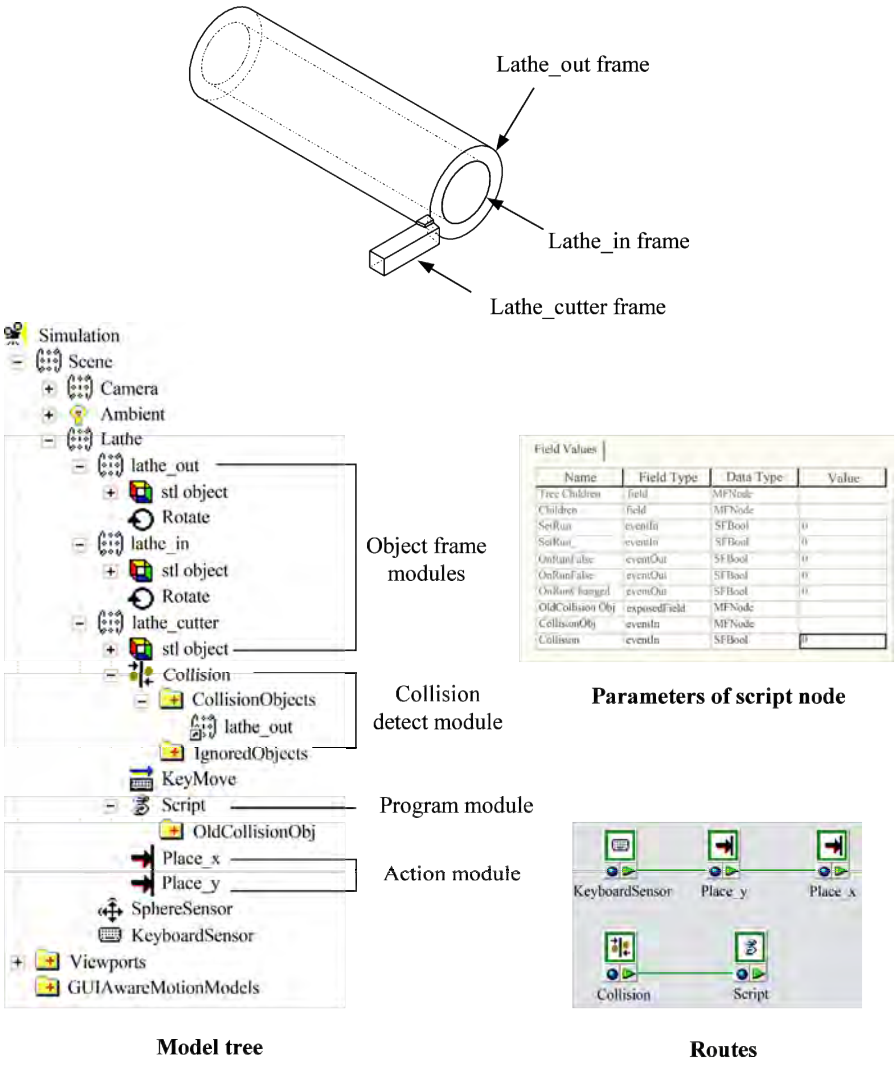


Figure 14. Relative data

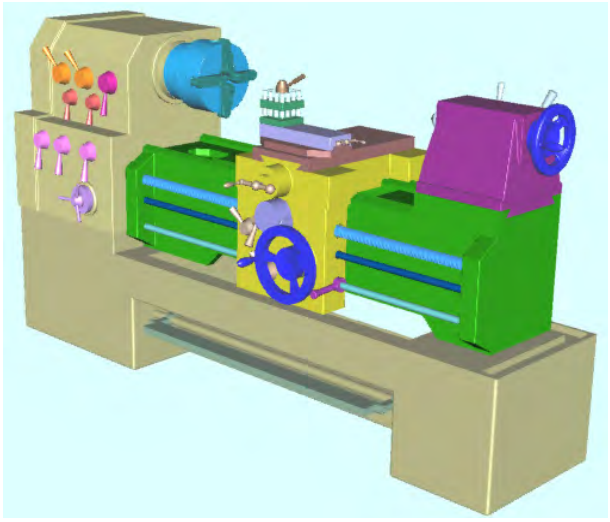
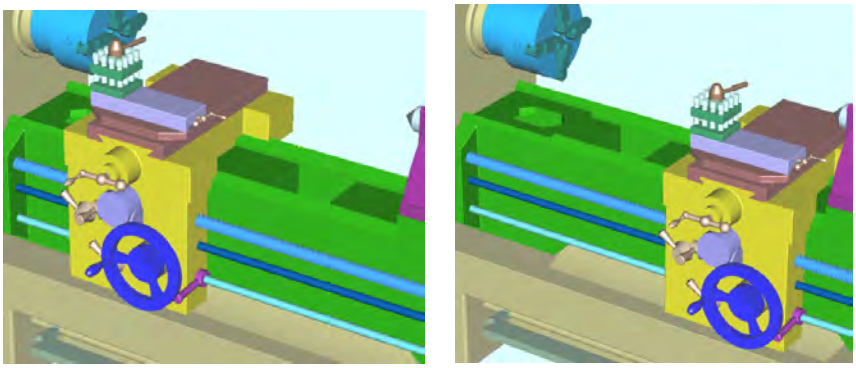
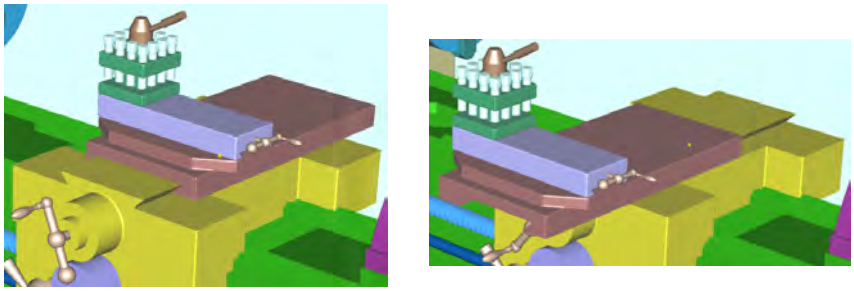


Figure 15. Virtual lathe machine



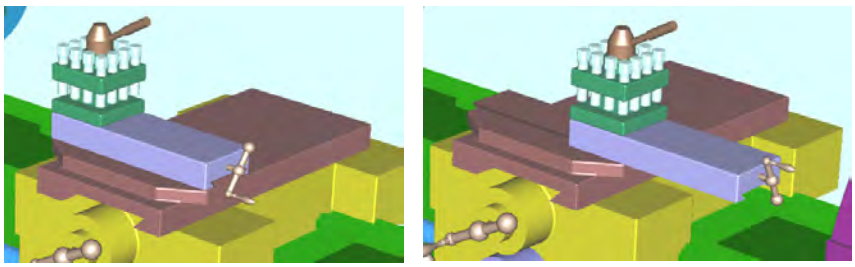
(a) Apron forward moving (b) Apron backward moving

Figure 16. Longitudinal feed operating process



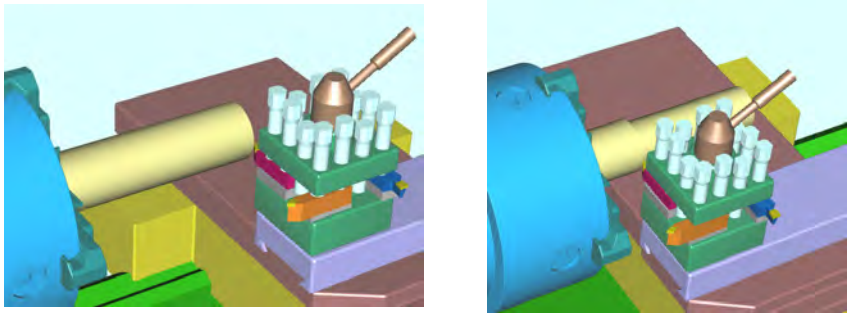
Carriage forward moving Carriage backward moving

Figure 17. Transverse feed operating process



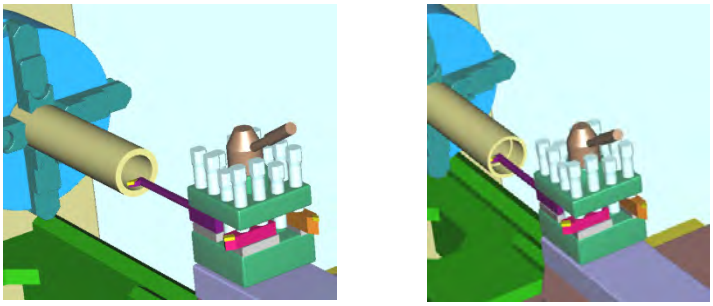
Compound rest forward moving Compound rest backward moving

Figure 18. Compound rest operating process



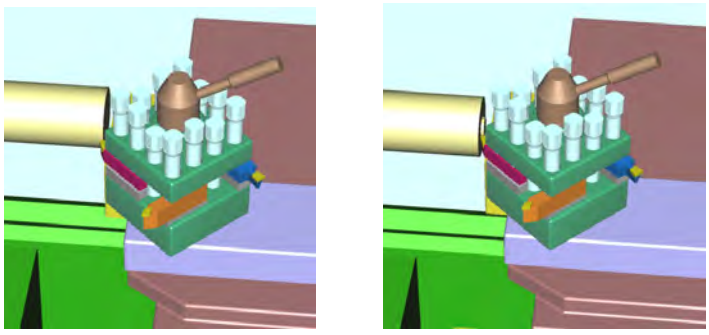
(a) Initial condition (b) During external straight turning process

Figure 19. External straight turning process



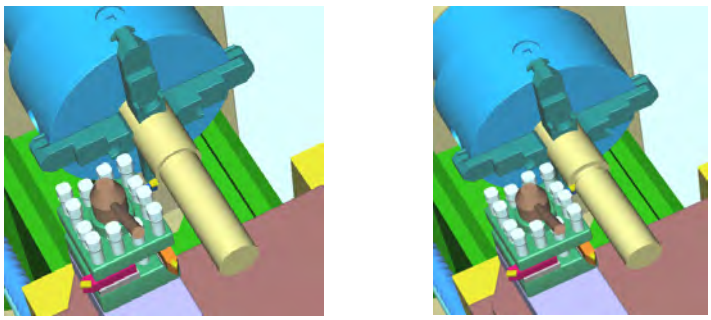
(a) Initial condition (b) During internal straight turning process

Figure 20. Internal straight turning process



(a) Initial condition (b) During facing process

Figure 21. Facing process

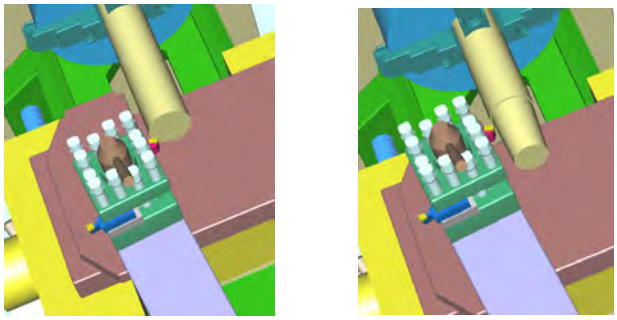


(a) Initial condition (b) During necking process

Figure 22. Necking process

(a) Initial condition (b) During external threading process

Figure 23. External threading process



(a) Initial condition

(b) During external taper turning process

Figure 24. External taper turning process

6. Acknowledgements

It is gratefully acknowledged that this research was supported by the National Science Council under contract No. NSC 95-2520-S-237-001.

REFERENCES

[1] M. Tavakoli, R. V. Patel, and M. Moallem, "A haptic interface for computer-integrated endoscopic surgery and training," Virtual Reality, Vol. 9, No. 2–3, pp. 160–176, 2006.

[2] E. Chen and B. Marcus, "Force feedback for surgical simulation," Proceedings of the IEEE, Vol. 86, No. 3, pp. 524–530, 1998.

[3] C. G. Shoaw, "The simulation system of virtual reality of intravenous injection," Master Thesis, National Central University, 2001.

[4] M. Dinsmore, N. Langrana, G. Burdea, and J. Ladeji, "Virtual reality training simulation for palpation of subsurface tumors," Proceedings of the 1997 Virtual Reality Annual International Symposium, Albuquerque, pp. 54–60, 1–5 March 1997.

[5] B. Korves and M. Loftus, "The application of immersive virtual reality for layout planning of manufacturing cells," Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, Vol. 213, No. 1, pp. 87–91, 1999.

[6] D. P. Sly, "A systematic approach to factory layout and design with factoryplan, factoryopt, and factoryflow," Proceedings of the 28th conference on Winter simulation, San Diego, pp. 584–587, 8–11 December 1996.

[7] R. G. Dewar, I. D. Carpenter, J. M. Ritchie, and J. E. Simmons, "Assembly planning in a virtual environment," Proceedings of Portland International Center for Management of Engineering and Technology, Portland, 27–31 July 1997.

[8] J. E. Brough, M. Schwartz, S. K. Gupta, D. K. Anand, R. Kavetsky, and R. Pettersen, "Towards the development of a virtual environment-based training system for mechanical assembly operations," Virtual Reality, Vol. 11, pp. 189–206, 2007.

[9] A. C. Boud, D. J. Haniff, C. Baber, and S. J. Steiner, "Virtual reality and augmented reality as a training tool for assembly tasks," 3rd International Conference on Information Visualization, London, pp. 32–36, 14–16 July 1999.

[10] S. Feiner, B. MacIntyre, and D. Seligmann, "Knowledge based augmented reality," Communications of the ACM, Vol. 36, No. 7, pp. 53–62, 1993.

[11] M. Billinghurst, S. Weghorts, and T. Furness, "Shared space: An augmented reality approach for computer support collaborative work," Virtual Reality, Vol. 3, pp. 25–26, 1998.

[12] N. Ye, P. Banerjee, A. Banerjee, and F. Dech, "A comparative study of assembly planning in traditional and virtual environments," IEEE Transactions on System, Man and Cybernetics, Part C: Application and Reviews, Vol. 29, No. 4, pp. 546–555, 1999.

[13] J. M. Ritchie, R. G. Dewar, and J. E. L. Simmons, "The generation and practical use of plans for manual assembly using immersive virtual reality," Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, Vol. 213, No. 5, pp. 461–474, 1999.

[14] S. Jayaram, U. Jayaram, Y. Wang, H. Tirumali, K. Lyons, and P. Hart, "VADE: A virtual assembly design environment," IEEE Computer Graphic and Application, Vol. 19, No. 6, pp. 44–50, 1999.

[15] R. Gupta, T. Sheridan, and D. Whitney, "Experiments using multi-modal virtual environments in design for assembly analysis," Presence: Teleoperators and Virtual Environments, Vol. 6, No. 3, pp. 318–338, 1997.

[16] Y. Hurmuzlu, A. Ephanov, and D. Stoianovici, "Effect of a pneumatically driven haptic interface on the perceptional capabilities of human operators," Presence: Teleoperators and Virtual Environments, Vol. 7, No. 3, pp. 290–307, 1998.

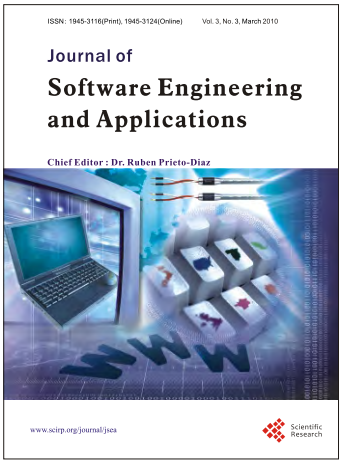
[17] Y. L. Wu, T. Chan, B. S. Jong, C. Yuan, and T. W. Lin, "A web-based virtual reality physics laboratory," Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies, Athens, 9–11 July 2003.

[18] E. Arroyo and J. Luis, "SRV: A virtual reality application to electrical substations operation training," IEEE International Conference on Multimedia Computing and

System, Vol. 1, 7–11 June 1999.

[19] L. Li, M. J. Zhang, F. J. Xu, and S. H. Liu, "ERT-VR: An immersive virtual reality system for emergency rescue training," Virtual Reality, Vol. 8, pp. 194–197, 2005.

Call For Papers



Journal of Software Engineering and Applications (JSEA)

ISSN 1945-3116 (print) ISSN 1945-3124 (online)
www.scirp.org/journal/jsea

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
Published Online March 2010 in SciRes (http://www.SciRP.org/journal/jsea).

TABLE OF CONTENTS

Volume 3, Number 3 **March 2010**

9771945311006 09