http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 5 (4): 709-718, 2006
ISSN 1812-5638

A Constraint-centered Two-layered Meta-model For Enterprise Modeling and Analysis

Liang Xue and Boqin Feng
Department of Computer Science and Technology,
Xi'an Jiaotong University, Xi'an 710049, China

Abstract: Enterprise model is a representation of the knowledge an organization has about itself and of what it would like this knowledge to be. The changing business environment makes a big challenge for the continuous improvement from as-is to to-be of the enterprise. Business constraints are defined as relationships maintained or enforced in the business structure and process to form the most volatile part of the business requirements. The proposed meta-model for enterprise modeling introduces the business constraints as the key element to make the model responsive to the business changes. It consists of two layers including conceptual layer and specification layer where the former provides a foundation for better communication and the latter provides business constraint language and constraint flow language to accommodate the changes and decrease the model maintaining effort. The meta-model forms a precise business as-is description based on which an Analytic Hierarchy Process based what-if analysis mechanism is proposed to analyze the business alternatives to give recommendations for the to-be design. Examples are presented to demonstrate the validness of the meta-model and its what-if analysis mechanism.

Key words: Meta-model, specification language, what-if analysis, Analytic Hierarchy Process (AHP)

INTRODUCTION

In the business environment of relentless change, the enterprise evolution is inevitable since changes generated by business policies and regulations need to be propagated onto the enterprise strategies and daily operating processes (Wan-Kadir and Loucopoulos, 2004). To study the organizational structure and behavior in the evolving enterprise, one needs an effective modeling method to capture the business as-is and to support the what-if analysis for better enterprise engineering in terms of the design of business to-be. Enterprise modeling is currently in operation either as a technique to represent and understand the structure and behavior of the enterprise, or as a technique to analyze business process and in many cases as support technique for business process reengineering (Mertins and Jochem, 2005). Broadly speaking, the enterprise modeling may be used to (Jureta and Faulkner, 2005; Vernadat, 2002): (1) achieve common understanding and agreement between stakeholders (e.g., managers, owners, workers, etc.) about different aspects of the organization; (2) analyze the current organizational structure and business processes in order to reveal problems and opportunities; (3) evaluate and compare alternative processes and structures; (4) reuse knowledge available in the organization.

The existing enterprise modeling approaches are focused on two domains including Computer Integrated Manufacturing (CIM) and software engineering.

Zhang and Deng (2004) indicates that in CIM domain IDEF is made of a series of modeling methods for functional modeling, information modeling, process modeling and ontology modeling of the organization. GRAI-GIM is a methodology for design and analysis of production systems and focuses on decision system analysis of the enterprise. CIMOSA provides guidelines, architecture and an advanced modeling language for enterprise modeling covering function, information, resource and organization aspects of the enterprise. PERA is a detailed methodology for enterprise engineering of industrial plants. GERAM is a generalization of CIMOSA, GIM and PERA and provides a methodology for enterprise engineering (PERA and GIM), a system life cycle (from PERA) and constructs for modeling (CIMOSA). IEM method employs the object-oriented approach to describe information and functions of objects as views on a single model of the system manufacturing company integrally. The core of the model structure contains the views of business process model and information model. In software engineering domain Zachman. (2005) framework defines a technique for achieving enterprise wide architecture. Object-oriented

Corresponding Author: Liang Xue, Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China

method is centered with the object concept and encapsulates the state and behavior. UML (OMG, 2003) is a modeling language which uses object-oriented methods for specifying, visualizing, constructing and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. Enterprise Knowledge Development (EKD) method (Rolland *et al.*, 2000) consists of three levels of models including goal models in business objective layer, role-activity model, object model, rule model, actor-role model in business process layer and information system model in information system layer to provide better comprehension from the knowledge perspective. Agent-based approach (Jureta and Faulkner, 2005) provides a set of concepts for conducting enterprise modeling to allow analysts to produce an enterprise model to capture the knowledge of an organization and of its business processes to derive the possible requirements specification of the system to-be. And formal analysis approaches include Petri-net (Desel and Erwin, 2000), situation calculus (Koubarakis and Plexousakis, 2002), simulation model (Nidumolu *et al.*, 1998), workflow management system (Yongjie, 2001) which all have complex models and variable states.

However, the existing enterprise modeling and analysis methods remain largely unharnessed (Delen and Benjamin, 2003) due to the following shortages: (1) They lack the change accommodation mechanism which makes the enterprise unresponsive to the enterprise changes and increases the maintaining overhead of the evolution of these models (Xu *et al.*, 2002); (2) Some enterprise models are just conceptual models and should be analyzed by hand. Others employ the complex mathematical models for analysis which are hard for the business users to comprehend and manipulate. On the other hand, the analysis models can not precisely capture the necessary variables with the complexity of the system increasing (Delen and Benjamin, 2003); (3) The knowledge reuse is difficult for the business users due to the heterogeneity of the enterprise models (Delen and Benjamin, 2003).

To tackle the above problems a business constraint centered two-layered meta-model is proposed to model the enterprise with the objective (1) to provide rapid responsiveness to the business changes on the enterprise; (2) to support the quantified what-if analysis of alternatives and make to-be design recommendations for the business users; (3) to form a set of reusable knowledge comprehensible for the different business stakeholders.

Business constraints are contained in the policies, rules, conventions, procedures, contracts, agreements, regulations, societal and physical laws which are the defining structure for an enterprise and are coded as business logic in information systems which form the IT facilities of an enterprise. They forge relationships between people, information, material and machines to make a system, form the rationale of the business operating and then can be used to change the behavior of a system to improve its performance, efficiency, or effectiveness. Once these constraints have been used to model the business behavior they can be systematically examined and, if necessary, tuned or replaced to accommodate changes and to improve the performance of the business system (Mayer *et al.*, 1996).

The main contributions of this study are geared towards:

- A constraint-centered two-layered meta-model proposed for modeling the enterprise to give better communication, rapid responsiveness and high reusability.
- An Analytic Hierarchy Process (AHP) based what-if analysis mechanism used to evaluate and compare the alternatives to reflect the performance comparisons via the multiple measurements or key performance indicators.

This approach has several advantages:

- To facilitate the communication between business users and technical users.
- To be responsive to the changes important for the enterprise evolvement and to evolve the model with the changing business environments.
- To give alternatives evaluation and comparison to recommend system to-be designs to improve business performance or facilitate business decision-makings.

CONSTRAINT-CENTERED TWO-LAYERED META-MODEL

This section presents the meta-model language for the enterprise to describe the enterprise from the conceptual and formalized perspectives to support better communication and automatic analysis. The meta-model language is classified into two layers including conceptual layer and specification layer. The conceptual layer indicates the key business elements and their relationships. The specification layer defines two languages including Business Constraint Language (BCL) and Constraint Flow Language (CFL) where BCL is used to describe the business constraints which are the key elements in the model and CFL describes the syntax and semantic of the constraint flow which forms the business behavior.

Conceptual layer: Figure 1 depicts the elements and their relationships. Formally speaking,

Definition 1 the conceptual layer is defined as a directed graph G = (V, E), where,

$V = V_n \cup V_u \cup V_a \cup V_e \cup V_m \cup V_g \cup V_p \cup V_c \cup V_t$ is a set of vertexes representing the business elements in an enterprise, where V_n specifies the finite set of entities and an entity is the concept or term designated in the business organization; V_u is the finite set of attributes and an attribute describes a piece of information of an entity; V_a corresponds to the finite set of activities and an activity is an action or movement with the actor as an executor; V_e is the finite set of events and an event is created after an activity occurred or a time period elapsed; V_m is the finite set of measurements and a measurement is an index or indicator used to valuate the business performance, e.g., cost and risk; V_g is the finite set of goals and a goal is a measurable end the organization tries to achieve; V_p is the finite set of processes and a process is considered as an abstract context to represent part of the business behavior; and V_c is the finite set of business constraints and business constraint is defined as a relationship enforced or maintained to restrict or govern one aspect of the business. $V_c = C_r \cup C_a \cup C_d$, where, $C_r = (f_r(n, u), \{T, F\})$ is a restriction constraint used to constrain the relationship between entities and entities or between entities and attributes, where, $f_r(n, u)$ is used to evaluate whether the entity n and attribute u objects satisfy the designated relationships; {T, F} describes the checking result with T for true and F for false; C_a = (precondition, a, successor state) is an action control constraint used to control the execution of a business activity, which describes that the activity a is triggered when the precondition is true and the successor state specifies the conditions which should be hold by the after-activity states; precondition $= f_s(n, u, e)$ is a function used to evaluate whether the entity n, attribute u and event e objects are in the state of s before the activity execution; $a = (n, u, e)^{s'}$ specifies an activity used to transform the state s of the before-activity into s' of after-activity; successor state = f(a) is a function used to check whether the objects state of s' satisfy the conditions designated in successor state; C_d = (prerequisite, conclusion) is a derivation constraint used to denote that the conclusion is derived when the prerequisite is true, where, prerequisite $= f_s(u)$ is a function used to evaluate whether the attribute object satisfy the state s; conclusion $= (u, m)^{s'}$ specifies that the state s' of the attribute or measurement is derived; V_t is the finite set of indirect interactions between the business constraints via the business objects they constrained. It is defined that constraint a impacts constraint b if only if the objects involved in the output nodes of the constraint

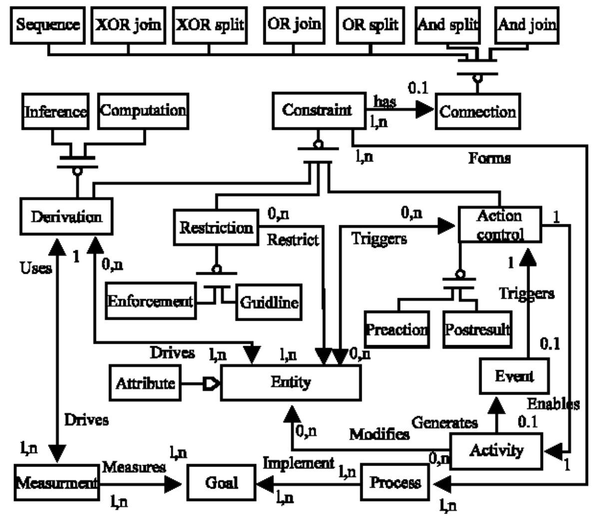

Fig. 1: The conceptual layer of the meta-model

a is contained in the input nodes of the constraint b which is denoted as $a \rightarrow b$. This is the sequential relationship between constraints. Another two relationships include synchronization and alternative. The synchronization includes AND-Join and AND-split, where AND-Join is expressed as $a \wedge b \rightarrow c$ denoting that the business constraint c is impacted by constraint a and b synchronously, where a, b and $c \in V_c$; AND-Split is expressed as $a \wedge b \rightarrow c$ denoting that the business constraint a impacts constraint b and c synchronously, where a, b and $c \in V_c$. The alternative includes XOR-Join and XOR-split, where XOR-Join is expressed as $a \oplus b \rightarrow c$ denoting that the business constraint c is impacted by one and only one of the constraints a or b, where a, b and $c \in V_c$; XOR-Split is expressed as $a \rightarrow b \oplus c$ denoting that the business constraint a impacts one and only one of the constraints b or c, where a, b and $c \in V_c$. $E = \{(x, y) | x, y \in V, x \neq y\}$ is a finite set of edges from x to y, for details an entity consists of a set of attributes; a restriction constraint restricts one or more entities; a action control constraint is triggered by the status or value of zero or more entities and by zero or one event. An activity is enabled by an action control constraint, generates an event after execution and modifies the status of zero or more entity; a goal is implemented by one or more business processes and measured by one or more measurements; a process consists of one or more business constraints; a derivation constraint derives the value of a measurement and uses the values of one or more entities.

In the meta-model the business constraints are separated from the business structure and behavior. They can be connected to different business objects to adapt to the changing environment. One example is given.

```
<constraint> = <restriction>|<action control>|<derivation>
<restriction> = <eliment><prep><relationship><element2>
<element> = <entity>|<entity><attribute
<prep> = <MUSTISHOULD><HAVEBE>
<relationship>:: = association>|<magnitude>|<subsume>|<relation op>
<association> ::= <ATLEAST|ATMOST|EXACTLY
<magnitude> ::= ONLY ONE IN|ANY ONE IN|NOT IN
<subsume> ::= CHILDREN OF|SIBLING OF|SUB-TYPE OF
<element2> ::= <element1>|<constant>
<constant> ::= <value>|<value list>
<entry> ::= <STRING>
<attribute> ::= <STRING>
<value> ::= <INTEGER>|<FLOAT>|<DOUBLE>|<STRING>|<USERTYPE>
<value list> ::= <ENUMERTION>
<value> ::= <preaction>|<postresult>
<preaction> ::= <IF><bool-expr>|<THEN><DO><activity>
<postresult> ::= <bool-expr><cheching>
<activity> ::= <STRING>
<cheching> ::= <FUNC_CHECHING>
<bool-expr> ::= <comparison><logic_op><bool_expr><comparison>
<comparison> ::= <arith_expr><relation_op><arith_expr>not<comparison>
<relation_op> ::= <|>|≤|≥|==|≠
<logic_op> ::= AND|OR
<arith_expr> ::= <term>+<arith_expr>|<term>|<arith_expr>|<term>
<term> ::= <factor>*<term>|<factor>/<term>|<factor>
<factor> ::= (<arith_expr>)|<element>|<constant>
<element> ::= <element>|<event>
<event> ::= <STRING>
<derivation> ::= <computation>|<infeerence>
<computation> ::= <result>"="<arith_expr>
<result> ::= <element1>|<measurement>
<measurement> ::= <STRING>
<inference> ::= if<bool_expr>then<result>" = "<constant>
```

Fig. 2: BCLof the meta-model

Example 1: In an supermarket the milk quality checking process “To begin the specification and quality checking after the ordered milk arrives” may be changed into “To begin specification checking after the milk arrives and to begin quality checking after the specification checking is finished”. To simulate the changes of the process the model should be first constructed. By the discovery approach (Xue, 2006) three constraints should be uncovered and defined as follows:

CID1 If the milk arrives do specification checking;
CID2 If the milk arrives do quality checking;
CID3 If the specification checking finished do quality checking;

The original process is represented by CID1 and CID2. In the changed process the CID2 is replaced as CID3. The lines connecting constraint CID2 with event “milk arrives” and activity “quality checking” should be changed into new lines connecting constraint CID3 to the event “specification finished” and activity “quality checking” in the constructed model based on the conceptual layer.

Specification layer: The specification layer provides two language definitions. One is BCL and the other is CFL where the former defines the syntax and semantic of the business constraint and the latter gives the constraint flow representing the business behavior.

BCL is defined to specify the business constraints to facilitate the comprehension and the analysis of the business constraints. It makes the business constraints be more responsive to the changes and saves the modification effort in maintaining mode. As shown in Fig. 2, the grammar, in Backus Naur form, defines these specification elements for each kind of constraints. The grammar uses the meta-level elements to improve the efficiency which makes it more abstract.

Example 2: The business manager make a new business policy said that the quality checking can begin when the specification finished 80%.Based on BCL the three constraints in the example1 should be represented as “If milk_arrive == "ready" then do check_specification”, “If milk_arrive == "ready" then do check_quality” and “If specification_rate == 1 then do check_quality”. The objects connected with the business constraints can be modified in the conceptual layer. The state changes of the objects in one business constraint then need the help of the BCL. Therefore, CID3 is changed into “If specification_rate > 0.8 then do check_quality” in BCL to reflect the policy modification. The business constraints

```
<FLOW> ::= <CONSTRAINTS>|<CONSTRAINTS>":"<FLOW>
<CONSTRAINTS> ::= <ATOMIC>|<ATOMIC><OPERATOR><CONSTRAINTS>
<ATOMIC> ::= "("<CONSTRAINTS>")"|<IDENTIFIER>
<OP> ::= →|⊕|∧
```

Fig. 3: CFL of the meta-model

defined by BCL make the elements of the constraints more delicate than in the conceptual layer which enables the modification more flexible.

The behavior description is the way of chaining the business constraints and their associated elements. The business behavior is defined as CFL to allow the meta-model to be completed by adding dynamic descriptions. This language is easily used by computer novices and sufficiently expressive to describe the business behavior in the form of chain of constraints.

Any behavioral definition language must contain, at the very least, the following chaining between task components: sequence, repetition, conditional execution, synchronization and parallelism (Goc *et al*., 2002). For CFL the sequence, conditional execution and synchronization are used and they are defined as the notations of "→", "⊕" and "∧" respectively in the language. The ";" can be seen as a business process context in which the set of business constraints serves a business goal (Fig. 3).

Example 3: The process modification in example1 triggers the modification of business behavior which is described by CFL from "CID1∧CID2" to "CID1→CID3".

The examples proves that the models constructed by the proposed meta-model is responsive to the changing environment. And the changes can be embodied in three levels from conceptual level (conceptual layer) to analysis and behavior level (specification layer).

AHP-BASED WHAT-IF ANALYSIS MECHANISM

With new opportunities emerging or the identification of the weaknesses in the exiting model the alternatives should be proposed and evaluated to get the best design. What-if analysis presents an approach to evaluate these alternatives against the business measurements and goals where an alternative is a sub-set of a constraint flow. AHP (Saaty, 1988) produces balanced weights of multiple measurements. AHP-based what-if analysis approach then transforms multiple measurements into one measurement, makes best alternative recommendations and facilitates the business users to make reliable and timely decisions or give to-be designs. The mechanism is designed into the following four steps.

Step 1 modeling enterprise: The constraint centered two-layered enterprise model can be constructed according to the modeling procedure defined as follows:

- Identifying business elements

All kinds of enterprise documents and domain knowledge should be collected to identify the goal via the goal-measurement tree (Xue, 2005), entity and their attributes via the ER diagram, find the activity and event via the event diagram and to discover the business constraints via the discovery approach (Xue, 2006).

- Constructing conceptual layer of the model

The identified business elements should be constructed and associated together based on the syntax and the semantic of the conceptual layer of the meta-model.

- Filling in the specification layer

BCL and CFL are used to specify the business constraints and describe the business behavior.

- Refining the model

The constructed model should be examined carefully to validate and verify the possible errors.

The final completed model indicates the static structure and the dynamic behavior of the enterprise. And the business constraints, the most volatile part of the business requirements, are separated from the structure and process of an enterprise which helps the model to be responsive to the business changes and improve the business performance continuously.

Step 2 generating alternatives: The business constraints in the constraint flow can be tuned or replaced to forge different behavior based on the user's preference and the changing business environment. Each alternative is measured by the user defined measurements in the model and try to achieve the goal in different degrees. The alternative execution algorithm is defined as follows:

While not end node
Analyzing the starting conditions
Finding the triggering constraints
Parsing the constraints
End while
Retrieve the values of the measurements

The execution results of the alternative are indicated as the state changes of the objects. The values of the measurement are the key indicators to the business performance.

Given M = $\{m_1,..., m_n\}$ is the finite set of measurements, P = $\{p_1,\dots,p_q\}$ is the finite set of alternatives. The values of the n measurements for the q alternatives are described as a matrix

$$R = (r_{ij})_{n\times q} = \begin{matrix} r_{11}, & r_{12}, & \dots, & r_{1q} \\ r_{21}, & r_{22}, & \dots, & r_{2q} \\ \dots & \dots & \dots & \dots \\ r_{n1}, & r_{n2}, & \dots, & r_{nq} \end{matrix}, \; i = 1, ..., n, j = 1, ..., q$$

where, r_{ij} is the value of the measurement m_i for the alternative p_j. And the values of the same column of the matrix should be normalized.

Step 3 multi-measurement balancing: The different measurements of an alternative should be assigned with different weights that may reflect the individual preference or observations of the domain experts.

The weights of the n measurements are $W = [w_1, w_2,\dots,w_n]$ which satisfy the following conditions:

$$\sum_{i=1}^{n} w_i = 1, \; w_i \in [0,1]$$

In simple situations these weights can be designated individually. To achieve more precision in complex problem the AHP approach (Saaty, 1988) is used to solve the weight assignment. The detailed steps of the approach are designed as follows:

As shown in Table 1, M_{ij} is the relevant significance of M_i to M_j where M_i and M_j are two measurements shows relationship between measurements

$$A = (M_{ij})_{n\times n} = \begin{matrix} M_{11}, & M_{12}, & \dots, & M_{1n} \\ M_{21}, & M_{22}, & \dots, & M_{2n} \\ \dots & \dots & \dots & \dots \\ M_{n1}, & M_{n2}, & \dots, & M_{nn} \end{matrix}, \; i = 1, ..., n, j = 1, ..., n$$

- Computing weights

The n weights $w_1,...,w_n$ are generated by multiplying the elements in each row and then get the n square of the production which is shown in formula 1. The w_i then should be normalized.

$$w_i = \sqrt[n]{\prod_{i=1}^{n} M_{ij}} \tag{1}$$

- Consistency validation

Table 1: Constructing the hierarchy matrix

M_{ij}	Description
1	The two measurements have equal significance
2	The former is more important than the latter
3	The former is explicitly more important than the latter
4	The former is strongly more important than the latter
5	The former is extremely more important than the latter
rcip rocal	The situation of the significance of the latter to the former, the value of i to j is the reciprocal of the value of j to I.

The hierarchy matrix can be generated based on table 1 and expert opinions.

Table 2: RI of matrix with different orders

Order(n)	1	2	3	4	5	6
RI	0.00	0.00	0.58	0.90	1.12	1.24

It deems that the consistency is satisfied when the random consistency rate (CR)< 0.10. Otherwise tune the hierarchy matrix until the CR is satisfied.

$$CR = \frac{Cl}{Rl} \tag{2}$$

$$\text{where, } CI = \frac{\lambda_{max} - n}{n-1} = 0.015, \; \lambda_{max} = \sum_{i=1}^{n} \frac{(AW)_i}{nW_i}$$

is the maximal eigenvalue of the matrix and RI can be found from Table 2 which shows the relationship of the RI and the matrix with order from one to six.

Step 4 alternative recommendation: The contribution of each alternative can be computed and then the degree of the goal achievement can be compared. The contribution of the alternatives is defined as $O = [o_1, o_2, \dots, o_q] =$

$$WR = [w_1, w_2,..., w_n] = \begin{matrix} r_{11}, & r_{12}, & \dots, & r_{1q} \\ r_{21}, & r_{22}, & \dots, & r_{2q} \\ \dots & \dots & \dots & \dots \\ r_{n1}, & r_{n2}, & \dots, & r_{nq} \end{matrix} \tag{3}$$

The o_{max} = Optimum $(o_1,o_2,\dots,o_q)$ designates the best alternative and then it can be recommended.

AN EXAMPLE

To validate the proposed approach a milk checking process in a supermarket is adopted. In the process the specification, quantity and quality of the milk need to be checked and the supplier should be notified when the checking results are not satisfied. The process is redesigned to promote the customer satisfaction through the proposed modeling and analysis approach after an customer complaining the spoiled milk.

Step 1 modeling milk checking process: Figure 4 shows the identified model elements including one process, one goal, seven measurements, three entities and seven

Goal(1): promote customer satisfaction
Process(1): milk checking
Measurement(7): quantity rate, specification rate, quality rate, cost, risk resource, and time.
Entities (3) and their attributes (7) : milk (specification. quantity). order (o_specification, o_quantity), and sample milk (sample rate, sample quantity, sample quality)
Event(1): milk arrives
Activity(4): check quality, check quantity, check specification, and notify supplier
Constraints(10):
CID1(preaction): If the milk arrives do checking the specification [20,0,15,2,10]
CID2(preaction): If the milk arrives do checking the quantity [40,0,2,5,20]
CID3(preaction): If the milk arrives do checking the quality [100,0,5,10,50]
CID4(computation): specification rate = specification/o_specification [10,0,1,2,]
CID5(computation): quantity rate = quantity/o_ quantity [10,0,1,2]
CID6(computation): sample rate = sample quantity/total quantity [10,0,1,2]
CID7(enforcement): sample quality must be any one in the category of "good, accept, and Bad" [5,0.7,0,0]
CID8(inference): If sample rate>10% and the quality is best then quality rate is 1 [15,0,8,1,5]
CID9(inference): If sample rate>20% and the quality is accept then quality rate is 1 [20,0.9,1,5]
CID10(preaction): If the quality is bad notify the user [5,0.5,1,5]

Fig. 4: Identified model elements of the example

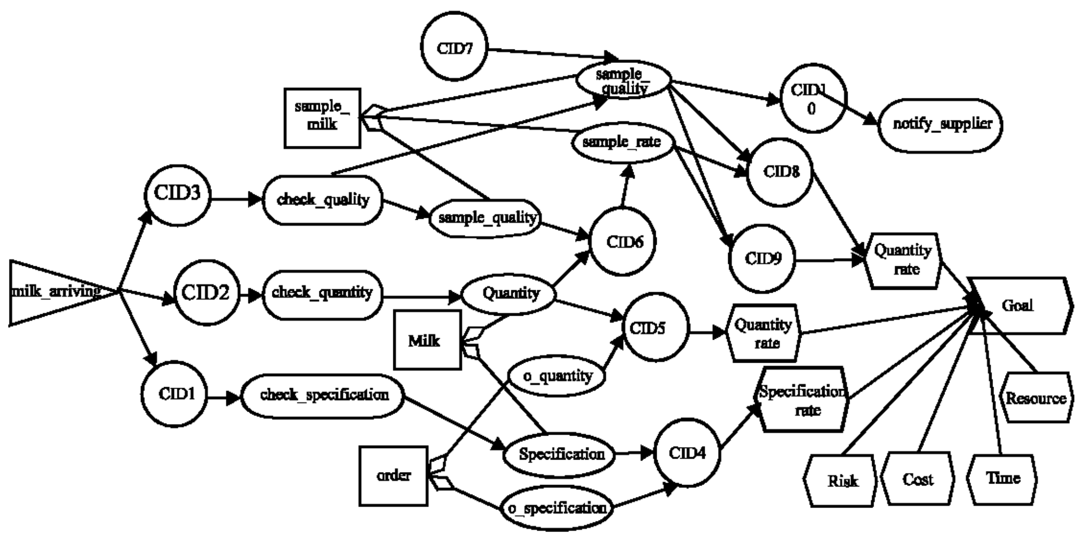

Fig. 5: Conceptual layer of the constructed model

CID1 If milk_arrive ==ready then do check_specification
CID2 If milk_arrive ==ready then do check_quantity
CID3 If milk_arrive ==ready then do check_quality
CID4 specification rate = milk.specification/order.specification
CID5 quantity_rate = milk.quantity/order.quantity
CID6 sample_rate = sample_quantity/milk.quantity
CID7 Any_one(sample_milk.quantity,good,accept,bad
CID8 if sample_milk.sample_rate>10% AND sample_milk.quality == good then quality_rate = 1
CID9 if sample_milk.sample_rate>20% AND sample_milk.quality == accept then quality_rate = 1
CID10 if sample_milk.quality == bad then do notify_supplier

Fig. 6: Specification layer of the constructed model

attributes, one event, four activities and ten constraints. In the seven measurements the first three are used to ensure the completion of the checking process and the last four are used to measure the overhead of each constraint and then to measure the alternative performance. In constraints part the name and body are and supplemented by the last four measurements values separated by comma. Those values are assigned by experienced domain experts. The conceptual layer of the constructed model is shown in Fig. 5 where rectangle represents an entity, ellipse represents an attribute, triangle represents an event, rounded rectangle represents an activity, circle represents a constraint, hexagon represents a measurement and the rightest fat arrow

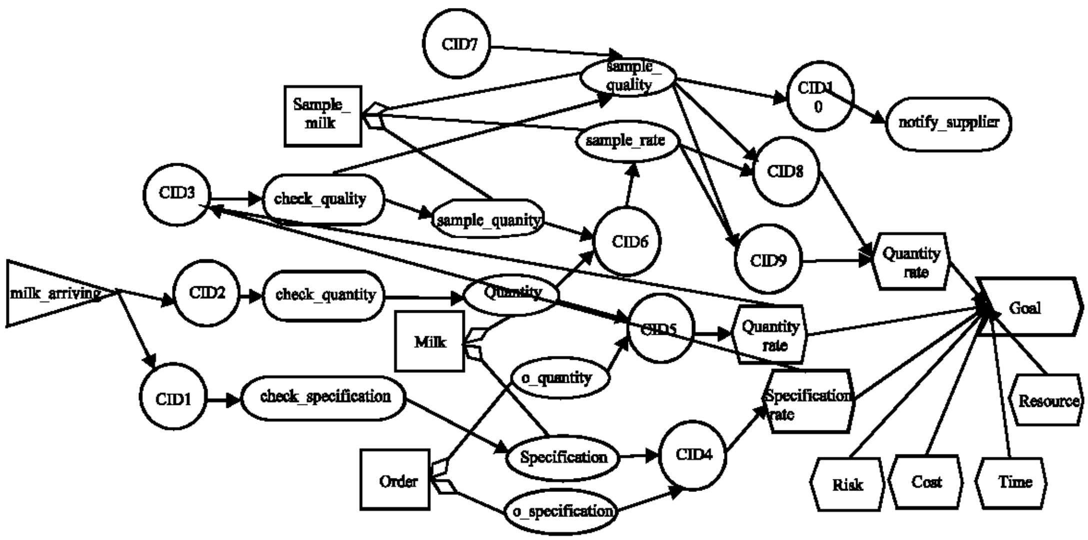

Fig. 7: Conceptual layer of alternative2

Table 3: Hierarchy matrix

	Cost	Risk	Resource	Time
cost	1	1/2	5	3
Risk	2	1	3	2
Resource	1/5	1/3	1	1/2
Time	1/3	1/2	2	1

represents a goal. The specification layer is defined in Fig. 6. The constraint flow of the example is defined as (CID1→CID4)∧(CID2→CID5)∧(CID3→(CID6∧CID7)→(CID8⊕CID9⊕CID10)). The constructed model gives the as-is description of the milk checking process.

Step 2 generating alternatives: The model is examined and three alternatives are proposed to promote customer satisfaction after the customer complained that the milk is spoiled.

Alternative1 is to increase the sample rate to control the quality. This can be reflected into the modification of business constraints CID8 and CID9. Then CID8 is replaced as "if sample_milk.sample_rate > 20% AND sample_milk.quality == "good" then quality_rate =1" and CID9 is replaced as "if sample_milk.sample_rate > 50% AND sample_milk.quality == "accept" then quality_rate =1" respectively. In CID8 the rate is increased to 20% and in CID9 the rate is increased to 50%. The new measurement value of cost, risk, resource and time are defined as [20, 0.4, 2, 5] and [25, 0.4, 2, 5] for new business constraints CID8 and CID9 respectively where both cost and resource increases though both risk decreases. The conceptual layer and the constraint flow need not to be changed.

Alternative2 is proposed to control the specification and quantity checks before quality checking to avoid some errors in time. Then the synchronized checks of quantity, specification and quality should be modified into sequential execution. This also can be reflected in the business constraints changes. The constraint CID3 is replaced as "if quantity_rate == 1 and specification_rate == 1 then do check_quality" which specifies that the quality check begins after the quantity and specification checks have been finished. The new values of cost, risk, resource and time are defined as [100, 0.3, 10, 50]. Fig. 7 shows the changed conceptual layer where the line from event to CID3 is deleted and two new lines are generated from the measurement quantity rate and specification rate respectively to the CID3. The constraint flow then turns into "((CID2→CID5)∧(CID3→CID4))→CID1→(CID6∧CID7)→(CID8⊕CID9⊕CID10)".

Alternative3 is the combination of alternative1 and alternative2. After execution and normalization the values of measurements cost, risk, resource and time of the three alternatives are $R_{3\times4}$ =

$$\begin{matrix} 245 & 2.95 & 25 & 57 \\ 235 & 3.65 & 23 & 91 \\ 245 & 2.75 & 25 & 91 \end{matrix} = \begin{matrix} 1 & 0.22 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{matrix}$$

Step 3 multi-measurement balancing: The hierarchy matrix of the four measurements has been defined as Table 3. The weight W = [0.35, 0.40, 0.09, 0.16] which is calculated by formula 1. The λmax= 4.156, CI = 0.052, RI = 0.09 and CR = 0.052/0.9 = 0.058<0.1 which is calculated by formula 2 and it conforms to the consistency criteria which prove that the calculated weight is valid. The weights assigned to the four measurements are 0.35, 0.4, 0.09 and 0.16, respectively. The measurements cost and risk have more priority than time and resource.

Step 4 alternative recommendation: The contributions of the three alternatives are 0.528, 0.56 and 0.60 respectively. In this example the alternative with minimum result will be the best satisfying the goal. Therefore, the alternative1 provides the best feasible way to promote customer satisfactory. The possible problem has been the sampling control in quality checking too loose.

In this example the business constraints can be flexibly tuned or replaced to evolve the model with the changing business environment and to facilitate decision making for the business users. To achieve better analysis results two points need to be considered. One is to design the appropriate measurements for the goal to precisely reflect the achievement of the goal. The second point is to tune the business constraints as many as possible to find the most appropriate alternatives. For example the sample rate in the example can be changed into different values to indicate different alternatives. Additionally the weights assigned to the measurements can be modified to represent the preference of the business users. For example the business user may prefer high usage of resource rather than low cost. In a word the constraint-centered two-layered model can easily describe the business as-is and evolves with the changing environment. Following the steps in the what-if analysis mechanism the alternatives can be evaluated to facilitate decision making and to recommend to-be design.

CONCLUSIONS

Most existing enterprise modeling approaches don't accommodate the business changes and provides complex analysis mechanism incomprehensible for business users. The business constraints represent the most volatile business requirements and demonstrate the data integrity from the relationship perspective which make it easily for the business users to tune or replace the constraints to make the model precisely describe the business as-is. Even as the enterprise changes over time, the enterprise models can evolve to reflect these changes and, hence, be reused in future analysis efforts. The two layered models can easily relate the results of analysis effort to the enterprise being analyzed. And it can be easily handled than the variables modification in general simulation models. The two layers enable the business user to exercise better control over the modeling efforts and to participate more fully in the design and evaluation of alternatives. It can be used to transfer enterprise-specific knowledge among business stakeholders in a comprehensible way. The benefits of using the constraint centered two-layered models therefore increase domain expert's acceptance and motivations to using analysis methods.

The constraint centered two-layered meta-model language makes the business more adaptive to the changing environment where the conceptual layer provids a good basis for the communication between different business stakeholders and specification layer gives formal specifications for business constraints and their flow to increase the responsiveness and decrease the maintaining effort than in the programming mode. The AHP-based what-if analysis mechanism evaluates and compares multiple alternatives against multiple measurements to provide business performance comparisons and give best to-be design recommendations.

The future work focuses on the verification and validation of the models both from the syntax and semantic perspectives.

ACKNOWLEDGMENT

This work was supported by the Joint Study between IBM China Research Lab and Xi'an Jiaotong University.

REFERENCES

Delen, D. and P.C. Benjamin, 2003. Towards a truly integrated enterprise modeling and analysis environment. Computers in Industry, 51: 257-268.

Desel, J. and T. Erwin, 2000. Modeling, simulation and analysis of business processes business process management. LNCS., 1806: 129-141.

Goc, L.M., C. Frydman and L. Torres, 2002. Verification and validation of the SACHEM conceptual model. Intl. J. Human-Computer Studies, 56: 199-223.

Jureta, I. and S. Faulkner, 2005. An agent-oriented meta-model for enterprise modelling. ER Workshops, LNCS., 3770: 151-161.

Koubarakis, M. and D. Plexousakis, 2002. A formal framework for business process modeling and design. Information Systems, 27: 299-319.

Mayer, R.J., K.M. Painter and M. Lingineni, 1996. Toward a method for business constraint discovery (IDEF9) Version 1.0.

Mertins, K. and R. Jochem, 2005. Architectures, methods and tools for enterprise engineering. Intl. J. Production Economics, 98: 179-188.

Nidumolu, S.R., N.M. Menon *et al.*, 1998. Object-oriented business process modeling and simulation: A discrete event system specification framework. Simulation Practice and Theory, 6: 533-571.

OMG., 2003. UML 2.0 Superstructure Specification. www.omg.org

Rolland, C., S. Nurcan and G. Grosza, 2000. A decision-making pattern for guiding the enterprise knowledge development process. Information and Software Technol., 42: 313-331.

Saaty, T.L., 1988. Multicriteria Decision Making: The Analytic Hierarchy Process. RWS Publications, Pittsburgh.

Vernadat, F.B., 2002. Enterprise modeling and integration (EMI): Current status and research perspectives. Ann. Rev. Control, 26: 15-25.

Wan-Kadir, W.M.N. and P. Loucopoulos, 2004. Relating evolving business rules to software design. J. Syst. Architecture, 50: 367-382.

Xu, H.X., L. Zhang and B.S. Zhou, 2002. Research on Enterprise Modeling of Agile Manufacturing. EDCIS, LNCS., 2480: 247-256.

Xue, L., 2005. Internal Report on Business Constraint Management.

Xue, L.B. and Q. Feng, 2006. A model-driven approach for business constraints discovery. Information Technol. J., 5: 454-459.

Yongjie, R., 2001. Multi-agent workflow model (MAWM) A workflow model designed for Chinese business processes. Ph.D Thesis, Chinese University of Hong Kong.

Zachman, J.A., 2005. Zachman Framework. http://www.zifa.com.

Zhang, X.M. and Z.Q. Deng, 2004. Analysis of enterprise modeling methodology and evaluation of modeling tools. J. Syst. Simulation, 16: 511-515.