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 (3): 454-459, 2006
ISSN 1812-5638

A Model-driven Approach for Business Constraints Discovery

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

Abstract: Business constraints restrict the enterprise structure, govern the business behavior and contribute to the business goals. Due to stuff turnover and lack of documentation the knowledge of what constraints exist and how they interact are often poorly defined. This can seriously hinder an organization to understand how the business behaves and operates and to seize new opportunities or improve the business performance. A model-driven discovery approach is proposed to help the user to uncover the business constraints from the enterprise documents. This approach proposes a meta-model to describe the ontological constructs of the business constraints enforcing environment, defines a set of formal templates to document the business constraints to facilitate communication, sharing and reuse, gives a verification algorithm to ensure the completeness and validness of the discovered business constraints, defines the interactions among business constraints and finally proposes an elicitation procedure to efficiently guide the business users to discover the business constraints. Moreover, an example is presented to demonstrate the discovery approach of the business constraints and their usage for decision-making.

Key words: Business constraints, meta-model, discovery approach, formal template, verification

INTRODUCTION

Business constraints initiate, enable, govern and limit the business behaviors to accomplish the goals of a business system (Mayer *et al*., 1996). However, due to staff turnover and lack of documentation, precise knowledge of what constraints are enforced by a system is often not available. The collection of business constraints that forge an enterprise system is generally poorly defined - the knowledge of what constraints exists and how those constraints interact is at best incomplete, disjointed, distributed and often completely unknown. This can seriously hinder an organization's ability to understand the data stored in its systems and to evolve the systems to implement new business policies (Fu *et al*., 2002; Fu *et al*., 2004). This study aims to discover and document the business constraints thoroughly and explicitly to build a useful knowledge pool for communication, sharing and reuse which forms a good foundation for business decision-making.

Business constraints are first systematically examined in IDEF9 (Mayer *et al*., 1996) in which it is defined as a relationship maintained or hold in a given context. It protects data integrity from the relationship view (Halle, 2002). For example, a mobile phone service provider may have "new customers must be eligible for free connections" as a constraint (Fu *et al*., 2002). It is defined from the business view and independent of technical implementation so that it is oriented for both the business people and the technical people (Fair Isaac, 2003). The business constraints possibly originate from three channels including enterprise provisions, knowledge workers and information systems. The enterprise provisions consist of regulations, specifications, laws and policies etc. The knowledge workers or domain experts should be interviewed for valuable business operations or rules. The information systems can provide user requirements, handbooks and source code. The former two channels provide a top down source containing business constraints and the latter is a bottom up source.

Several researches are focusing on discovering business constraints or business rules. These efforts deal with three kinds of discovering mechanisms. One is to mine business rules from the code (Wang *et al*., 2004; Huang *et al*., 1996; Sneed and Erdos, 1996), the second is to provide a discovery procedure for business analyst to identify business constraints or rules (Mayer *et al*., 1996; Halle, 2002; Kardasis and Loucopoulos, 2004) and the third is to discover the business constraints or business rules from the decision space based on algorithms (Rosca *et al*., 2002; Rosca and Wild, 2002).

Wang *et al*. (2004) propose a framework for extracting business rules from large legacy systems. The proposed framework consists of five main steps: slicing program, identifying domain variables, data analysis, presenting business rules and business validation. Huang *et al*.

Corresponding Author: Liang Xue, Department of Computer Science and Technology, Computer Teaching and Experiment Center, Xi'an Jiaotong University, Xi'an 710049, China Tel: 86-029-88010054

(1996) propose another solution to the business rule extraction problem that combines variable classifications, program slicing and hierarchical abstraction. Sneed and Erdos (1996) use data output identification and program stripping techniques to identify and extract business rules. Mayer *et al*. (1996) define a discovery procedure tailored for business users to find business constraints based on interviewing notes and organization materials like policies, regulations and manuals etc. The procedure includes project organization and planning, collection of evidence, candidate constraints substantiation, constraints validation and refining. A more refined discovery procedure (Halle, 2002) is provided for business users and it begins from the business requirements like use case, through the discovery roadmaps and finally reaches the business rules. Kardasis and Loucopoulos (2004) propose a simple procedure for discovering one category of constraints which consists of five steps: (1) to select the particular context; (2) to identify the actors and main activities, (3) to find the information objects and activity enables; (4) finally to get the operational rule. Based on the requirement of decision space, some algorithms are derived to discover business rules. In the algorithm the rules are classified into three levels with each level contributing to its upper level an evaluated weight. The lowest level rule is derived from decision tree and then the rules are derived and the best solution is provided based on the weight (Rosca *et al*, 2002; Rosca and Wild, 2002).

There are some shortages in the existing discovery approaches. The first discovery mechanism can be difficult to impossible since the manifestation of the rules might be scattered throughout the code, also it might be flawed. This case may be the source of an inflexible enterprise behavior since the business rules are hard coded and distributed into the system implementation and therefore difficult to retrieve and change (Rosca *et al*., 2002). The second discovery mechanism only provides operating procedures which act as guidelines. And without a systematic approach the new user usually can not find out all the constraints of the enterprise, whereas the third mechanism is focusing on the algorithms and it's hard to extend and apply them in the business scenarios. Moreover, these works did not give any more formal expressing approach for the communication of the business constraints.

To tackle these problems we propose a novel discovery approach, which aims for business user with ease of use, is a systematic approach which helps to discover the business constraints thoroughly and ensure their usefulness. It proposes a meta-model to describe the underlying rationale and ontological construct of the enterprise where the business constraints are enforced. A meta-model can be seen as a design framework that describes the basic model elements and the relationships between the model elements as well as their semantics (Rosemann and Green, 2002). The relationships form a foundation for the business constraints when instantiated. The main contributions of this article are: (1) a meta-model which acts as a formal basis; (2) a set of formal templates defined to adapt the business constraints into the explicitly documented asset; (3) a verification algorithm proposed to ensure the validness and the completeness of the business constraints; (4) interactions among constraints defined to specify the dynamic business behavior of an enterprise from the business constraints perspective; (5) an elicit procedure used to guide the business user to uncover the business constraints from the enterprise information.

This approach has several advantages:

- To discover and document the implicit knowledge important for the enterprise evolvement.
- To facilitate the communication between business users and technical users.
- To help the business managers make decisions with intact data supporting.
- To give an approach to easily tune the business constraints for change management and performance improvement.

DISCOVERY APPROACH

This section defines a meta-model to describe the ontological construct of the enforcing environment of the business constraints to facilitate the identification of the business constraints, designates a set of formal templates to describe the business constraints, proposes a verification algorithm to ensure the completeness and validness of the business constraints, defines the constraints interactions to guide the decision making and presents an elicitation procedure to guide the business user.

Meta-model: The business constraints maintain the system structure and behavior to achieve system goals from the system view. And they coordinate the business objects including entities and activities to achieve the business goals in an enterprise. The meta-model is then developed based on a business system. It consists of the business objects and their relationships. The business objects include the entity, attribute, activity, event, measurement and goal where, the entity is the concepts or terms designated in the business organization; attribute

describes a piece of information; activity is defined as an action or movement with the actor as an executor; event is created whenever an activity occurred or a time period elapsed; measurement, an index, indicates the extent of the goal achievement; and goal is a measurable end the organization tries to achieve. Their relationships form a constructive source for the business constraints when instantiated in a particular scenario or organization.

Definition 1: A meta-model for business constraints discovery is defined as a directed graph $G = (V, E)$, where, $V = V_n \cup V_u \cup V_a \cup V_e \cup V_m \cup V_g$ is a set of nodes representing the business objects in an enterprise, where, V_n specifies the finite set of entities, V_u is the finite set of attributes, V_a corresponds to the finite set of activities and V_e is the finite set of events, V_m is the finite set of measurements and V_g is the finite set of goals; $E = E_r \cup E_a \cup E_c$ is a set of edges specifying the relationships between the business objects, where, $E_r \in (V_n \times V_n) \cup (V_n \times V_u)$ is the directed edges specifying the general relationships between the business entities or between the business entity and its conrresponding attributes. For any two objects $v_1 \in V_n$, $v_2 \in V_n \cup V_u$, then $(v_1, v_2) \in E_r$ indicates that v_1 has a relationship with v_2; $E_a \in (Ve \times Va) \cup (V_n \times V_a) \cup (V_u \times V_a)$ is the directed edges specifying the triggering relationships between the business events and the activities or between the business entities and the activities or between the attributes and the activities. For any two objects $v_1 \in V_e \cup V_n \cup V_u$, $v_2 \in V_a$, then $(v_1, v_2) \in E_a$ indicates that the occurrence or the state of v_1 composites the prerequisite of the execution of v_2; $E_d \in (V_u \times V_u) \cup (V_u \times V_m)$ is the directed edges specifying the relationships between the business attributes or between the atributes and the measurements. For any two concepts $v_1 \in V_u$, $v_2 \in V_u \cup V_m$, then $(v_1, v_2) \in E_d$ indicates that the state or value of v_2 can be generated from the state or value of v_1

Definition 2: Business constraints are defined as $C = \{c_r\} \cup \{c_a\} \cup \{c_d\}$, where, $c_r = f_r(E_r)$ is a restriction constraint used to assert the instantiated relationships between entities or between entities and attributes. The warning message or guidelines should be aroused when the assertion results are false; $c_a = f_a(E_a)$ is the action control constraint used to govern the activity by following the guarding conditions and the triggering events. The conditions expressed as the state of the associated attributes or the entities should first be evaluated. The activity then is triggered for the execution when the evaluation results are true and the event occurs; $c_d = f_d(E_d)$ is a derivation constraint used to generate new knowledge via the relationships of computation and inference between the attributes or between the attributes and the measurements.

A set of formal templates: More elaborately, restriction constraints are classified into five types of relationships including association, magnitude, range, comparison and subsume, where, the association describes the associative relationships between two entities; magnitude represents the allowed amount of the entities; range refers to the value scope of the entity or its attributes; comparison is that the value of the attribute of an entity is compared with that of another entity or some value; subsume specifies the containment relationship between two entities. The templates of the five types are defined as follows:

Association: Template "<entity1> must|should have <at least|at most|exactly> <n> < entity2>" specifies that entity1 must or should have a minimum, maximum, or exactly set of n instances of entity2. For example, a business policy of "Order must be signed by the manager" is expressed as a business constraint of "Order must have at least one manager".

Magnitude: Template "<entity1> must|should have <at least|at most|exactly> <n>" specifies that entity1 must or should have a minimum, maximum, or exactly n instances. For example, a business provision of "Not greater than 20 students" is described as a business constraint of "Student must have at most 20".

Range: Template "<entity1> <attributes> must|should be <only one in|any one in|not in> <value list>" specifies that the attributes of entity1 must or should be only one in, any one in, or not in the value list which has been designed in advance. For example, a business convention of "Commodity belongs to three categories including food, fruit and merchandise" is expressed as a business constraint of "Commodity must be any one in the set of food, fruit and merchandise".

Comparison: Template "<entity1> <attributes> must/should be <greater|less than|no less than|no greater than|equal|not equal> (<entity2><attribute>|<value>)" specifies that the attributes of entity1 must or should be greater, less than, no less than, no greater than, equal or not equal to the attributes of entity2 or some value of the same type. For example, a business policy of "The age of the manager is greater than 19" is expressed as a business constraint of "Manger age should be greater than 19".

Subsume: Template "<entity1> must|should be a <super-type| child> of <entity2>" specifies that entity1 and entity2 must or should be in the relationship of inheritance. Or they must or should be in the relationship of containment of the same type. For example, a business regulation of "School supermarket manager should be a student" is expressed as a business constraint of "Student should be a super-type of the school supermarket manager", which means that the object of the manager inherit the object of the student.

Action control constraint governs the business activity by the triggering event and evaluating the guarding conditions. It describes the relationship between the business activity and its associated preconditions. The template has two parts including preconditions and atomic activity. It denotes that the activity is triggered when the precondition is satisfied.

Template "IF <condition> DO <activity>" specifies that the activity is executed if the condition is true, where, condition is the set of logical or relational expressions; the activity denotes the name of the task. For example, the guideline of "To do sampling when the specification, quantity and expiry date are qualified" is expressed as a business constraint of "IF the specification, quantity and expiry date are qualified DO sampling".

Derivation is used to specify the relationship between the attributes or between the attributes and the measurements. It generates a new value or state mainly by the relationship of inference and computation. There are two derivation constraint templates including inference template and computation template.

The computation template is expressed as "<result> = <expressions>". For example, a sampling rate computation formula is expressed as a business constraint of "sampling rate = sample amount/total amount".

The inference template is expressed as "IF <condition> THEN <conclusion>". For example, a business policy of "the VIP customer has 20% discount" is expressed as a business constraint of "IF a customer is a VIP THEN it has a 20% discount".

Verification algorithm: Two criteria including completeness and validness have been proposed. The former is to ensure the business constraints have been discovered thoroughly and the latter is to ensure the business constraints are valuable against the business goals or the user problems. In order to verify the business constraints an extended meta-model is proposed. The definitions are given as follows:

Definition 3: An extended meta-model is defined as a directed graph $G' = (V', E')$, where, $V' = V \cup V_c$ is the union of the set of business objects in the meta-model and the set of the business constraints; $E' = \{(V_c, V)\} \cup \{(V, V_c)\}$ is the set of edges specifying the relationships between the objects and the associated constraints who restrict them, govern them or derive the values of them. For $\forall x \in V$, $N^+(x) = \{y|(x, y) \in E\}$ is the set of nodes leading from the node x and is called the output nodes of x; whereas, for $\forall x \in V$, $N^-(x) = \{y|(y, x) \in E\}$ is the set of nodes leading to the node x and is called the input nodes of x. There is a path between any two nodes $v_1 \in V$ and $v_2 \in V$ when there exits such an ordered sequence of nodes $(v_1, \ldots, v_i, \ldots, v_2)$, $v_i \in V$.

Definition 4: Completeness criterion: there is a path between any two constraints.

Definition 5: Validness criterion: there is a path between any constraint and the global business goal.

A simple verification algorithm (Sadiq *et al*., 2005) is modified to extend the verification of the constraints to all the nodes in the model of the G'. It is described as follows:

Verification Algorithm

```
Check := model nodes
If Check = {} then Return (''No Constraint Violation'')
For i = 1 to Count (Check)
    For j = 1 to Count (Check)
        If InPath(Check[i], Check[j]) <> TRUE
            Return-Error (''Completeness Violated'')
        End-If
        If InPath(Check[i], Node(goal))<>TRUE
            Return-Error (''Validness Violated'')
        End-If
    End-For
End-For
Return (''No Constraint Violation'')
```

Business constraints interactions: There are 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 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 $\in$ C; AND-Split is expressed as a$\rightarrow$b $\wedge$ c denoting that the business constraint a impacts constraint b and c synchronously, where, a, b and c $\in$ 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 $\in$ 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 $\in$ C.

These interactions form the dynamic business behavior of the enterprise from the the view of the business constraints. Combined with the statistics they generate the solutions to the user problem and aid for the business decision-making.

Elicitation procedure: The elicitation procedure begins with collecting the enterprise provisions, interviewing the domain experts and listing out the information systems. The collected documents, interview notes and source codes should be documented and pooled. The business objects including goals, measurements, entities, attributes, events and activities should be identified and marked out from the above collected information. The models should be constructed against the meta-model from the identified business objects. More particularly, the relationships between entities, attributes, activities and measurements should be examined carefully and refined into business constraints in formal templates. The business constraints should be verified to ensure completeness and validness fianlly, their interactions should be analyzed and documented. The detailed steps are defined as follows:

Elicitation Steps

Step 1. Collecting the information
Step 2. Identifying the business objects
Step 3. Building up the models
Step 4. Tracing the relationships
Step 5. Refining the business constraints
Step 6. Verifying the business constraints
Step 7. Analyzing the interactions among the business constraints

AN EXAMPLE

The school supermarket manager wants to make a decision on the supplying strategy of the Taiping biscuit which is one kind of biscuits in their supermarket. The business analyst discovers a set of business constraints which can contribute to the decision-making. He follows the elicitation steps.

Step 1. Collecting the information: He collects the sales report from January to July and interviews with the domain experts for some useful theories such as the theory of ABC Classification (Pareto's Law, 2005).

Step 2. Identifying the business elements: A set of business objects are collected. In this example the business entity is report; the set of attributes include quantity, sales, quantity of Taiping biscuit, sales of Taiping biscuit; and the set of measurements include sales percent, occupation, category and supplying strategy.

Step 3. Building up the models: Figure 1 shows the business objects identified in step 2 and their relationships which form only part of a complete model. In the diagram the rectangle represents the entity, the round rectangle specifies the attribute and the indented rectangle designates the measurement.

Step 4. Tracing the relationships: A computation relationship exists among the sales, sales of Taiping biscuit and the measurement of sales percent. There is a computation relationship exists among the quantity, quantity of Taiping biscuit and the occupation. There are inference relationships among the sales, occupation and category. And there are inference relationships among the category and the supplying strategy.

Step 5. Refining the business constraints

CID 1: IF the commodities have 10% occupation and 70% sales THEN they belong to "A" category;
CID 2: IF the commodities have 20% occupation and 20% sales THEN they belong to "B" category;
CID 3: IF the commodities should have 70% occupation and 10% sales THEN they belong to "C" category;
CID 4: The occupation percent = quantity/total quantity;
CID 5: The sales percent = sales/total sales;
CID 6: Sales = price * sold quantity;
CID 7: IF the commodities belong to "A" category THEN they adopt the strategy of "supply periodically";

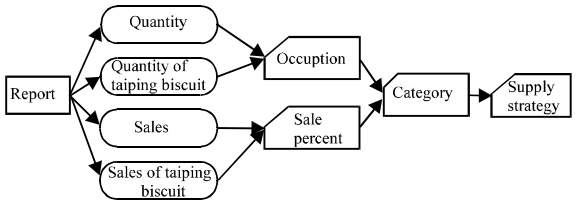

Fig. 1: Part of the model

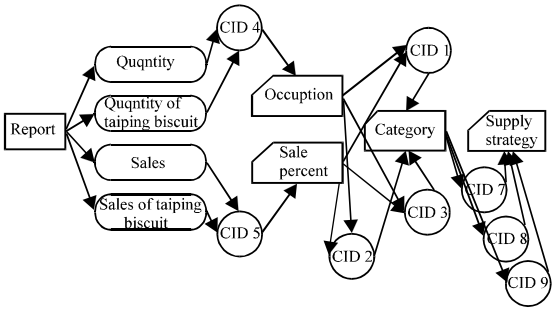

Fig. 2: Part of the extended model

CID 8: IF the commodities belong to "B" category THEN they adopt the strategy of "supply with rations";
CID 9: IF the commodities belong to "C" category THEN they adopt the strategy of "supply with mixed methods";

Step 6. Verifying the business constraints: The completeness and validness should be verified by inserting the business constraints into the model as illustrated in Fig. 2 against the extended meta-model. Obviously, the completeness and validness have been ensured for there are path between any two constraints or between constraint and the measurement of supply strategy which can be seen as an end node in the model.

Step 7. Analyzing the interactions among the business constraints: Their relationships are of CID 6 → (CID 5 ∧ CID 4) → (CID 1 ⊕ CID 2 ⊕ CID 3) → (CID 7 ⊕ CID 8 ⊕ CID 9).

Finally, the sales report document gives the real statistics of the total quantity, total sales, the quantity and the sales of the Taiping biscuit. The final conclusion easily reached by the supermarket manager is that the supply strategy of Taiping biscuit is to supply with rations based on the explicitly documented business constraints and the interactions given in step 6.

CONCLUSIONS

Business constraints are the implicit knowledge in the enterprise. These constraints provide a valuable asset for business decision-making as well as performance improvement when discovered thoroughly and documented explicitly. They protect the data integrity from the relationship view and are oriented both for the business users and the technical users. They facilitate communication among all kinds of stakeholders of the business and provide a basis for sharing and reuse.

The proposed discovery approach has a meta-model as a formal basis, specifies a set of formal templates to describe the business constraints tersely and concisely, proposes a verification algorithm to ensure the validness and completeness of the business constraints, discovers the interactions and builds up an elicitation procedure to guide the business user to uncover the business constraints from the enterprise documents. This approach has been applied in several scenarios including supermarket, meta-retrieval software and EU-Rent virtual company etc. It proves to be useful and practical.

The future research plans to propose the what-if scenario analysis and change management mechanism. And the software tool is in development to support the management of business constraints automatically.

ACKNOWLEDGMENT

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

REFERENCES

Fair Isaac, 2003. Fair Isaac Blaze Advisor: How it Works, www.fairisaac.com/rules.

Fu, G. *et al*., 2002. Representing Constraint Business Rules Extracted from Legacy Systems. DEXA 2002, LNCS 2453, pp: 464-473.

Fu, G. *et al*., 2004. Algorithms for analyzing related constraint business rules. Data and Knowledge Engineering, 50: 215-240.

Halle, V.B., 2002. Business Rules Applied: Building Better Systems Using the Business Rules Approach.Wiley Computer Publishing.

Huang, H. *et al*., 1996. Business rule extraction from legacy code, in Proc. of IEEE 20th Computer Software and Applications Conference (COMPSAC 96), pp:162-167.

Kardasis, P. and P. Loucopoulos, 2004. Expressing and organizing business rules. Information and Software Technol., 46: 701-718.

Mayer, J. R., K.M. Painter and M. Lingineni, 1996. Toward a Method for Business Constraint Discovery (IDEF9) Version 1.0.

Pareto's Law, 2005. http://346learn.ie.tsinghua.edu.cn/school/10001/door/ie-xcd03.htm.

Rosca, D., S. Greenspan and C. Wild, 2002. Enterprise Modeling and Decision-Support for Automating the Business Rules Lifecycle. Automated Software Engineering, 9: 361-404.

Rosca, D. and C. Wild, 2002. Towards a flexible deployment of business rules. Expert Systems with Applications, 23: 385-394.

Rosemann, M. and P. Green, 2002. Developing a meta model for the Bunge-Wand-Weber ontological constructs. Information Systems, 27: 75-91.

Sadiq, W.S., E.M. Orlowskaa and W. Sadiq, 2005. Specification and validation of process constraints for flexible workflows. Inform. Sys., 30: 349-378.

Sneed, M.H. and K. Erdos, 1996. Extracting business rules from source code. In Proceeding of IEEE 4th International Workshop on Program Comperension (IWPC 96), pp: 240-247.

Wang, X. *et al*., 2004. Business rules extraction from large legacy systems. In Proceeding of IEEE 8th European conference on Software Maintenance and Reengineering.