A Formulation of the Iterative Process Prototyping Methodology

TungLun Tsai[1)], Ryo Sato[2)]

[1)] University of Tsukuba, Doctoral Program in Policy and Planning Sciences (tsai@sk.tsukuba.ac.jp)
[2)] University of Tsukuba, Institute of Policy and Planning Sciences (rsato@sk.tsukuba.ac.jp)

Abstract

The SAP R/3 System is a general-purpose database management system (DBMS) application that is designed for enterprise to meet most of their business requirements. The process of fitting the business requirement into the R/3 is called customization and can be done by setting parameters. The customization in the R/3 can be very much complicated. Iterative Process Prototyping methodology (IPP) was proposed to provide an effective way for customization. The IPP is designed on the basis of six components of the R/3 and their mutual relationships. Since the IPP was proposed as a loose framework, the knowledge of technical correspondence among these components is needed when apply the IPP methodology. This paper clarifies the components and relationships among them as an ERP Consultant Model. This formulation shows that the IPP needs four components. They are Enterprise Model, Business Process Model, R/3 Prototyping, and Business Object Model. Furthermore, in order to show the advantage of the ERP Consultant Model, a customization example is provided.

1. Introduction

The SAP R/3 System is a DBMS application with thousands of database tables and 4GL programs. It is developed by SAP AG since 1979 and can support most of the business requirements [5]. The R/3 can be customized into an enterprise by setting the parameters that are related to the business requirements without any programming. The SAP AG takes responsibility of maintaining the R/3 and continually developing new functions for applying to new business requirements.

Keller and Teufel (1998) proposed Iterative Process Prototyping methodology (IPP) to implement the business requirements of an enterprise with the R/3. The IPP methodology contains six components, which are divided into business level and system level, and thirty links among the components. All of the components are quoted from the R/3. Each link can guides customization. The IPP methodology is used to analyze business process of an enterprise [1]. The main problem of the IPP methodology is that most of the thirty links are shown by example, and then ambiguous and implicit.

In order to remedy this situation, we propose an ERP Consultant Model in this paper as a basis to formulate the IPP. The ERP Consultant Model uses a object-oriented Unified Modeling Language (UML) [4] to define its four components. They are R/3 Prototyping, Business Object Model, Business Process Model, and Enterprise Model. We then propose a customization procedure that follows the formulated IPP methodology. Some services will be defined in the procedure by using SQL as a component jump to solve the problem occurred in the customization.

2. SAP R/3 System and Iterative Process Prototyping

The R/3 is the integrated software, known as the enterprise resource planning (ERP) application, that are designed to fit the various business requirements. From software-oriented view, the R/3 distributes those business requirements to three hierarchy layers. Presentation layer contains software components to form a graphical user interface (GUI), which serves as an interface for the R/3 to input data from its user. The gathered data will be sent to application layer, which contains a set of running programs, for processing. Database layer, which manages and restores the data used in the R/3, provides the programs of the application layer with various data accessing services [5].

Keller and Teufel (1998) proposed an Iterative Process Prototyping methodology (IPP) to implement feasible

business processes [1] in the R/3. The IPP methodology adopts six components from the R/3 as a basis to for the customization. Reference Process Model is a model for describing the processes used in the R/3. Organization Model presents a hierarchical view of the R/3 organization units. Data/Object Model provides a view of the data objects used in the R/3. The three components are located in business level of the IPP to give a conceptual view of the R/3. The other three components are categorized into system level for they are related to system implementation. Prototyping is the set of programs related to business process while Customizing related to parameter setting process. Data dictionary maintains the data structure that is used in the Prototyping. The mutual link among components provides guidance to the customization. The components and links are shown in Fig. 1.

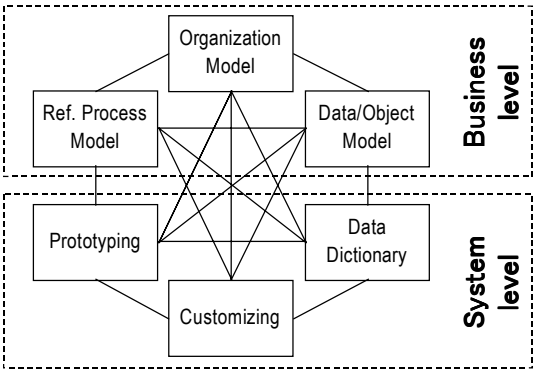

Fig. 1 IPP components and links

The IPP methodology demonstrates a customization procedure. The procedure is applied to each business area, such as "Sale", or "Production", in which processes (in the leaf, it is called a function) are hierarchically organized. In each hierarchy, sequential flow of the processes is graphed by event process chain (EPC) [1]. Components of process/function, event, link, and logical operator are arranged in a logical order to form an EPC graph. The IPP methodology is to check each process in the EPC to see if the process is necessary. In Fig. 2, for example, exactly one of "Event 1" and "Event 2" must occur to trigger "Process 3". The unnecessary processes is marked and there is no need to further explore into its detail EPC graph, such as "Process 2" in Fig. 2. For each necessary process in the EPC graph, the following guidance can be applied.

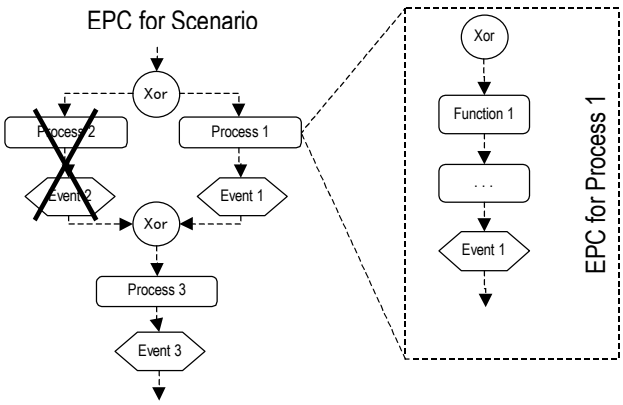

Fig. 2 EPC hierarchy and the unnecessary process

1. Link from Reference Process Model to Prototyping: To understand conceptual EPC graph, each process might be able to jump to corespondent screen of executable program. The R/3 provided a virtual company example, International Demonstration and Education System (IDES), to support the Prototyping.

2. Link from Prototyping to Organization Model: The related organization structure in Organization Model can be shown to help the understanding of the Prototyping. The IDES have provided a set of organization units in the organization structure.

3. Link from Organization Model to Customizing: If it is necessary to customize the ready-made organization structure in the Organization Model, we can jump to Customizing to add or delete organization units.

4. Link from Reference Process Model to Customizing: Since the R/3 is customized by parameter setting, to run the processes properly, some parameters, such as “customer type”, are necessary to be set in the Customizing.

5. Link from Prototyping to Reference Process Model: The R/3 is highly integrated, so it is quite often for the running program in the Prototyping to reference the must field which is maintained in other programs. Hence, we must jump to the EPC graph, say "customer master processing", that describe execution sequence of the other program. Then the “customer master processing” will be set to execute previously than the “inquiry processing”.

3. ERP Consultant Model

Because of loose structure among IPP components, there must be an R/3 expert to remedy this gap [1]. It is possible for the different expert to have different result. Furthermore, when the enterprise changed its business process in future, it seems to be difficult to reflect the changes in the related business process. Therefore, using formulated IPP in customization becomes critical success factor for the enterprise start to use the SAP R/3 System.

According to the report of using IPP, most of the links and components of IPP are seldom used, such as data model (0.5%), data dictionary (0.5%), and organization model (4%) [1]. We propose ERP Consultant Model as a new structure for IPP methodology. The ERP Consultant Model is a formulation of IPP and consists of Enterprise Model, Business Process Model, Business Object Model and R/3 Prototyping. (Fig. 3) The R/3 prototyping is a description of user-interface part of the R/3, while the Business Object Model is that of the data dictionary of the R/3. The Business Process Model uses output of R/3 Prototyping to further condense the business process into the enterprise-specified business process to fulfill enterprise requirements that are described in the Enterprise Model. In the following subsections, we show their respective structure.

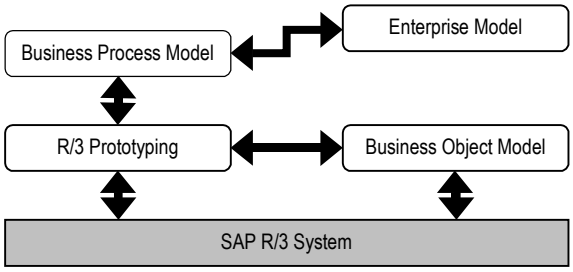

Fig. 3 Structure of ERP Consultant Model

3.1. Business Object Model

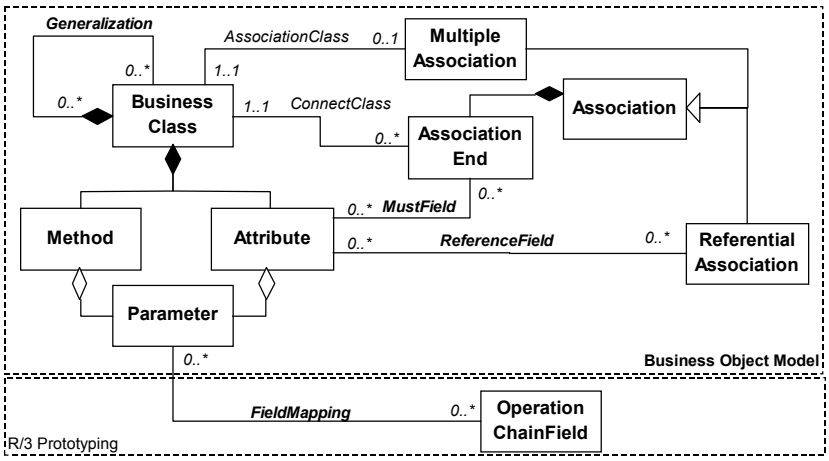

Fig. 4 Meta-model of Business Object Model

The Business Object Model is to model the instances (entities) that are used in the business process. The goal of

Business Object Model is to provide an UML description of the data dictionary of the R/3. Fig. 4 shows the UML model of Business Object Model. A rectangle represents a class, while a line between the classes is an association. A line with a triangle is a “generalization”. A line with a black diamond-shape is called “composite part-of ” association and with the white one is "aggregate part-of" association.

(1) BusinessClass / Attribute

A business class is the description of a set of objects that share the same attributes, methods and associations. Each business class is mapping to one or groups of tables in the data dictionary. An attribute is a named slot within business class that describes a range of values that instances of the business class may hold. The attribute is mapping to "table field" of data dictionary. The attribute has its data type, which is either a primitive type or a user-defined class. In order to uniquely identify each instance of a business class, some attributes are marked as key attributes. In the meta-model, ***BusinessClass*** is a composite of ***Attribute***.

(2) Method / Parameter

The method describes behavior of the business class. Parameters are input fields of the method. The R/3 is an DBMS-based application, so the type of the method is limited to "new", "update" and "delete" and these parameters are the attributes of the business class. In the meta-model, the ***BusinessClass*** is a composite of ***Method***. Both the ***Method*** and the ***Attribute*** are aggregations of the ***Parameter***. The ***Parameter*** has an ***FieldMapping*** association with ***OperationChainField*** in the R/3 Prototyping.

(3) Association / AssociationEnd

An association defines a semantic relationship among the business classes. An association end is the endpoint of the association that connects to the business class. Each association end is part of one association and contains a multiplicity and a role name attributes. If for all instances in a class, there exists one or zero instance of another class for reference, then there is a referential association between the two classes. If the part class can be aggregated to the aggregate class and it is also contained in other aggregate classes, then there is an aggregation association between the part and the aggregate classes. If the part class is strongly owned by the aggregate class, i.e. the part class may not be part of any other classes, then there exists a composite association. If the multiplicities of both association end are more than one, then the association is called multiple association. All of the association types are shown in Fig. 5.

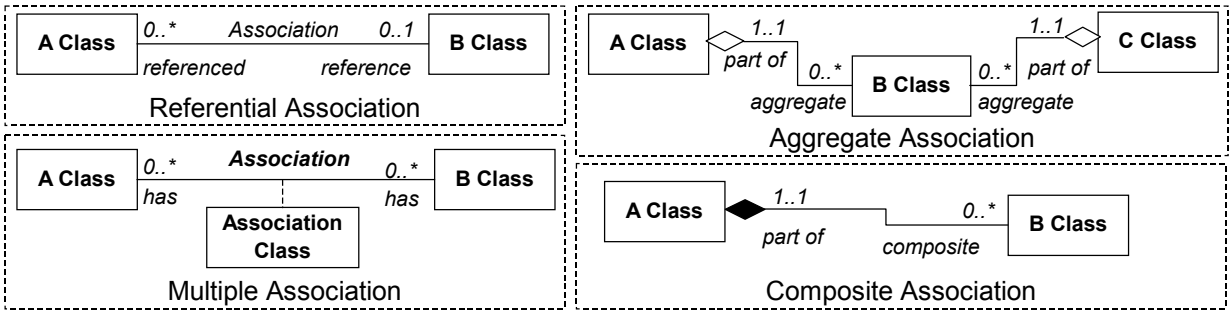

Fig. 5 Association type and their representation

In the meta-model, ***AssociationEnd*** is composite part of ***Association*** and each instance of the ***AssociationEnd*** can connect to one instance of the ***BusinessClass***. A multiple association can be implemented as a business class, thus there is an *AssociationClass*, referential association linked from ***MultipleAssociation*** to the ***BusinessClass***. A referential association can be implemented as attributes in reference side business class, hence ***ReferenceField*** association class is a implementation of referential association. ***MustField***, a multiple association between the ***Attribute*** and the ***AssociationEnd***, is to record the necessary referencing attributes in referenced class for a referential association.

(4) Generalization

A generalization is a taxonomic relationship between a more general class and a more specific class. The more specific class is fully consistent with the more general element and may contain additional attributes, methods, or associations. The specific class can inherit attributes, methods and associations of its general class. Fig. 4 shows that “ Association” class is a generalization of the “Referential Association” and the “Multiple Association” class. In the meta-model, ***Generalization*** is a multiple association between a general ***BusinessClass*** and a specific ***BusinessClass.***

3.2. R/3 Prototyping

When considering coding efficiency, similar or related programs are often integrated to form a program unit. Likewise, some similar user interfaces are coupling together to pursue operation efficiency. It could be efficient for a professional user to use such interface, but it might be difficult for an inexperienced user. In order to meet the both requirement, R/3 Prototyping records execution sequence for the coupled user interface. In addition, customization programs, implementation guide (IMG), are also abstracted in the R/3 Prototyping. Fig. 6 shows the UML description of the R/3 Prototyping.

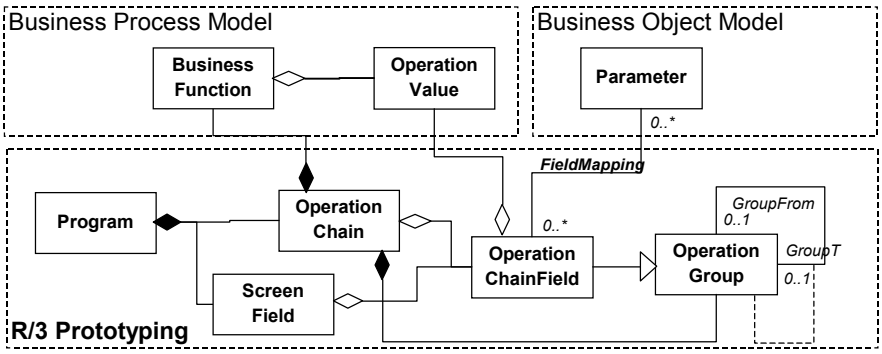

Fig. 6 Meta-model of R/3 Prototyping

(1) Program / ScreenField

A program of the R/3 is designed for the similar business processes. The program is executed and unique-identified by its transaction code. Each program contains at least one screen in its interface. After the program was started, only the initial screen is shown. A screen filed is the field that R/3 designs to communicate with user. There are two types of screen field in the R/3 Prototyping. “Data field” is the field that user can input data, whereas “Action field” is used to command the R/3 to take the next action.

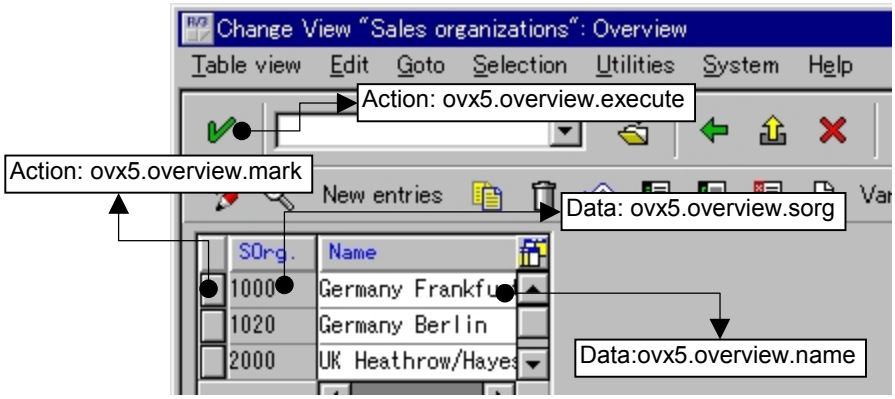

Fig. 7 Example of Screen Fields

Since a transaction code is synonymous with a program, we use it as key attribute of ***Program***. The ***ScreenField*** is the abstraction of the fields in a screen of a program. In the meta-model, the ***Program*** is a composite of ***ScreenField***. Fig. 7 shows both data fields and action fields in a screen (“overview”) of a program (“ovx5”).

(2) OperationChain / OperationChainField

A manipulation on a screen field is called an operation, such as inputting data to a textbox, pressing the “ok” button, or choosing the menu items. These operations can be logically organized to form an operation chain. The operation chain is defined as the smallest and non-dividable transaction to database. The screen field that is used in the operation chain is called operation chain field. In the meta-model, ***OperationChain***, a composite part of the ***Program***, is to declare the operation chains of the program. The ***OperationChainField***, which is a aggregate part of the ***OperationChain*** and the ***ScreenField***, is used to record the screen fields used in the operation chain.

(3) OperationGroup

An operation group is a logical sequence between two operation chain fields. Four types of logical sequence(logical

link operator) is designed in the R/3 Prototyping. “Sequential” operator defines the successive relationship between two operation chain fields and written as “→”. If two or more operation chain fields must be done before the previous or after the following operation chain field, then these operation chain fields are grouped by an “AND(∧)” link operator. If at least one of the operation chain fields must be done before the previous or after the following operation chain field, then those operation chain fields are related by an “OR(∨)” link operator. If exactly one of the operation chain fields must be done before the previous or after the following operation chain field, then those operation chain fields are grouped by an “XOR(×)” link operator. There is no link operator designed for a zero or more relationship. It can be represented by combining an “True” value with the “OR” relationship as “(True∨A∨B)”. The zero or more relationship is often seen in the input of unnecessary data fields. Fig. 8 shows graphic form of the link operators and the zero or more relationship.

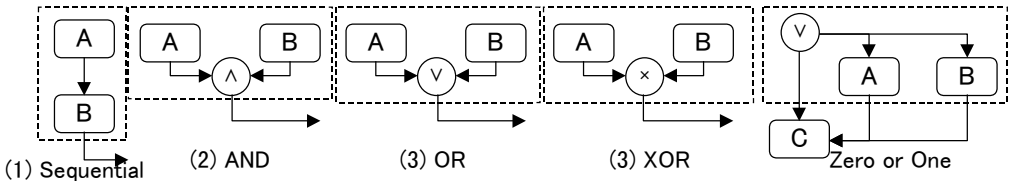

Fig. 8 Types of operation group

An operation chain field is an operation group. If A and B are operation groups, then (A→B), (A∧B), (A∨B) and (A×B) are operation groups. All operation groups are generated by applying the above rules [3]. In the meta-model, the operation chain field can be a member of operation group, accordingly the ***OperationChainField*** is a specification of ***OperationGroup***. The operation group can also be a member of another upper operation group, so the ***OperationGroup*** has a referential association with itself. An operation chain contains many operation groups to show its logical sequences, hence ***OperationGroup*** is a composite part of the ***OperationChain***. Fig. 9 shows the operation groups of operation chain “ovx5.new_record” in both graphic and symbolic form.

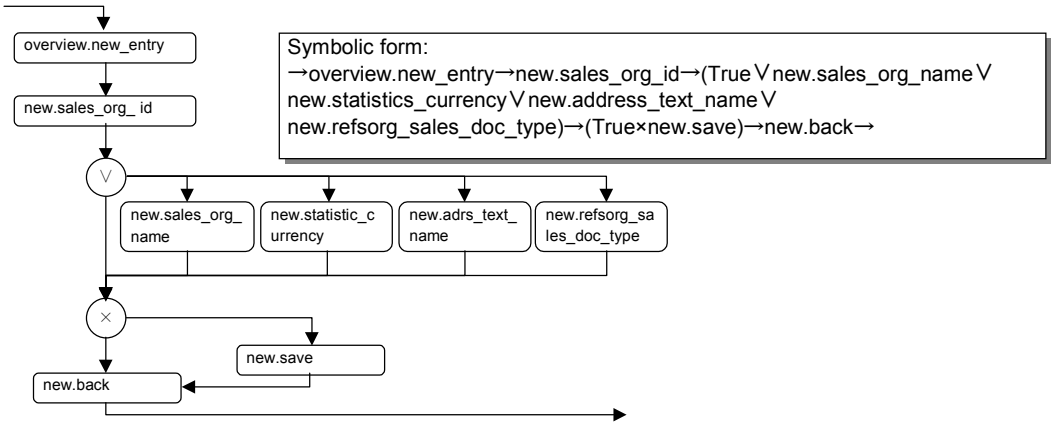

Fig. 9 Operation chain diagram and its symbolic expression

(4) FieldMapping

For the referential association exists in the Business Object Model, it is possible for a screen data field of a program to reference the value which is maintained in referenced program. For the integrality of database, we can not freely manipulate database record without the aid of the R/3 program. Therefore, there must have links between the process-oriented R/3 Prototyping and the object-oriented Business Object Model to meet the requirements above. In the meta-model, ***FieldMapping*** is an association class associates the ***Parameter*** and the ***OperationChainField***.

3.3. Business Process Model

The purpose of Business Process Model is to pick the enterprise-specified operation chain, which is defined as business function, from the R/3 Prototyping and organize sequence of the business functions, which is defined as business process, to meet the enterprise requirement in the Enterprise Model. Fig. 10 shows the meta-model of the Business Process Model.

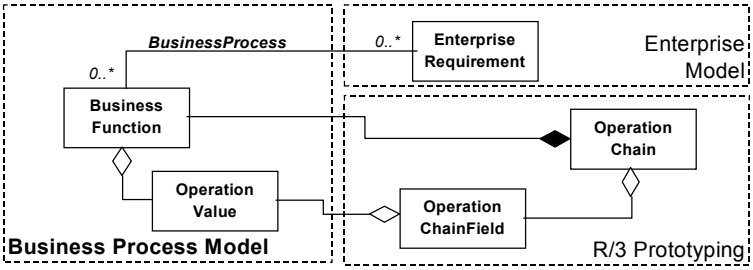

Fig. 10 Meta-model of the Business Process Model

(1) BusinessFunction / OperationValue

A business function is the enterprise-specified operation chain that can independently reference the execution of a R/3 program. The business function is defined by granting enterprise-specified value to the operation chain fields and removing the unnecessary ones under the regulation of the link operators. Fig. 11 shows the business function "create_sales_organization" is defined under the operation chain "ovx5.new_record". In the meta-model, various business function can be defined on the same operation chain with different requirement, hence ***BusinessFunction*** is a composite part of the ***OperationChain***. The enterprise-specified value will be recorded in ***OperationValue*** for each business function, hence both the ***BusinessFunction*** and the ***OperationChainField*** are the aggregates of ***OperationValue***.

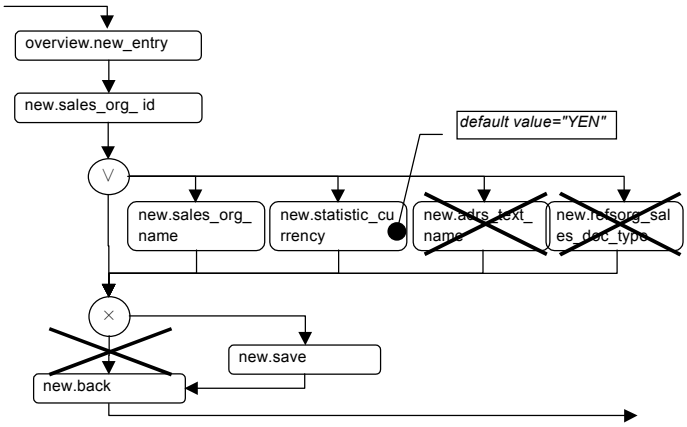

Fig. 11 Defining a business function in operation chain diagram

(2) BusinessProcess

A business process is defined by a set of linked business functions. Since each business function represents an standalone enterprise-specified executable unit, the sequence of the business functions can be freely assembled to meet the enterprise requirement specified in the Enterprise Model. Hence, in the meta-model, ***BusinessProcess*** is the multiple association between the ***BusinessFunction*** and ***EnterpriseRequirement***. Fig. 12 shows an example of "standard sales order processing" business process that is defined by composing 5 business functions.

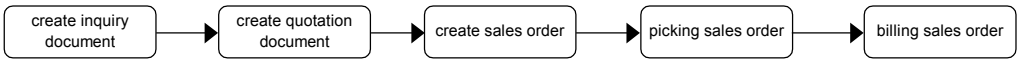

Fig. 12 Defining a business process – standard sales order processing

3.4. Enterprise Model

Enterprises are regarded as market efficient when they fully realize the potential of the market [2]. The not market-efficient enterprise will be quit from market naturally in process of competition. Suppose business environments won't change at all, then the market-efficient enterprise can be always efficient because the current way of running the enterprise, we define it as "enterprise requirements", has been proved the best fit to the environments at that time.

However, the environment is changeable and changing all the time. Enterprises try to achieve market efficiency by stressing their competitiveness when creating important business process [2]. In order to keep itself always in market

efficient status, the enterprise must figure out some policies and change itself to keep up with the changing environments. The performance of management is in proportion to how well the efficacy of policy approach to the enterprise goal. The proven right policy will be integrated into the enterprise to form a new enterprise requirement. Fig. 13 shows UML description of the Enterprise Model.

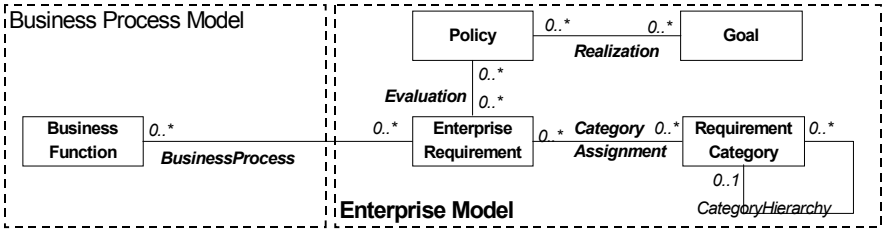

Fig. 13 Meta-model of enterprise model

(1) EnterpriseRequirement / RequirementCategory / CategoryAssignment

The enterprise requirement is a declaration of present business activity. The enterprise requirement defines "what the enterprise is now", while the business process in the Business Process Model defines "how to meet the enterprise requirements". The enterprise requirements can be classified into different requirement categories, such as "departments", "employees", or "projects". The enterprise requirements is assigned to each of the requirement category. In the meta-model, the multiple association ***CategoryAssignment*** between the ***EnterpriseRequirement*** and ***RequirementCategory*** shows that many categories can point to the same enterprise requirement and similar enterprise requirements can be assigned to the same category. ***CategoryHierarchy*** referential association from the higher ***RequirementCategory*** to the lower ***RequirementCategory*** constructs the hierarchy structure of the requirement category.

(2) Goal

The goal is the target that set by chief executive officer (CEO). Before starting to realize goal, it must be approved by the president or in the general meeting of stockholders. The goal setting is related to performance of the CEO. The president won't admit too easy goal but the performance of the CEO will be effected by setting impossible goal. Hence, the CEO can only consider proper goals for the enterprise. In the meta-model, **Goal** is to record the approved goals.

(3) Policy

In order to achieve the goal, the CEO has to lead his subordinate to make policies for the enterprise. Some policies are developed to accommodate enterprise with new events from the environments, such as the revision of law or the invention of new technology. Other policies could be the reengineering of the current business process, such as the reform of business organization. In the meta-model, ***Policy*** is the abstraction of all the policies made by the CEO.

(4) Realization / Evaluation

A good policy is the policy that meets the goal and is possible to be accomplished by the current enterprise requirement. Before the policies are to be executed, the CEO must both examine if all the goals can be realized by the policies and evaluate the impacts of the new policy toward the enterprise requirements. In the meta-model, a multiple association ***Realization***, which is between the ***Goal*** and the ***Policy***, is the abstraction of the instances that policies satisfy goals. Another multiple association ***Evaluation***, which is between the ***Policy*** and the ***EnterpriseRequirement*** records the instances that how policies influence enterprise requirements.

4. Customization of SAP R/3 System Based on ERP Consultant Model

The formulation of the IPP is based on four components and six links of the ERP Consultant Model. The structure of the formulated IPP is shown in Fig. 14. The four components are located in two distinct levels: Enterprise Model (EM) and Business Process Model (BP) are in business level while Business Object Model (BO) and R/3 Prototyping (RP) are in system level. Three of the six links, EM_BP, RP_BP, and BO_BP, are directly linked while the other links, RP_BO, EM_RP, and EM_BO are linked indirectly through the directly links. Services are provides by the links.

Customization is nothing but to build an ERP Consultant Model for the enterprises who start to use the R/3. In the following, we will show how to customize the R/3 by using formulated IPP methodology. In the procedure, SQL will be

embedded to show the implementation of some services. We assume that the instances of system level components of ERP Consultant Model have already built by SAP AG or other third party companies. The remain work of the customization is following the formulated IPP to iteratively prototype enterprise-specified business process and record the result in the Enterprise Model and Business Process Model. The following sequence of steps are recommended though every entrance point into the R/3 makes no difference by using ERP Consultant Model. ("A>>B" means jump from A to B)

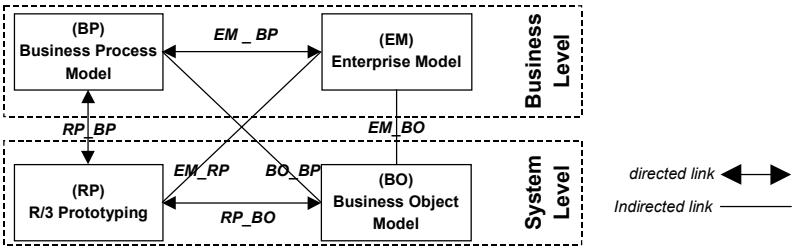

Fig. 14 IPP structure based on the ERP Consultant Model

1. Building Enterprise Requirement (EM): Before start the customization, it is better to verify and investigate the current requirement of enterprise in order not to mix the solutions provided by the R/3. The enterprise requirements can be recorded in the ***EnterpriseRequirement***. It is better to classify those enterprise requirements in the ***RequirementCategory*** to have more readability. In order not to make the classification too restrictive, one business requirement can be assigned to one or more related requirement categories in the ***CategoryAssignment***.

2. Search available business function for enterprise requirement (EM>>BP): For the defined enterprise requirements, we must find the applicable business functions to meet them in the ***BusinessProcess***. However, this service is only possible under an assumption that an IDES-like example was established to provide available business function querying service.

3. Search available operation chain for defining business function (BP>>RP): If the example does not exist, we must jump to R/3 Prototyping to search in ***OperationChain*** for possible business function. For example, an operation chain "ovxg.sales_area_assignment" can be found to satisfy "building sales organization structure" enterprise requirement. (SQL: select program, operation_chain from OperationChain where operation_chain like "sales")

4. Browse related data objects for specified operation chain (RP>>BO): It is easy to find the business object which is used in the operation chain. For example, to search the related business class for operation chain "ovxg.sales_area_assignment", the following SQL is applied. (SQL: select BusinessClass.name from FieldMapping where Program.name="ovxg" and OperationChain.name="sales_area_assignment" group by BusinessClass.name) The possible result is "tvta (sales area)". Furthermore, by looking up the ***AssociationEnd*** and ***Association*** of the business class, a association diagram related to "tvta" can be easily drawn. In Fig. 15, we know that in order to create an instance of sales area, instances of both "tvkos (sales-division)" and "tvkov (sales-distribution channel)" class must exist beforehand.

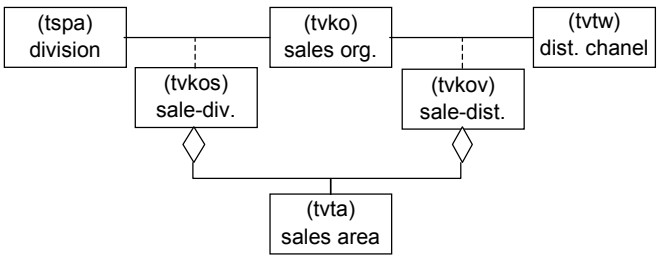

Fig. 15 Association diagram for business class "tvta"

5. Find suitable operation chain for some method (BO>>RP): If we want to new an instance for "tvkos (sales-division)", the proper operation chain in R/3 Prototyping must be searched by the following SQL (select

Program.name, OperationChain.name from FieldMapping where BusinessClass.name="tvkos" and Method.type="new" group by Program.name, OperationChain.name). Then we can choose one of the executable operation chains, say "ovx5.assignment" to create an instance for "tvkos". Following the logical operation sequence, which is composed by ***OperationGroup***, of "ovx5.assignment", we can compare its operation chain fields with physical screen fields when executing the operation chain .

6. Solve referencing problem (RP>>BO): It is possible that input value of some operation chain field, such as "new.stastics_currency", have to reference the instance of other business object "tcurc (currency master)". By querying ***FieldMapping***, ***ReferenceField***, and ***AssociationEnd,*** we can get the related association end that is connecting with the referenced business class. Furthermore, the referenced business class can be found by *ConnectClass* association. However, problem might occur if the referencing relationship continue from operation chain to operation chain. The situation might even worse if the referencing relationship becomes a circuit. Hence we need the help of Business Object Model to tell us what operation chain fields are necessary in order to complete the work in the referencing program by querying ***MustField*** and ***AssociationEnd*** with the association end found above. Then the must attributes is known and it is easy to jump back to R/3 Prototyping to search the related operation chain..

7. Define business function (RP>>BP): When traversing over the operation chain diagram, the enterprise-specified data to each operation chain field can be recorded in ***OperationValue*** under the regulation of logical operator. Fig. 11 shows the result of definition.

8. Define business process (BP>>EM): According to the enterprise requirement, the business function defined in step 7 can be grouped to form an business process by recording the business function sequences in ***BusinessProcess***. We can check ***EnterpriseRequirement*** for the enterprise requirement which is not yet satisfied by business process, i.e. the R/3. The implemented enterprise requirements are equal to the realizable capability of the enterprise. After enterprise goals is created in ***Goal***, the CEO can run the business by making the best policy (stored in ***Policy***) that is either satisfied by the goal and also can be done by the enterprise capacity.

5. Conclusion

We have proposed the ERP consultant model. It has four components and mutual links. The components are Enterprise Model, Business Process Model, Business Object Model, and R/3 Prototyping. They are not clearly defined in the original IPP methodology.

Since our formulation provides explicit structures of the components and their relationships, the difficulties in "iterative jump" of the original IPP methodology are defined as service and shown to be resolved. Therefore, we can say that the ERP Consultant Model is an effective devise for the implementation of ERP into business process. In order to show how IPP can be improved, we proposed an customization procedure which the ERP Consultant Model is employed instead of the original six components.

References

[1] Keller G., Teufel T., "SAP R/3 Process Oriented Implementation" Addison-Wesley, 1998.

[2] August-Wilhelm Scheer, "ARIS-Business Process Modeling", Springer, 1999.

[3] Chin-Liang Chang and Richard Char-Tung Lee, "Symbolic Logic and Mechanical Theorem Proving", Academic Press, 1973

[4] Grady Booch, Ivar Jacobson, and Jim Rumbaugh, "UML Semantics", Rational Software, 1997.

[5] SAP AG., SAP Online Help, SAP AG. 1998.

[6] TungLun Tsai, "A formulation of the iterative process prototyping methodology", Master thesis, University of Tsukuba, 2000.