Model-based Design of Tools for Business Understanding and Re-engineering

Stan Jarzabek and Tok Wang Ling
Department of Information Systems and Computer Science
National University of Singapore

Abstract

Tools can provide useful assistance for business re-engineering planning. Activities such as business knowledge acquisition, business process modeling, performance, quality and impact analysis all can be done more effectively if supported by proper tools. In this paper, we describe a design scenario for business understanding and re-engineering tools that is based on systematic modeling of business knowledge. The business knowledge model forms conceptual schema for the tool repository. We start by building a generic business model. As both the model and required tool characteristics vary from company to company and from one business re-engineering project to another, we customize the generic model and tools to reflect needs of a given company and a business re-engineering project in hand. We achieve a required level of tool flexibility by applying meta-CASE techniques. The physical repository schema and tools themselves are automatically generated from the customized business model specifications. In the paper, we describe components of an integrated computer-aided business understanding and re-engineering tool environment. We illustrate benefits of building tools on a rich business model, focusing on tools and analysis methods that have not been extensively described in other sources. In particular, we describe an interview assistant tool, support for impact analysis methods and an end-user query language in which a user can define new business analysis methods, not supported by a generic tool environment.

I. Introduction

No company is so perfect that cannot be changed for better. The scale and type of required changes varies from a company to company, though. While some companies require radical re-thinking of business processes [8], others may benefit from improvement [10] or innovation [6] of existing operations. In yet other companies, TQM may be all what is required. In this paper, we use term *business re-engineering* to mean the many types of operational business transformation described in the literature and applied in practice [6,7,8,9,10,20]. To control and re-engineer business one needs to thoroughly understand business operations in the context of company strategic goals, company culture and in terms of business operation efficiency measures. The premise of the work reported in this paper is that tools can ease a complex task of building up the understanding of business situation. Many such tools have been developed [7,9,20]. Most tools provide graphical editors for diagramming business processes. Some allow one to specify performance characteristics in order to simulate business behavior [20]. Yet other tools provide repository facilities and front-ends for groupware approach [7].

What is the role of tools in business understanding and re-engineering? Business re-engineering is often driven by intangible factors of an economic, sociological, organizational and psychological nature - is there a room for tools? No doubt, business re-engineering is a creative thinking process (like software or engineering design) and as such cannot be fully automated. But tools can assist business analysts in business re-engineering, just like CASE tools assist software engineers in software design. Some of the information that is collected and analyzed during business re-engineering can be expressed in machine-processable, accurate, complete and consistent form and stored in a central repository (e.g., in a relational database). Many business re-engineering planning activities can then be effectively supported by tools built around such a repository. Business re-engineering must be master-minded and conducted by humans, with tools playing the role of assistants.

Business models and tools that we build reflect this human-centered nature of the business re-engineering process. Based on business re-engineering methods and case studies described in the literature, we identified the requirements for Computer-Aided Business Understanding and Re-engineering tools, CABUR for short. The following is a list of most important tool requirements:

- support for multiple modes of *business knowledge acquisition* (such as groupware and interviews with company staff),
- some of the business characteristics should be stored in precise form, to enable automatic *simulation of process behavior*, analysis of business process performance, quality and impact analysis,
- *graphical editors* should allow analysts to manipulate business processes in the form of diagrams,
- all the business information acquired during groupware, interviews and produced during workshop sessions should be stored in the meaningful form in a *repository*,
- tools should be *integrated* through the repository, i.e., the results of one tool's action should be visible to other tools,
- a high-level *query facility* should be provided to obtain multiple business views from the repository, to answer user's questions about various characteristics of business operations and to specify non-standard types of analysis of business knowledge,
- tools should be based on a wide range of generic (i.e., company and re-engineering project independent) models and functions. At the same time, *strong customization capabilities* should be provided to make tools and repository speak the business language specific to a company and to a business re-engineering project in hand.

The above requirements are not easy to satisfy but, for practical reasons, can be hardly compromised. To meet these requirements, we need to apply advanced tool engineering solutions that integrate concepts of repository-based meta-CASE, business modeling methods, business knowledge acquisition methods, knowledge representation, performance/quality analysis and business simulation. Such tools cannot be built in an ad hoc way. They must be based on a well-defined, *comprehensive model of business knowledge*. The business model should identify information that is to be collected, analyzed and understood in course of business re-engineering planning. The representation structures to capture business knowledge in a meaningful, machine-processable form should be also provided. The business model must capture enough semantics about the business to support the many types of business analysis required in business re-engineering. The business model must maintain a good balance between formal and informal knowledge to reflect various modes of business analysts operation (informal, creative “soft” thinking and formal, analytic “hard” thinking, as explained in [7]).

Our approach to tool design is based on systematic modeling of business knowledge pertinent to business re-engineering. We build the core of the CABUR tool environment around a generic business model. We derive physical schema for the tool repository from the business model, design a generic, high-level repository query language, build generic graphical editors, knowledge acquisition and impact analysis tools. We customize CABUR tools to realities of a specific company and to a re-engineering project in hand by customizing the generic business model first (Fig. 1). The business model is expressed in terms of generic business features, their characteristics and dependencies between features. Model customization is done by adding/deleting business entities, entity characteristics and entity dependency types. Then, a meta-CASE is used to re-generate tools from the customized business model.

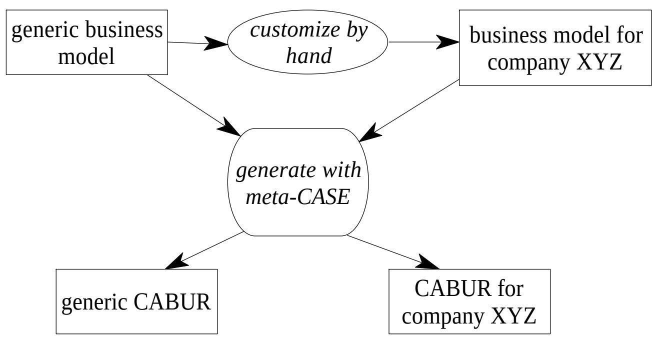

Fig. 1. Business model and tool customization

An important aspect of our approach is *separation of the internal representation of the business model from business views* that are actually presented to business analysts. While the role of the business model is to provide a comprehensive description of business knowledge, the role of CABUR tools is to provide multiple, use-friendly views of various aspects of business operations for ease of understanding. The business model can be huge, complex and partially formal, but business views presented to the user should be graphics-based, selective and simple.

In the following sections, we discuss related work, describe business modeling concepts and types of business re-engineering tasks supported by our tool prototypes.

II. **Related work**

Many sources motivate and promote business re-engineering approach, describe detail business re-engineering methods [6,8,10] and discuss tools that can support business re-engineering activities [7,9,20]. Tools exist to help analysts in flow charting business processes as well as in many types of business data analysis [20]. Dennis [7] reports on positive results with groupware approach to business modeling. Repository-based tools with front-ends supporting groupware and IDEF0 modeling method were used in reported business re-engineering experiments. APACHE [3] is a proprietary business process modeling methodology and tool. It supports data collection, data flow analysis, process flow diagrams and provides reporting facilities. Hansen [9] argues that complexity of business re-engineering efforts requires scientific, analytical techniques. Non-analytical, informal approaches lead to many failures of re-engineering projects. Business process behavior depends on many, interrelated parameters. By changing parameter values, we can modify process behavior. However, as the number of parameters grows, it becomes increasingly difficult to understand and control process behavior through parameters. Simulation and modeling can effectively reduce the complexity of this task. Tools such as CAPRE[1] [9] and **ithink** [21] are based on a precise mathematical model that enables simulation of system behavior and performance analysis. **ithink** concentrates on business dynamics and views business as a system. Basic modeling concepts in **ithink** are processes, resources, flows, feedbacks, delays and policies. While performance analysis has a good mathematical model, it may not be equally easy to formalize other business characteristics that also require understanding and tuning. Yu and Mylopoulos [22] show that by capturing goals, rules and methods we can formally reason about implications of proposed business process changes. We agree with authors of simulation and reasoning tools that a possibly precise internal information model is the first necessary step towards building successful tools. We also think that to effectively support business analysts, the underlying model should comprehensively cover many aspects of business structure and dynamics. In addition to executability, the model should record many kinds of dependencies between business entities that have to do with understanding of business operations.

[1] CAPRE is a trademark of Computer Aided Process Improvement, West Chester, OH

III. The generic business knowledge model

We refer the reader to [14] for a full description of the generic business model. In this section, for completness, we briefly explain the modeling method and the scope of the business information that we model.

We assume a business re-engineering scenario based on a popular, although not unique, approach in which assessment of currently used business processes is done prior to the design of improved business processes [6,10]. Other approaches advocate design of business processes from scratch, based on company goals, to allow for radical changes without any limitations that may be implied by the current way of doing things [8].

We apply knowledge engineering methods [1] to model the business knowledge. We model business in terms of *features*. Features represent business entities that we want to analyze during business re-engineering and about which we want to store information. Features are described by properties. For example, feature *BusinessProcess* has properties such as process name, owner, cost, work flow, efficiency and many others. Features can be related one to another using inheritance, aggregation and association. Relationships may also have properties. Types of property values include usual basic types, formal expressions, faceted classification schema [17] and informal text. In either case, each feature property identifies a piece of knowledge about business that must be collected during groupware sessions, staff interviews, modeling workshop sessions or through monitoring the ongoing business operations. A syntactic construct to describe properties of features and feature relationships is called a *frame*, the term that we borrow from knowledge representation [1]. Frame slots contain properties of features.

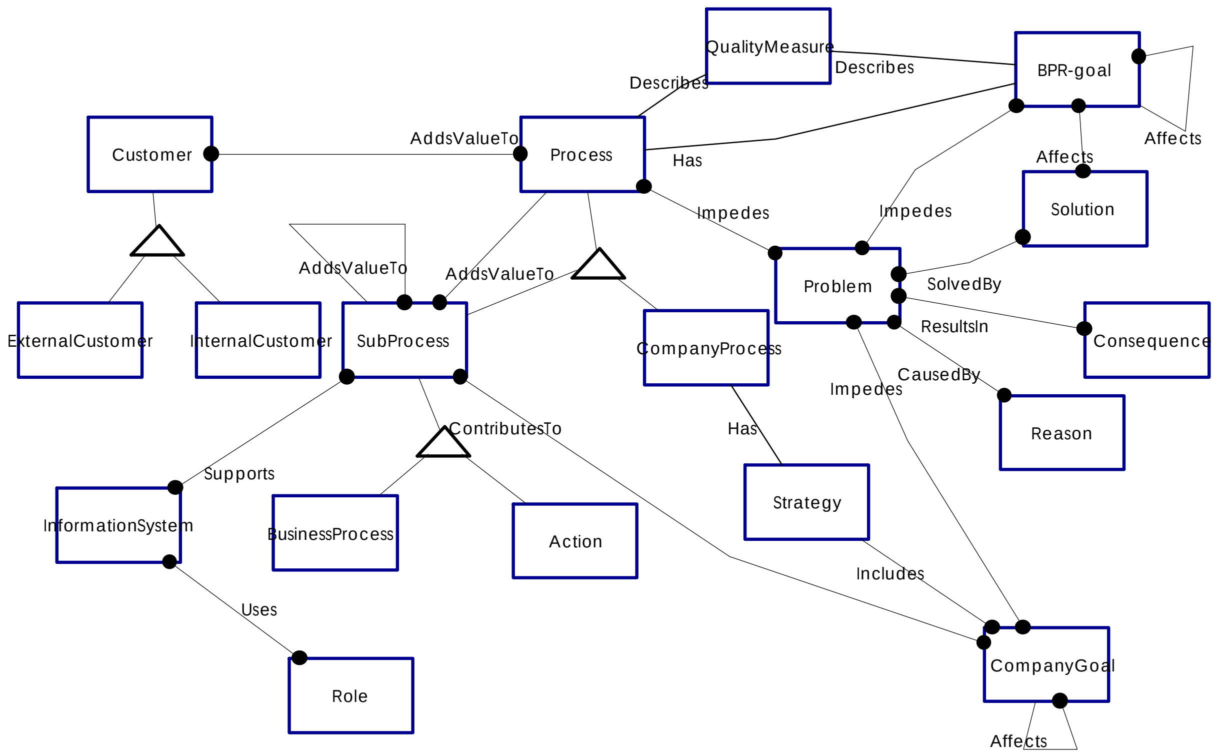

Fig. 2. Major features in the business knowledge model

Fig. 2 depicts examples of features and feature relationships that we model. We use Rumbaugh notation OMT [19] to model features and their relationships. Features are represented by rectangular boxes, triangles represent inheritance and lines between features represent binary relationships. Dots stand for 'many' connectivity in a relationship link. The meaning of a relationship link is clarified by a role name attached to a link (in italic). In the inheritance link, a parent feature appears above the triangle and derived features appear below. We believe the model of Fig. 2 is intuitive, so we only briefly comment on it. We see three types of processes derived from abstract feature *Process*, namely *CompanyProcess*, *BusinessProcess* and *Action* (feature and relationship names start with capital letters). We model company *Strategy*, *CompanyGoals* and *Problems* that impede *CompanyGoals* and *Processes*. We show how *Solutions* to *Problems* affect goals of business process re-engineering (feature *BPR-goal*). Value-added impact of *Processes* on each other and on *Customers* are modeled by a family of relationships *AddsValueTo*. (Remark: A relationship defined for feature *Process* applies to all the features derived from *Process*.) The model also shows which *BusinessProcesses* and *Actions* contribute to which *CompanyGoals*. *CompanyGoals* and *BPR-goals* are organized into a hierarchy by relationship *Affects*. Based on feature *QualityMeasure*, *Process* quality can be judged. Feature *QualityMeasure* describes the quality of current *Processes* and quality requirements to be met by re-engineered *Processes*. Target values for *Process* characteristics are set up based on company standards and world-class operation benchmarks. Some of the *BPR-goals* are expressed in terms of *QualityMeasures*. By examining *QualityMeasures*, we select candidates for business process re-engineering and evaluate the improvements of business process re-engineering to see how business process re-engineering goals are met. Features *Problem* and *BPR-goal* reflect application of business process re-engineering methods.

IV. **A tool environment for business understanding and re-engineering**

Fig. 3 depicts groups of tools in a Computer-Aided Business Understanding and Re-engineering (CABUR) environment.

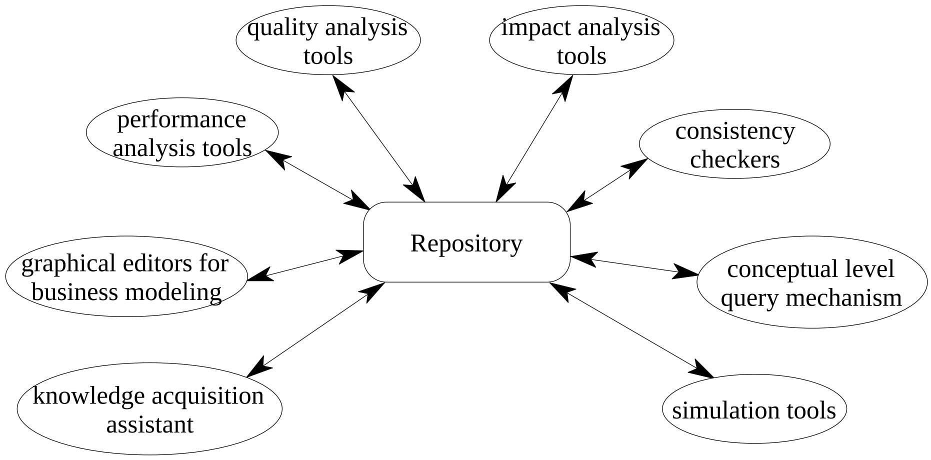

Fig. 3. Tools for business understanding and re-engineering

The prototype that we are currently building is based on the generic business knowledge model (as outlined in the last section and fully described in [14]). The objective of the prototype is to validate critical design decisions and to evaluate types of automated support for business re-engineering that, in our opinion, are important but did not get enough attention in tools implemented by others. Below, we describe a repository facility, a knowledge acquisition assistant tool, impact analysis tools and a general query mechanism for analysis of business information. We also briefly describe our graphical modeling editor which is a standard component of any CABUR tool environment.

A. **A repository**

A repository in a CABUR environment plays a similar role to the CASE repository: it is a central store for business knowledge and provides means for data integration between tools. All the information collected during groupware, interviews, modeling workshops and through monitoring of business processes is stored in the repository in the meaningful form. We derive physical schema for the repository by normalizing the business knowledge model. All the business features, their properties and relationships defined in the business knowledge model can be traced down to their physical representation in the repository. In our tool prototype, the repository is implemented in PROLOG. The advantage of PROLOG over a relational database is that schema evolution in PROLOG is easier. This simplifies modifications of the repository schema after customization of the business model to reflect needs of a specific company.

PROLOG schema can be automatically re-generated after most of the model customizations. PROLOG schema can also be extended once the system is operational without affecting the contents of the repository. We find such a flexibility of the paramount importance in design of CABUR tools. Impact analysis tools and a high-level query facility (described in section 4.4 and 4.5) can be conveniently designed on top of the PROLOG repository.

B. Knowledge acquisition tools

Dennis [7] describes tools to support groupware. We are building a knowledge acquisition tool called an interview assistant. This tool supports a business re-engineering facilitator in conducting interviews with the company staff and in feeding business knowledge acquired during question/answer sessions into the repository.

To the design the interview assistant, we studied interview materials, types of questions asked and modes of interactions between an interviewer and company staff. We derived many interview questions from the business knowledge model: for each information item in the business model, an interview question is designed to acquire that piece of information.

The interview assistant is based on the concept of a *generic question*. An example of a generic question is “how long does it take to complete a given process?”. Many *concrete questions* can be obtained from a generic question. Here are examples of concrete questions: “how long does it take to complete the business process of servicing customer orders (from accepting the customer order to delivering a product to a customer)?” and “how long does it take to complete an action of filling the customer order form?”. These examples illustrate two ways of how a generic question can be instantiated. One way is to specify an instance of a feature (the business process or action name in this case); another way is to replace a superclass feature (*Process*) by a subclass feature (*BusinessProcess* and *Action*, see feature models in Fig. 2). Of course, only concrete questions are asked during interviews. We store answers to the questions in two forms: as an informal text and in the meaningful form, in the repository. For example, if the answer to question “how long does it take to complete an action of filling the customer order form?” is “10 minutes on average” then the property ‘duration’ of the instance of feature *Action* named “fill customer order form” is assigned the proper value in the repository.

To implement the above concept, we defined templates for generic questions and answers. A template has typed slots. Slots refer to business model features, feature properties and feature relationships. For example, a generic question “how long does it take to complete a given process?” is specified as:

```
feature P IsA Process
    <P.name=?>  <P.duration=??>
```

The specification says that, to obtain concrete questions, ‘P’ can be any feature derived from *Process* (Fig. 2) and that the name of the *Process* should be given (a single question mark in the duration slot). A double question mark indicates information items that we expect to obtain from the answer, *Process* duration, in this case. We could generalize the above question in the following way:

feature P IsA Process
feature-property prop
<P.name=?> <P.prop=??>

This template can be instantiated to an inquiry about value of any property of a Process such as duration, cost, quality, etc.

The internal mechanism for handling generic questions is hidden from the user. The user interface of the interview assistant helps the interviewer to instantiate questions, to extract business information from the answer and to insert the answer into the repository. Additional questions can be formulated and added to the repertoire of generic and concrete questions during interviews, as necessary.

All the questions and answers are stored in the repository. Groups of questions relevant to different business aspects can be selected from the pull of interview questions. For example, questions related to all the actions or to specific actions (such as “filling the customer order”) or to certain feature properties, etc. can be selected. This is helpful in preparing questions for an interview and in analysis of the business knowledge. The interview assistant provides a flexible mechanism that allows one to form multiple groupings of interview questions, depending on the situation that may arise. For example, the following categories of interview questions may be of interest:

- general questions related to the company as a whole,
- questions related to identifying business processes,
- questions that lead to identifying problem areas in current business operations,
- questions related to specific business processes and actions,
- questions related to identifying solutions to problems,
- questions relevant to high managers,
- questions relevant to technical staff, etc.

We implemented a prototype interview assistant with GURU[2] system. The tool will be used on the business modeling project at the Temasek Polytechnic in Singapore.

C. **Graphical editors**

Diagrams can depict various dimensions of a company in an easy to comprehend way. For example, company structure and work flows can be depicted by suitable diagrams and supported by graphical editors. In [14], we described business views that can be obtained from the business knwoledge model and supported by our graphical modeling tools. As an example, we shall briefly describe a work flow diagram.

---

[2] GURU is a trademark of MDBS, Inc

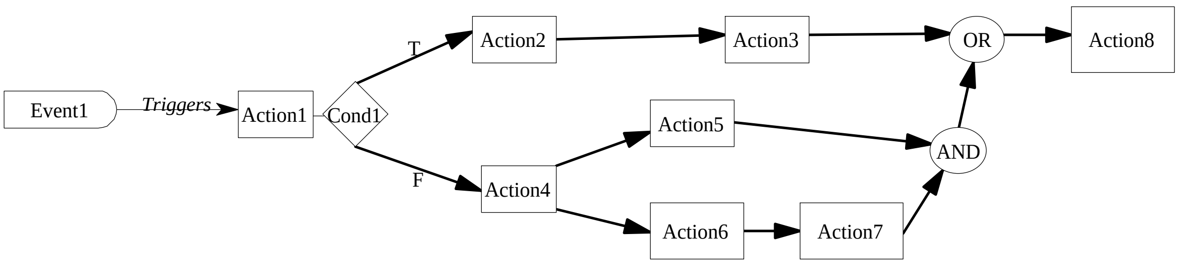

Fig. 4. Business Process flow chart

In our business model, *BusinessProcesses* are chains of *Actions* (fig. 4). Each *Action* can be further exploded into a more detail work flow chart. *Action* execution is governed by *BusinessRules*. Diamonds mark decision points at which alternative *Actions* can be taken. Issues that affect the decision are modeled as conditions. *BusinessRules* describe what happens at decision points or indicate chains of *Actions* that can be performed in parallel. Timing constraints (such as "wait for 1 hour before sending a report") are also modeled as *BusinessRules*. In Fig. 4, Event1 triggers the *BusinessProcess* (i.e., Action1 that starts the BusinessProcess) whose first level of decomposition involves eight Actions. After Action1 is completed, depending on condition 'Cond1', either Action2 or Action4 is initiated; Action3 is initiated once Action2 is completed; when Action4 is completed, Action5 may be performed in parallel with Action6 and Action7. In the diagram, small circles mark the end points of chains of alternative or concurrent *Actions*. Fat arrows depict flows of business Objects (information, documents, materials, decisions, etc.) between Actions. By clicking on icons (Actions, Events, decision points and flows) the detailed properties of a given business element can be viewed or entered. Properties of features can be modified during customization of the business model.

The semantics of diagrams is defined within the business knowledge model and diagrams provide user-friendly presentations of (part of) the business model. When, for example, a workflow diagram (such as one in Fig. 4) is used to model business processes, business knowledge is extracted from diagrams and stored in the repository.

We use MetaEdit[3] to obtain editors required in our current tool prototype. MetaEdit generates editors from non-procedural specifications of a modeling method to be supported. This is useful in the context of building a business re-engineering environment, as diagramming conventions can be easily changed as required. However, the MetaEdit tool does not support the concept of repository, so generated tools cannot be integrated. We partially overcome this limitation by building bridges between graphical editors and the PROLOG repository. As MetaEdit has a powerful report generating facility, we can extract information from diagrams and after re-formatting, feed the information into the PROLOG repository. This solution allows us to extract business knowledge from diagrams, store it in the central repository, check

[3] MetaEdit is a trademark of MetaCase Consulting Oy

the consistency of information and conduct many types of business data analysis. Unfortunately, MetaEdit editors lack the import facility. This means that we can transfer business data from editors to the repository but not in the opposite direction. So the modifications of business data cannot be made transparent across editors. As different diagrams operate on overlapping parts of the business model, lack of bio-directional data communication is a serious limitation. Nevertheless, MetaEdit allows us to experiment with many concepts of tools in a cost-effective way. Based on the prototype, we started implementation of the tool environment that will run on multiple platforms and will provide, apart from customization capabilities, bio-directional bridges between tools and the repository. This new version of tools we hope to develop with close cooperation with consultants who would be interested in using our tools on real-life re-engineering projects.

D. **Impact analysis**

Understanding the business means understanding complex relationships between business features and understanding the impact of detail characteristics of business features on the business as a whole. Business re-engineering can be viewed as modification of business features, their characteristics and their relationships to achieve quality and performance improvements at the global level. In a complex business situation, comprehending the impact of changes is a challenge. Understanding the dependencies between various aspects of business and tracing the impact of changes is an essence of business re-engineering. Therefore, we tried to explicate important types of dependencies between business features within our business knowledge model. Tools built around the model can then help business analysts to evaluate the impact of proposed modifications and assist them in selecting the optimal re-engineering solution. To illustrate what we mean, let us examine value-added assessment and a framework for evaluating alternative business re-engineering solutions.

Value-added assessment is one of the essential business re-engineering methods [18]. The business model supports value-added assessment in the following way. In Fig. 2, we have relationship *AddsValueTo* between *Processes* and *Customers*. As feature *BusinessProcess* is derived from *Process*, an instance of this relationship applies to *BusinessProcesses*. Properties of relationship *AddsValueTo* describe the overall impact of a *BusinessProcess* on *Customers*:

```
AddsValueTo (BusinessProcess, Customer)
    rating-of-added-value : INTEGER
    target-rating : INTEGER
    added-value-description : TEXT
end AddsValueTo
```

In addition to the overall value-added impact of a *BusinessProcess* on *Customers*, we may be interested in a more detail description of how a *BusinessProcess* actually affects a *Customer* (e.g., which *Customer* needs are affected by which characteristics of a *BusinessProcess*? how important is this impact for a *Customer*?). To address this, detail impact assertions can be specified at the feature instance level. The following schema describe impact of the duration of *BusinessProcess* ‘OrderProcessing’ on *Customer* ‘Buyer’ need for timely delivery of goods:

Impacts (OrderProcessing.duration, Buyer.need.timely-delivery)
    impact-rating : INTEGER
    target-rating : INTEGER
    impact-description : TEXT
end Impacts

The above schema together with relationship AddsValueTo (BusinessProcess, Customer) form a group of descriptions related to adding value to *Customer*. From Fig. 2 we also read that BusinessProcesses can add value to each other. Relationships AddsValueTo (BusinessProcess, Customer) and AddsValueTo (BusinessProcess, BusinessProcess) together with assertions form a description of total value added by a *SubProcess*. The total value-added rating is represented by a property ‘accumulated-value-added’ of a *BusinessProcess*. Value-added analysis can be also applied at the level of *Actions* that comprise a *BusinessProcess*. *Actions* that add no value can be then revised and, eventually, eliminated.

E. **A user level query mechanism for business knowledge**

Once the business knowledge base grows big (and only at this point it becomes potentially useful), it becomes more and more difficult to analyze, comprehend and make sense out of the stored knowledge. The role of tools is to display simplified, selective views of the information, focusing on a specific aspect of business structure or function that is to be analyzed and understood. A number of standard methods can be built into a CABUR tool environment. Can we, however, predict in advance all types of analysis of business information that one may want to do during business re-engineering planning? Here are examples:

- what value does a given business process add to the customer?
- which customer needs are poorly addressed by a given business process?
- what problems are responsible for poor performance of a given business process?
- what are the reasons for these problems, possible solutions and what is the expected impact of solutions on the performance of a business process?
- which staff update information in customer files?
- which processes read information from producer files?
- is information acquired from different sources consistent one with another?
- what information is missing to complete a given analysis task, e.g.,
    - ◊ to identify business processes,
    - ◊ to formulate a business process model,
    - ◊ to evaluate the quality of a business process,
    - ◊ to do comparative study between the two business re-engineering solutions.

Taking into account a generic nature of CABUR tools, the variety of companies and the variety of business re-engineering methods, probably we cannot predict all kinds of queries, views and types of data analysis that may be needed in business re-engineering projects. This realization motivated us to define an end-user level *b*usiness *q*uery *l*anguage, *BQL* for short. In *BQL*, a business analyst can interrogate the business knowledge base on his own and specify new, non-standard analysis methods. *BQL* is based on the concept of Entity-Relationship query languages [4,18] and on a program query language used in our tool environment for program understanding [13]. *BQL* queries are written in terms of the business knowledge model, such as one outlined in section 3 and fully described in [14].

In *BQL*, each business model feature represents a set of its instances. Relationship signature (for example, AddsValueTo (Process, Customer), represents a subset of pairs of feature instances that are involved in a relationship. Tuples may involve feature instances as well as property values. Here are examples:

<Action, Problem, Consequence> - a set of triples of instances of features *Action*, *Problem*, *Consequence*, respectively,

<Action, Role.name> - a set of pairs first of which is an instance of feature *Action* and second one is a property name of feature *Role*.

A *BQL* query specifies a tuple of elements to be retrieved. Elements may be feature instances or property values. Here are examples:

Select Action - this is one element tuple

Select Action.name

Select <Action, Problem, Role.name>

In most situations, we wish to select a certain subset of elements rather the whole set. We constrain properties of the target subset by specifying conditions to be satisfied by its elements. Conditions are expressed in terms of:

) feature instances, property values and constants, and

) participation of features in relationships.

The following is a format of a business query:

select-cl ::= [*name*-cl] *declaration** **Select** *result* (*with*-cl | *suchthat*-cl)*

Clauses in rectangular brackets are optional. Star '*' means repetition of 0 or more times. Stepping from left, the (optional) *name*-cl gives a name to a retrieved view The *declaration*s introduce synonyms for features (in SQL synonyms are called range variables). Synonyms can be used in the remaining part of the query to mean a corresponding feature. The *result* specifies a view to be produced. In the *with*-cl, we constrain property values (e.g., Role.name="clerk"). The *suchthat*-cl, specifies conditions in terms of relationship participation. We explain queries by examples. All the queries refer to business knowledge model such as one in Fig. 2. Here are examples of *BQL* queries:

Q1. Which problems impede action "fill-customer-order"?

Select Problem **such that** Impedes (Problem, Action)

with Action.act-name = "fill-customer-order"

Explanation: keywords are in bold. Dot '.' notation means reference to the property value of an feature. According to Fig. 2, relationship *Impedes* involves features *Problem* and *Process*. As feature *Action* is derived from *Process*, relationship *Impedes* may also involve *Action*.

Q2. Which other *BPR-goals* are affected by the goal "reduce time to market by 5 days"?

BPR-goal bg1, bg2

Select bg1 **such that** Affects (bg1, bg2)

with bg1.goal-name="reduce time to market by 5 days"

Explanation: declaration introduces two variables that represent *BPR-goals*.

Q3. Which customers are affected by the problem of "lack of communication between departments A and B", problem id# 25?

Select Customer **such that** AddsValueTo (Process, Customer) **and**

Impedes (Problem, Process) **with** Problem.id#=25

Q4. Which of the proposed business re-engineering solutions will affect customer Supplier?

Select Solution **such that** Affects (Solution, BPR-goal) **and** Has (Process, BPR-goal)

and AddsValueTo (Process, Customer) **with** Customer.cust-name= "Supplier"

Q5. Which actions are impeded by more than 5 problems?

Select Action **such that** Impedes (Problem, Action>5)

Explanation: constraint ">5" applies to the number of occurrences of an *Action* as the second argument in relationship *Impedes*. First argument is unconstrained.

Q6. Find Actions that are supported by at least one information system.

InformationSystem infoSyst

Select Action **such that exists** [infoSyst **such that** Supports (infoSyst, Action)]

Q7. Find business processes that add no value to customers

BusinessProcess bp

Customer cust

Select bp **such that not exists** [cust **such that** AddsValueTo (bp, cust)]

Q8. List actions of business process "Customer Service" along with problems that impede actions, reasons for problems and problem consequences.

BusinessProcess bp

Select <Action, Problem, Reason, Consequence> **such that** ConsistsOf (bp, Action)

with bp.bp-name="Customer Service" **such that** Impedes (Problem, Action)

and CausedBy (Problem, Reason) **and** ResultsIn (Problem, Consequence)

V. **Conclusions**

Our approach to designing tools for business understanding and re-engineering is based on systematic modeling. We started by modeling business knowledge pertinent to various business analysis tasks and re-engineering transformations. We use knowledge representation methods to build the business model. Apart from providing a reference model for business re-engineering methods, our business model forms a basis for building an integrated, repository-

-based tool environment for business understanding and re-engineering. The physical schema for the repository are derived from the business model. Tools can support a range of business re-engineering planning activities:

- business knowledge acquisition through groupware, interviews with the company staff and by measuring current business operations,
- modeling of various dimensions of company structure and operation using diagrams,
- impact analysis,
- performance, quality analysis and simulation of business processes,
- consistency checking, and
- evaluation of alternative business re-engineering solutions.

In the paper, we described a tool called an interview assistant, impact analysis and a user-level language to query the business knowledge.

The core of our tools is build around a generic business model. The generic business model is customized to reflect the realities of a given company and of a business re-engineering project in hand. After model customization, tools are re-generated from business model specifications. The tool generation approach provides the level of flexibility required in a tool environment for business understanding and re-engineering. In our prototype, we used a CASE tool generator MetaEdit to obtain graphical editors, GURU system to implement an interview assistant and PROLOG as a central repository.

We started converting our prototypes into tools that can be used on real projects. We want to seek cooperation of business re-engineering consultants in formulating further requirements for our tools and in testing tools on real projects. Tool requirements and design will evolve during the business modeling project at the Temasek Polytechnic in Singapore.

VI. Acknowledgments

This work was supported by National University of Singapore Research Grant RP920613. Thanks are due to Seow Mui Leng who designed graphical modeling tools, experimented with automatic business data analysis methods and refined the business knowledge model in many useful ways. Lau Ai Leng, Kuan Sook Peng and Catherine Tan implemented the interview assistant.

References

[1] Adeli, H. (editor) *Knowledge Engineering*, vol I, McGraw-Hill, 1990

[2] Berztiss, A. (1990) 'The Specification and Prototyping Language SF', Report No 78, SYSLAB, The Royal Institute of Technology, Sweden

[3] Born, G. “APACHE: A Pictorial CASE Tool for Business Process Engineering,” in *Software Assistance for Business Re-engineering*, Spurr, K., Layzell, P, Jennison, L. and Richards, N. (Editors), John Wiley & Sons, 1993

[4] Chan, H. “An Entity-Relationship Enhanced Logic System,” *Proc. Second Int. Symp. on Database Systems for Advanced Applications*, Tokyo, 1991, pp. 401-410

[5] Chen, P. "The Entity-Relationship model -- toward a unified view of data," *ACM Transactions on Database Systems*, vol. 1, no. 1, 1976, pp. 9-36

[6] Davenport, T.H. *Process Innovation: Reengineering Work through Information Technology*, Harvard Business School Press, Boston, Massachusetts, 1993

[7] Dennis, A., Daniels, R. Jr., Hayes, G. and Nunmaker, J. Jr. "Methodology-Driven Use of Automated Support in Business Process Re-engineering," *Journal of Management of Information Systems*, vol. 10, no. 3, Winte 1993-94, pp. 117-138

[8] Hammer, M. and Champy, J. *Re-enginereing the corporation: A manifesto for business revolution*, Nicholas Brealey Publications, 1993

[9] Hansen, G. *Automating Business process Re-engineering*, Prentice Hall, Englewood Cliffs, 1994

[10] Harrington, H. *Business Process Improvement*, McGraw-Hill, Inc., 1991

[11] Jarzabek, S. and Tan, C.L. "Modeling multiple views of common features in software re-enginereing for reuse" *Proc. 6th Int. Conference on Advanced Information Systems Engineering, CAiSE'94*, Utrecht, June 1994, *Lecture Notes in Computer Science*, no. 811, Springer-Verlag, pp 269-282

[12] Jarzabek, S. and Lim, W.M. "Modeling in strategic re-engineering," *Proc. 6th International Conference on Software Engineering and Knowledge Engineering, SEKE'94*, Jurmala, June 1994, pp. 249-256

[13] Jarzabek, S. "Systematic Design of Static Program Analyzers," *COMPSAC'94*, Taipei, November 1994, pp. 281-286

[14] Jarzabek, S. and Ling, T.W. "Knowledge model for Business Process Re-engineering tools," *Technical Report TRC3/1995*, Department of Informations Systems and Computer Science, National University of Singapore, March 1995

[15] Keen, P.G.W. *Shaping the future: Business design through information technology*, Harvard Business School Press, 1991

[16] McBrien, P. *et al* "A Rule Language to Capture and Model Business Policy Specifications," *Proc. 3rd Int. Conference on Advanced Information Systems Engineering CAiSE'91*, Trondheim, May 1991, Lecture Notes in Computer Science, no. 498, Springer-Verlag, pp. 307-318

[17] Prieto-Diaz, R. "Classification of reusable modules," *IEEE Software* 4(1), 1987, pp. 6-16

[18] Roesner, W. "DESPATH: An ER Manipulation Language," in Entity-Relationship Approach (Editor Peter Chen), IEEE CS Press, 1985, pp. 72-81

[19] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W. *Object-Oriented Modeling and Design*, Prentice-Hall, 1991

[20] Spurr, K., Layzell, P, Jennison, L. and Richards, N. (Editors) *Software Assistance for Business Re-engineering*, John Wiley & Sons, 1993

[21] Stevenson, R. "Strategic Business Process Engineering: A Systems Thinking Approach using ithink," in *Software Assistance for Business Re-engineering*, Spurr, K., Layzell, P, Jennison, L. and Richards, N. (Editors), John Wiley & Sons, 1993

[22] Yu, E.S.K. and Mylopoulos, J. "Using Goals, Rules and Methods to Support Reasoning in Buisness Process Reengineering," *Proc. 27th Annual Hawaii International Conference on System Sciences*, 1994, pp. 234-243

Appendix. Sample diagrams produced with graphical business modeling tools

We used an XYZ merchandising company described in "Structured Systems Development: Analysis, Design, Implementation" by Powers, Cheney and Crow, as a case study. We customized the generic business model described in [14] to obtain a business model for the XYZ company. Then, we used MetaEdit to generate editors for five business views derived from the customized model. One view displays the company structure and other four views display the process structure.

Fig. A1 shows the hierarchy of departments, where the “Top Level Management” oversees the “Personnel” department, “Sales” department, etc. Any department can be further broken down into smaller units (not shown here). Departments interact with each other as shown by dashed directed lines. The “Sales” department involves two roles, namely “sales assistant” and “sales assistant (backorders)” and communication between the roles.

Fig. A2 shows the company process that is divided into two business areas (not shown). Each business area can consist of several business processes. Fig. A3 depicts a part of a business process that deals with processing customer orders. In case of complex diagrams, it is useful to filter out various details in order to produce simplified views, focusing on aspects of business to be analyzed (Fig. A4)

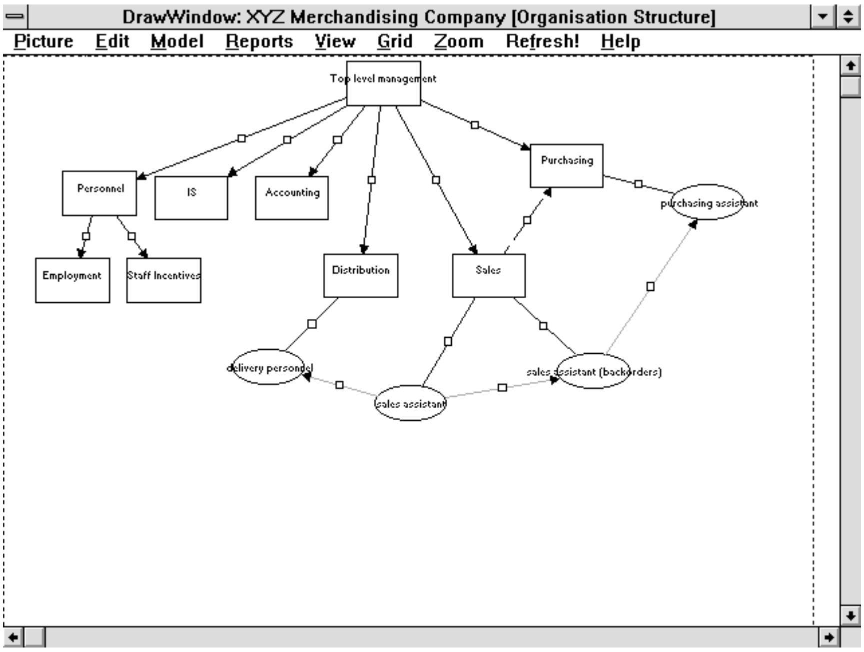

Fig. A1 Company departments and roles

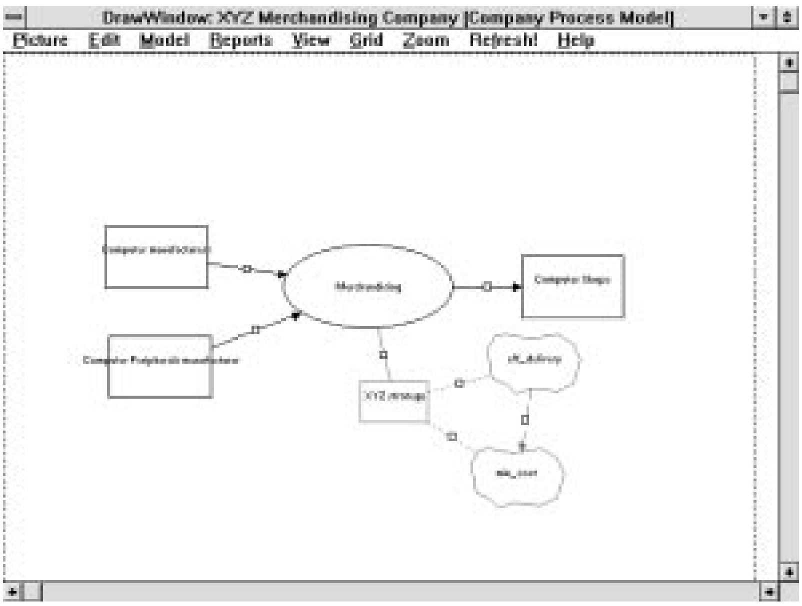

Fig. A2 The company process

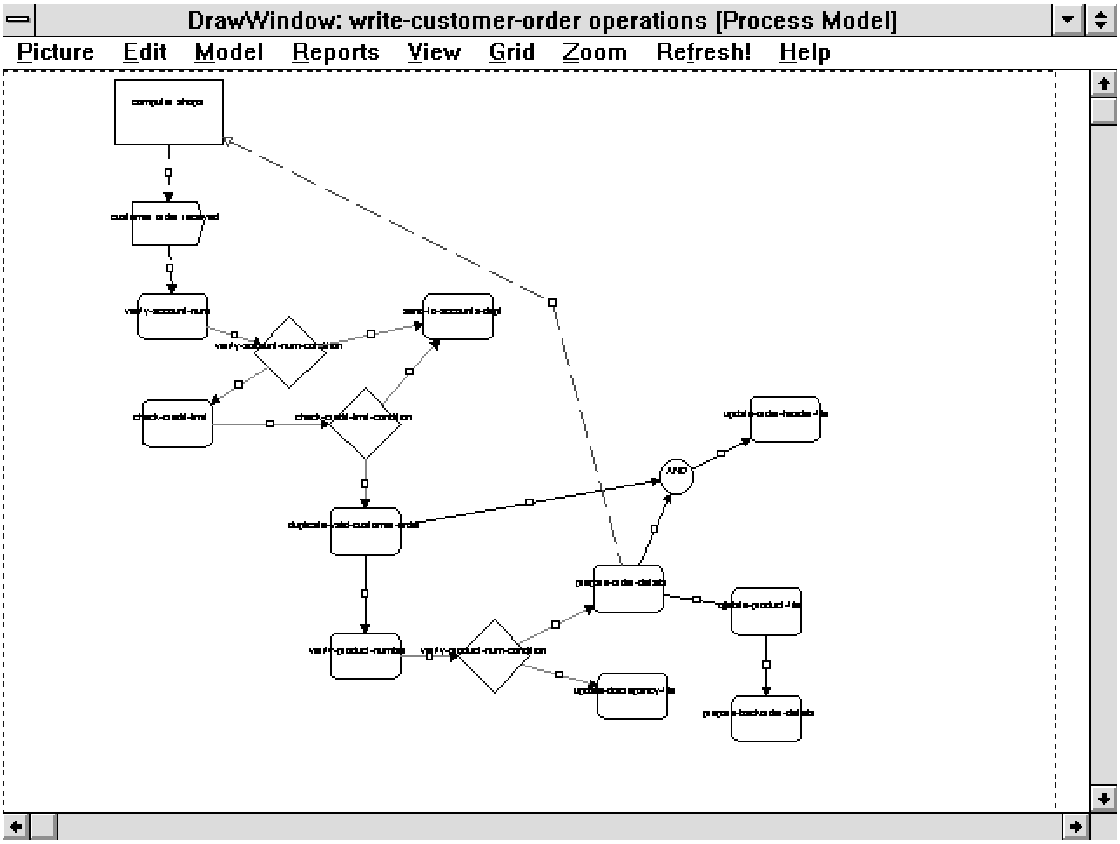

Fig. A3 A part of a customer order processing diagram

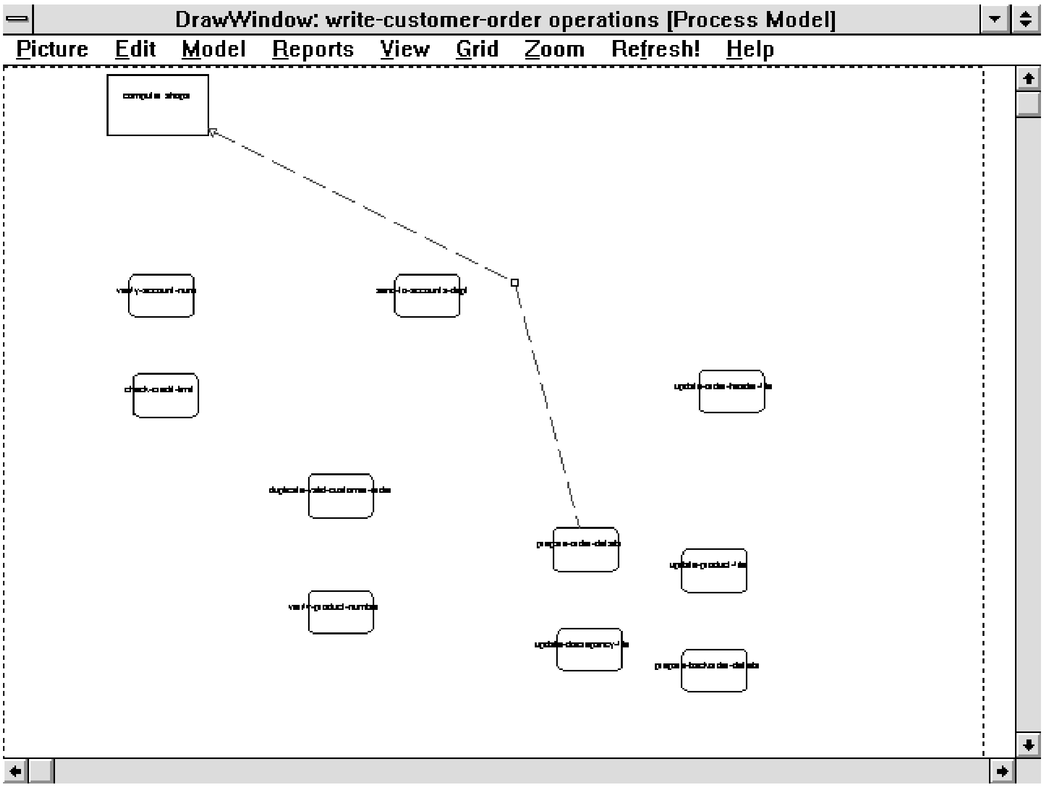

Fig. A4 A view focused on actions and values added to customers