Building Business Applications By Integrating Heterogeneous Repositories Based on Ontologies

Noriaki Izumi and **Takahira Yamaguchi**
Dept. Computer Science, Shizuoka University
3–5–1 Johoku Hamamatsu Shizuoka 432-8011 JAPAN
{izumi, yamaguti}@shizuoka.ac.jp

Abstract

This report proposes an integrated support methodology for constructing business models including employing new business models, transplanting existing business activities to computers, and decision making support in employing new environment of computers. In order to model enterprises and business activities and to implement them as software applications, heterogeneous repositories in different granularities of business models are integrated based on ontologies. By devising a framework, which picks the main concepts of repositories up and make correspondence among them, our framework achieves the unified reuse of existing repositories of business activities and software libraries. We have implemented the prototype system by JAVA and confirmed that it supports us in various phases of business application development including business model manifestation, detailed business model definition and an implementation of business software applications.

1 Introduction

Due to the rapid change of business environments introducing the Internet, it becomes very important to achieve the rapid adaptation of the corporate structure by employing a variety of heterogineus reusable componets such as business models, best practices, software libraries, and so on.

Because of the above context, a lot of research and development projects, which construct business repositories of various concepts and ideas relating to business activities, have been activated.

In the field of MS (Management Science), one of the famous results is the e-business Process Handbook project[MIT Process Handbook Project] carried out by MIT. The e-business Process Handbook is a substantial contribution as a business repository, which contains approximately 4,600 definitions of business activities from abstract processes to the specialized one to the business over the Internet. Its formality, however, is not strict since the most part of the definitions are described with natural language.

From the viewpoint of the formality on the process specification, there is the enterprise ontology[Uschold et.al. 1998] of Edinburgh University in the field of artificial intelligence. Its formality is very strong and it contributes the reuse of business models nevertheless it covers only so general and abstract concepts that it is very hard to construct concrete business models with operability.

On the other hand, one of the developed library for building knowledge systems on the concrete level is the inference catalogue of Common KADS methodology[Schreiber et.al. 1999; Common KADS]. Common KADS is utilized for analysis and development of knowledge systems and offers the language and primitives to clarify conceptual models. In late years, special method library REPOSIT is proposed for the implementation of inference primitives of Common KADS, but the methodology of the application construction including requirement analysis and development is still examined.

Owing to the difference of purposes among those repositories, when we try to build the real applications based on the existing domain, the integrated support are hardly performed in the whole process of the construction and the reengineering of business applications.

So, in order to achieve the unified support of the development, the computing framework, which integrates the different sorts of repositories, supports dynamic, should be developed for the construction of business models and applications.

From above-mentioned background, we propose the development methodology of business applications based on ontologies with reusable repositories such as e-business process handbook, Common KADS and REPOSIT.

In order to construct the business models and to implement them based on the existing domain, we rebuild the heterogeneous repositories into two repositories on different-grain-levels: the business specification repository on the level of business activities and the business software repository on the level of software applications.

It is the main characteristic of this work that our proposed framework enables integrated support of modeling business activities from business documents, and constructing business application from business models, by developing a platform to create the relationships dynamically between the descriptions of the business specification repository and the business software repository. We have implemented the prototype system by JAVA and confirmed that it supports us in various phases of business application development including

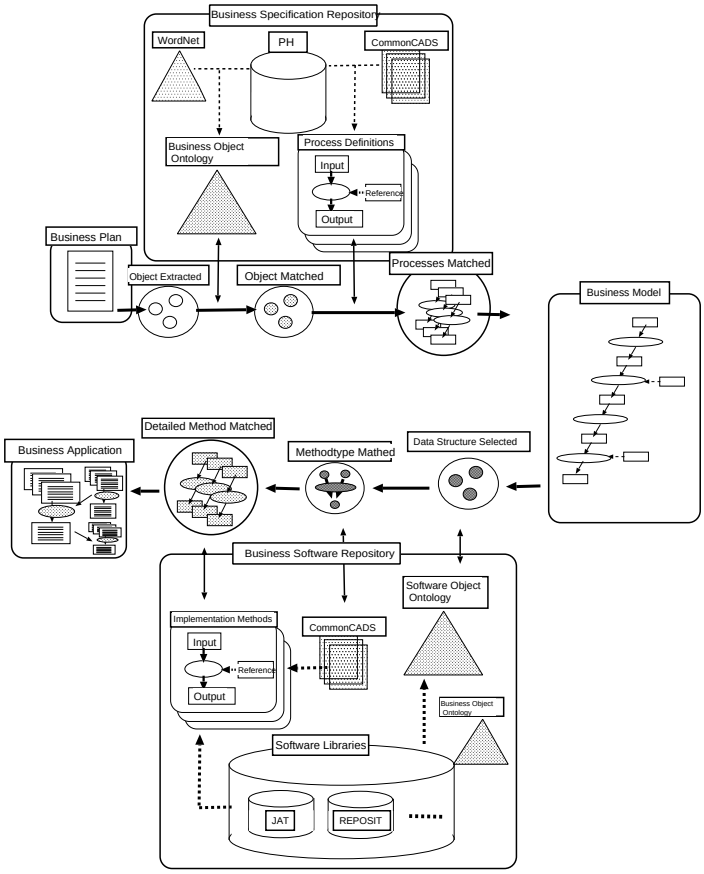

Figure 1: Overview of Development Support

business model manifestation, detailed business model definition and an implementation of business software applications. Since our work can be applied to advanced management judgment on introducing new business models, re-engineering of existing business processes, and employing new environment for business computing, we expect the repercussion effect on the management activities as the result of the information integration.

2 Overview of Proposed Development Support

In order to achieve the unified treatment of models on different levels such as making business models clear, implementing the detailed models, building software applications and so on, our research aims at the establishment of integrated support from the analysis level to the implementation level. Our standpoint of the integration and the reuse of heterogeneous repositories is to obtain the key structure of each repositories, which corresponds to noun concepts, and to relate them each other to bridge the whole structure of processes and activities including verb concepts.

The key idea to achieve our aim is how to extract the verb and noun concepts and their relationship as the common structure of information from the heterogeneous repositories. The target of development support is the construction of business application as a business models obtained from business documents, which contains the facility such as communication with users through network via E-mail or Web, and the file system sharing with the users for customer relationships and data management. Brief description of business application development through the repositories integration by our framework is as shown in Figure 1.

At first, in order to pick the key concepts up from business documents, we construct the business specification repository including ontologies of noun and verb concepts. Noun concepts are extracted from the e-business Process Handbook and classified into the business object ontology according to WordNet[Fellbaum 1998] as the general concept ontology provided by Princeton University. Verb concepts of e-business Process Handbook are classified by using co-occurrence information of noun concepts. By making reference to the history database of the correspondence between words of documents and ontologies of nouns and verbs, a business model is constructed by the business specification repository.

Next, as a repository for transplantation of the business model to a business software application, we provide the business software repository including method libraries. In order to correspond activities of business models to software mod-

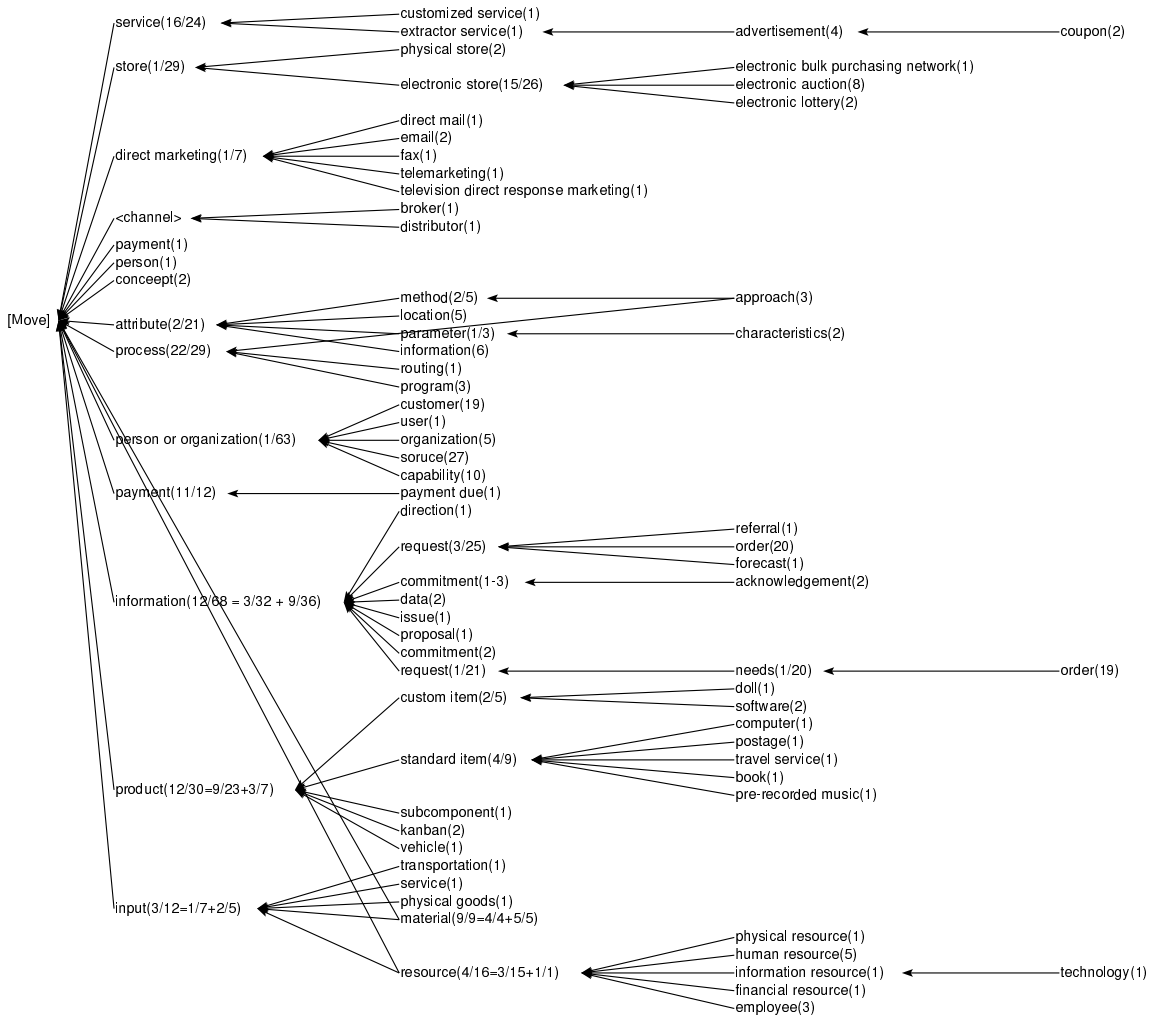

Figure 2: Object Tree Structure Obtained (a part of it)

ules, we construct a library of typical patterns of the input-output relations and the module structure of software systems based on JAT (JAVA Agent Template)[Petrie 1996] provided by Stanford University. Each pattern is formalized as a combination of inference primitives of Common KADS by introducing a software object ontology that provides the classification of objects with control and data structures. By consulting the software object ontology and REPOSIT[Izumi et.al. 1999b] provided by Shizuoka University as a library of the implementation patterns, the business model, obtained on the previous stage, are supplemented with control structures of software codes. According to the frequency and history of corresponding among three libraries of JAT, Common KADS and REPOSIT, a detailed model of the software application are obtained based on the software object ontology.

3 Construction of Business Models

In this work, we employ the e-business Process Handbook of MIT, called Process Handbook for short, as a library classifying business activities. In order to construct the business specification repository by extracting the required information, WordNet is also employed as a general lexical repository. First, the business specification repository is provided as a key structure bridging the business documents and Process Handbook. Then, the wrapper framework is constructed as an extract method of required activities from a repository of Process Handbook.

3.1 Building Business Object Ontology

When the abstraction degree of a business plan is high, a verb concept of an activity in the business plan is often vague for specification makers due to the difference of viewpoints on the definitions. In contrast, a noun concept of the activity is comparatively clear and appears regularly in the document. In order to classify the noun concepts extracted from Process Handbook, we employ the WordNet as a general ontology that contains over 17,000 concepts. However, if we utilize WordNet as it is, the number of candidate explodes because of the variety of the word's meaning and the ambiguity of the word in the document.

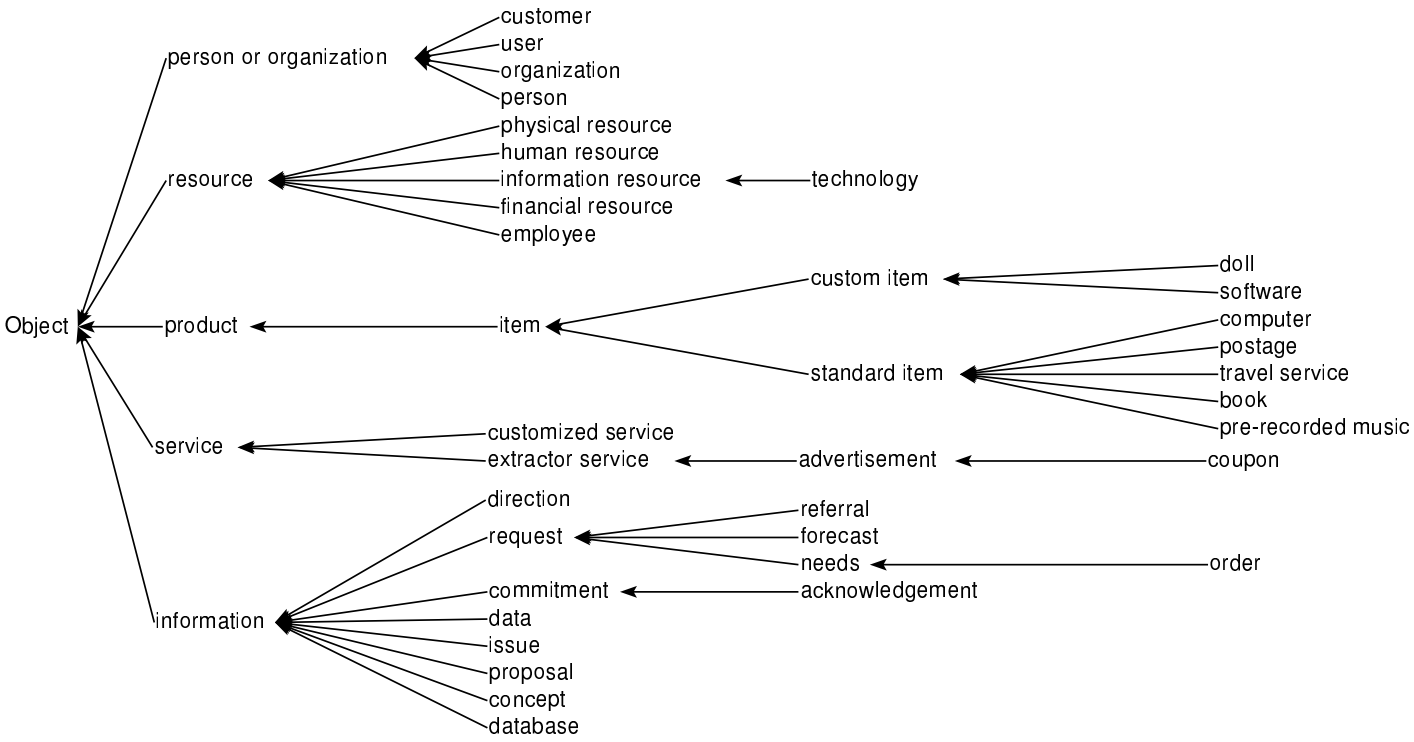

Figure 3: Business Object Ontology (a part of it)

So, in order to classify the noun concepts appearing in Process Handbook, we choose the major concepts with respect to the degree of abstraction and frequency by using WordNet in the following way.

First, noun concepts consisting Process Handbook are extracted and rearranged into a similar structure to the activities of Process Handbook. Then, the frequency of each word is counted and sum up with respect to the inherited structure. A structured tree of the words with the attribute of appearance frequency is obtained as shown in Figure 2. In the figure, the number of appearance is given in the fractional expression where the numerator corresponds the frequency of appearance as it is and the denominator means the inherited frequency of sub-concepts. The addition expression is a trace, which indicated the word, appears in the different position of the tree structure.

As the criteria to select a major noun concept, we pick the concepts with the more frequency up and rebuilt into an upper ontology. According to the WordNet's structure, we repeat the same way described above and define the substructure of concepts obtained above as an upper ontology, provided that the priority of the meaning is given to business domain over the relation of WordNet. Furthermore, when we fix the top ontology for constructing business domain ontology, concept drift, that is a kind of semantic shift on a specific domain, often occurs and causes inefficiency on building ontologies. Due to reduce the cost of construction, we employ the methodology for resolving the concept drift[Yamaguchi 1999]. Figure 3 shows the structure of the business object ontology obtained.

3.2 Determining Business Activities

When obtaining the definition of the business activity corresponding to a business document, it is difficult to utilize the hierarchical structure of the process handbook because of the gap between words of actual documents and process handbook. On this account, we construct the business object ontology, as a top ontology, which bridges the variety of words in documents and nouns in Process Handbook. In order to identify business activities from a sentence given by a user, we devise an extraction mechanism as a wrapper for Process Handbook based on the business object ontology.

The wrapper tool is composed of the databases of the following information. First, the co-occurrence information of noun concepts in the definition of a business activity is obtained and classified with respect to the structure of the business object ontology. Then, the information is made accessible as the database of the co-occurrence. At the same time, the frequency information is also available in collecting the co-occurrence one. By using both of the co-occurrence and frequency information, the wrapper tool help us to search the definition of activities in the space of Process Handbook. The proto-typing tool is shown in Figure 3.

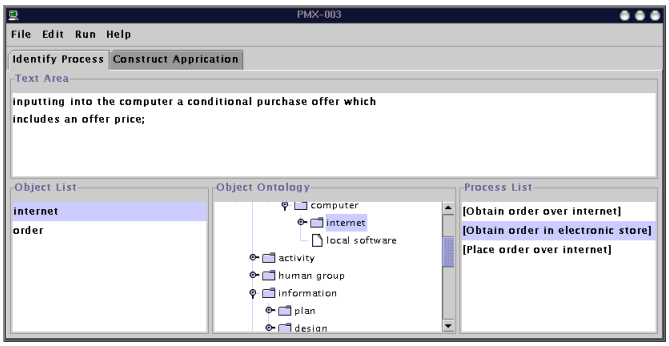

Figure 4: The Wrapper Tool

4 Building Business Applications

4.1 Building Software Object Ontology for Reuse of Libraries

In the same way of the business object ontology, the software object ontology (Figure 5) is constructed as a domain independent ontology from the libraries intended to employ. The software object ontology gives words for expanding domain ontologies such as building a set, picking up an atom of a set, indicating a calculation stage, data structures for implementation details and so on.

On the purpose of the developing business applications from the business model obtained above, a detailed definition of each activity of the model is required. In order to give the activities the operational information that is used for application development, we prepare the library of the application template, which defines the structure of the part of application in the fashion of the knowledge system development with respect to the software object ontology. By constructing the business software repository with the reusable template of REPOSIT, Common KADS, JAT and historical databases, the business application is obtained.

4.2 Model Refinement Based on Application Templates

We consider the structure of business applications based on the agent architecture to be composed by the inference engine to attain a task, the sensor to get the information of the outside and the effector to carry out their task. The sensor is characterized by the following three functions:

(1) the function that accesses the inside and the outside resources of the agent,

(2) the function that examines the place and the contents of resources,

(3) a function to acquire a message from the user and to interpret the message.

The effector is defined by two of the next:

(a) the function to form and modify the inside and the outside resources of the agent,

(b) a function to make and to send a message to the user.

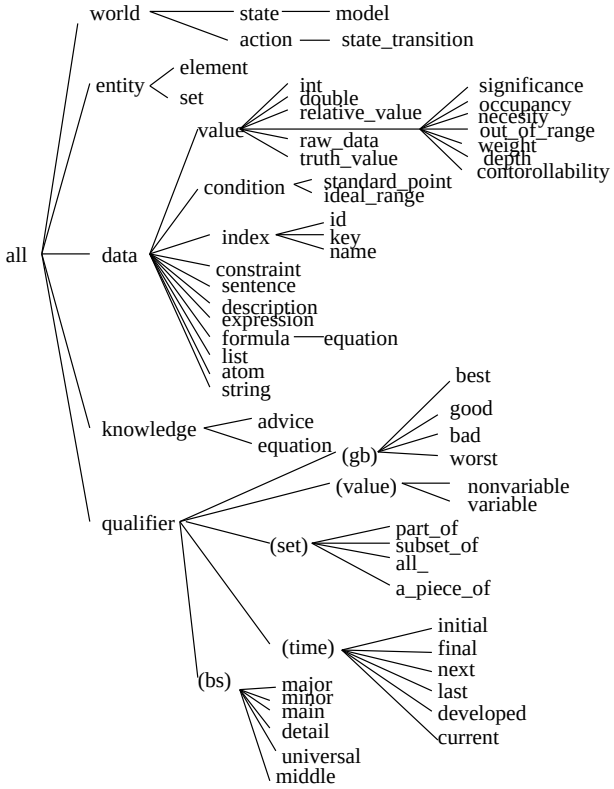

Figure 5: Standard data hierarchy

The framework of the combination with the above functions and the inference engine is organized as agent templates by referring to JAT (JAVA Agent Template) of the Stanford University. Furthermore, detailed templates, corresponding to eight types of communication models given by Common KADS, are formed as interaction templates with the user and resources (Figure 6).

4.3 Building Reusable Templates for Implementation of Applications

From the importance of a unified language for the reflection of the change on a business model, we rebuild and extend inference primitives of Common KADS into “REPOSIT (REusable Pieces Of Specification-Implementation Templates)” which combines declarative semantics employed in Common KADS and procedural semantics like Prolog. A unit of a description in REPOSIT, defined as a relationship among input, output and reference knowledge, is called a “unit function”. A set of unit functions is rebuilt into the method ontology by abstracting knowledge types of input, output and reference.

Furthermore, patterns of a combination of unit functions, which appear frequently in the development process, are gathered, sorted out and constructed as a method library based on the following standpoint:

(1) providing refinement policies,

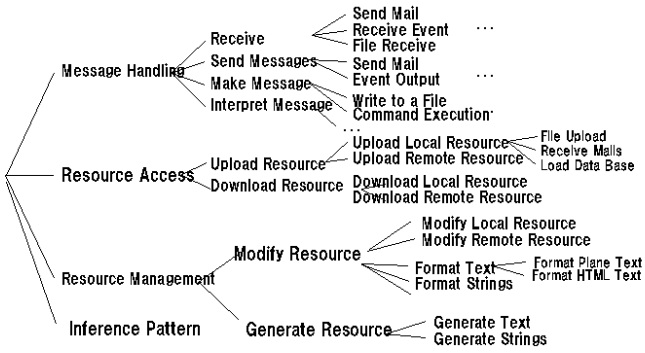

Figure 6: Primitives of Application Templates

(2) standardizing a way of the knowledge (data) management,

(3) classifying the adding patterns of control structures given to specifications.

In order to keep a correspondence between descriptions of specifications and implementations, REPOSIT supports step-by-step operationalization of an abstract description into a detailed implementation model, as the following way (Figure 7):

a. selecting a pattern of the method library according to a task type of a knowledge system,

b. concreting knowledge type of input, output and reference by using the software object ontology and the obtained business model as the requirement specification,

c. adding a control structure to the description with the obtained information of knowledge type,

d. selecting a pattern for each unit function of the description and continuing the above process.

Finally, we've provide 22 methods on the abstract-pattern level, 92 methods including prolog-build-in functions on the program-code level, and 69 methods on the middle-level.

4.4 Experimental Study

In order to consider the validness and usability of proposed framework, we've implemented the above mechanism by JAVA into the proto-typing tool and applied it into case studies of constructing business applications from description documents.

In each case study, some patent texts obtained from the Internet are used as a business document. We have compared between the models of case studies provided by Process Handbook and the ones built by the proto-type tool (Figure 7).

Roughly speaking about the result, approximately 70 % activities of each case study model are determined from patent texts, and, each implementation has been completed around 18 hours after receiving the patent text of claims and details as specifications.

The above means that the cost of the application development has been reduced more than 50% as compared with by hand, that reuse of the system has been performed about

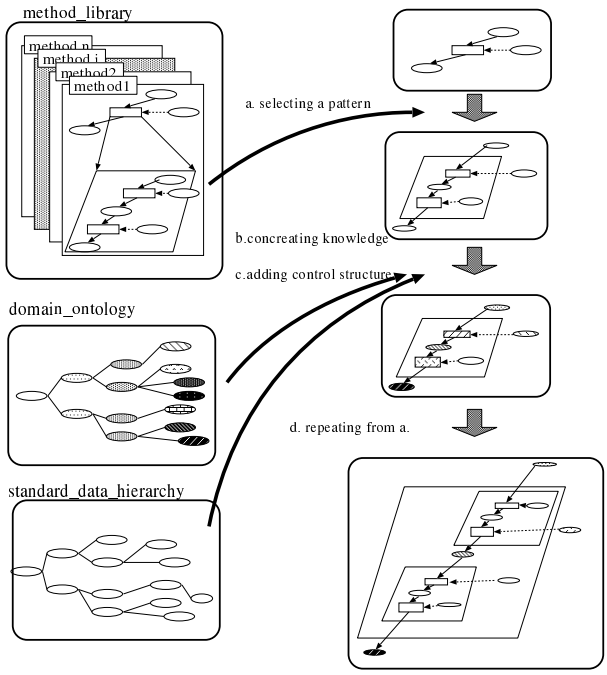

Figure 7: Overview of a Development Process

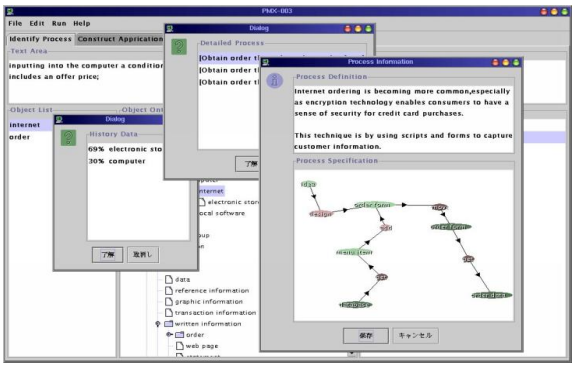

Figure 8: Execution of Supporting Tool

the common structure of business applications, and that main business structure could be reused if we have stacked and open some experience to the public at our library. Now, we are analyzing and investigating about the deeper experimental studies on the development on heterogeneous systems and the result will be open until the conference.

5 Discussions

The above result of the experimental study means that the cost of the application development has been reduced more than 50% as compared with by hand, that reuse of the system has been performed about the common structure of business applications, and that main business structure could be reused if we have stacked and open some experience to the public at our library. Now, we are analyzing and investigating about the deeper experimental studies on the development on heterogeneous systems.

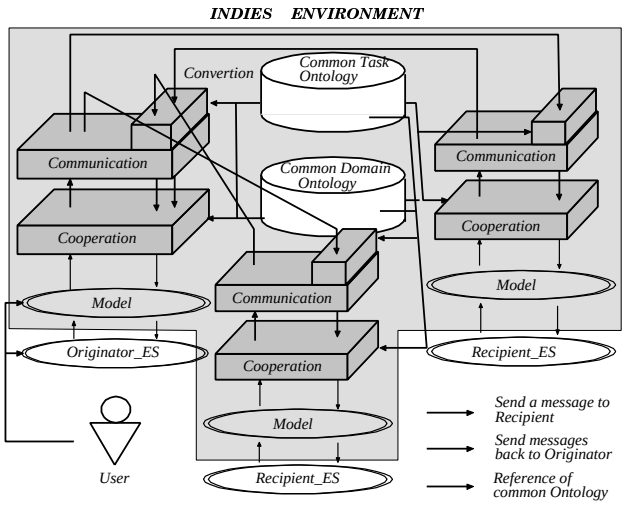

Figure 9: An Overview of INDIES

As comparison with related work, there are three main fields of research areas: clarifying specifications, building an application and reusing existing libraries.

First, as discussed in Introduction, numbers of work on analyzing business specification has been done by MIT, Edinburgh University and so on. Their work are very significant as a fundamental research, however, most of them are around abstract and general framework. Recently, Process Handbook is revised into e-Business Process Handbook and provides a hundred of specification as case studies. But most of them are just defined by natural language text. So, our work can be regarded as the integrated work to utilize the related work.

Second, a lot of works are carried out on building applications Including software engineering field[Code, et.al. 1997; **?**], but they use the different type of tool and languages on the different phases of development. So, models and languages should be unified into on framework as we proposed.

Our framework is based on a standpoint that it is difficult to automate the whole business, but possible to do many part of it. Recently, a few of the researchers consider the modeling methodology of the whole enterprise structure as a multi-agent system[Kendall 1999]. It is worth paying attention but still remain on the abstract structures of defining agent roles.

Finally, a lot of projects concentrate on reusing libraries, ontologies and applications and provide a number of repositories. One of our previous works is on interoperation of the heterogeneous expert systems[Izumi et.al. 1999a]. Because each expert system is modeled by its own vocabulary, it needs a conversion facility so that it can understand the messages sent from other expert systems. In the work, we employ a specification-sharing(SS)-based cooperation, called assisted coordination[Genesereth 1994]. The shared specification comes from REPOSIT library which serves as a common structure of noun and verb concepts named the common domain ontology and the common task ontology. As the methods of modeling, operationalizing, cooperating and communicating (wrapping) distributed expert systems come up, we put them together into an interoperation environment for distributed expert systems INDIES(Fig. 9).

In our previous work, a number of significant lessons obtained in exchanging the messages among the heterogeneous expert systems. However the way to construct the common ontologies is still remain as the future work. Our current work can be regarded as a new approach to reuse heterogeneous repositories based on ontologies.

6 Conclusion

As conclusion, the computing environment, which supports dynamic construction of business models and applications from business document, should be developed for the purpose to perform the re-engineering business processes according to the changes of business situations. From standpoint that the heterogeneous repositories should be integrated to achieve the unified support of the application development, we have proposed the framework of the extraction of the required information based on ontologies with reusable repositories such as e-business process handbook, Common KADS and REPOSIT. In order to construct the business models and to implement them as the actual business including software applications, we develop two repositories on different-grain-levels: the business specification repository on the level of business activities and the business software repository on the level of software applications.

We have implemented the prototype system by JAVA and confirmed that it supports us in various phases of business application development including business model manifestation, detailed business model definition and an implementation of business software applications. Furthermore, we are re-organizing our product in order to open it to the public.

References

[Code, et.al. 1997] P.Code, D.North, M.Mayfield, “Object Models:Strategies, Patterns, & Applications”, Yourdon Press, 1997.

[Fellbaum 1998] C.Fellbaum ed: “Wordnet”, The MIT Press, 1998.

[Fowler 1997] M.Fowler, “Analysis Patterns: Peusable Object Models”, Addison-Wesley, 1997.

[Genesereth 1994] M.R.Genesereth and S.P.Ketchpcl: Software Agents, CACM.ol.37.No.7. (1994) 48–53.

[Izumi et.al. 1999a] N.Izumi, A.Maruyama, A.Suzuki, T.Yamaguchi: “An Interoperative Environment for Developing Expert Systems” Proc. EKAW’99 LNAI.1621, pp.335–340, Springer–Verlag (1999).

[Izumi et.al. 1999b] N.Izumi, A.Maruyama, A.Suzuki, T.Yamaguchi: “Design and Implementations of Reusable Method Library for Development of Expert Systems”, Journal of JSAI, 14, 6, 1061–1071, (1999), in Japanese.

[Izumi and Yamaguchi 2000] N.Izumi, T.Yamaguchi: “Developing Software Agents Based on Product Ontology and Process Ontology”, Proc. ECAI-00 Workshop on Applications of Ontologies and Problem–Solving Methods (2000) 8-1—8-6.

[Kendall 1999] E.A.Kendall, “Role Models: Patterns of Agent Analysis and Design”, British Telecom Journal Special Issue on Decentralized Business Systems, (1999).

[MIT Process Handbook Project] The MIT Process Handbook Project: http://ccs.mit.edu/ph

[Petrie 1996] C.J.Petrie: “Agent-Based Engineering, the Web and Intelligence” IEEE Expert, 1996. URL: http://java.standord.edu/

[Schreiber et.al. 1999] Guuns Schreiber, et al: Knowledge Engineering and Management: The CommonKADS Methodology,MIT Press (1999).

[Uschold et.al. 1998] M.Uschold, et al: The Enterprise Ontology, Knowledge Engineering Review,Vol.13,Special Issue on Putting Ontologies to Use(1998).

[Yamaguchi 1999] T.Yamaguchi: Constructing Domain Ontologies Based on Concetp Drift Analysis, IJCAI Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends , 13-1 - 13-7, (1999.8)

[Common KADS] See http://www.commonkads.uva.nl