

An Ontology Based Reuse Algorithm towards Process Planning in Software Development

Shilpa Sharma

Sanghvi Institute of Management and Science
Indore, India

Maya Ingle

Indore Institute of Computer Applications
Indore, India

Abstract— The process planning task for specified design provisions in software development can be significantly developed by referencing the knowledge reuse scheme. Reuse is considered to be one of the most promising techniques to improve software excellence and productivity. Reuse during software development depends much on the existing design knowledge in meta-model, a “read only” repository of information. We have proposed, an ontology based reuse algorithm towards process planning in software development. According to the common conceptual base facilitated by ontology and the characteristics of knowledge, the concepts and the entities are represented into meta-model and endeavor prospects. The relations between these prospects and its linkage knowledge are used to construct an ontology based reuse algorithm. In addition, our experiment illustrates realization of process planning in software development by practicing this algorithm. Subsequently, its benefits are delineated.

Keywords- *Ontological Knowledge Modeling; OntoReuseAlgo; Ontolayering Principle.*

I. INTRODUCTION

Reuse practice has become a significant issue in the field of software engineering. The increasing demand for progressively larger and complex systems is one of the causes that reuse has been preferred when dealing with such tribulations in software development. Reuse also responds to an increasing insist for highly reliable, high excellence and less expensive systems.

Accordingly, knowledge reuse will benefit and improve the process planning in software development greatly. Process planning is an intermediate phase between design and implementation. Lucidity and prescribed specification of concepts play a key role in the inclusion of reuse in the process planning. Therefore, a most important issue is to build a common conceptual base characterized by knowledge. Our exploration focuses on this task through the use of defined system elements and layering principal, illustrating various prospects. We have proposed a knowledge based process planning method and it seems indispensable and potential in real applications.

As to knowledge-based process planning, a significant prose already exists for general software implementation processes. Ontology is the basis for the sharing of resources, which provides a common understanding of field knowledge. The scope of ontology application is widespread, and its pursuit is large and whole, and applicable to entire software

engineering environment [1] [2]. Therefore, we construct and use a fixed ontology oriented procedure through locating the linkage knowledge. When executing it, irrelevant knowledge items would consequently be avoided in knowledge retrieval so as the effectiveness and efficiency of knowledge reuse may be improved. This may help train novice users to quickly become familiar with the process and achieve efficient and high quality builds.

Ontology has been defined as the vocabulary of unambiguous domain-related concepts and meanings of the concepts anchored in consensus domain knowledge [3]. Also, ontology may provide a reusable and sharable information model. It is popular in engineering knowledge modeling and retrieval for its devastating searching ability [4]. Thus, we have proposed an ontological knowledge modeling in Section II that illustrates the meta-level descriptions to divulge the building blocks of it. Based on ontological knowledge modeling, Section III confers ontology based reuse algorithm towards process planning in software development. Section IV exemplifies a case study to substantiate the efficacy of ontology based reuse algorithm. Finally, we conclude with the advantage of practicing it towards process planning in software

II. ONTOLOGICAL KNOWLEDGE MODELING

We have proposed an ontological knowledge modeling for knowledge integration and reuse towards process planning in software development. It constitutes *System Element Classification*, *Ontolayering Principle* and *Knowledge Reuse Scheme* to provide reusable and shareable engineering applications. *System Element Classification* is developed to capture important characteristics to reduce the growing complexity of information and increasing need to exchange it among various software applications. Subsequently, the *Ontolayering principle* uses the ontology in a resource usage manner, specifically by understanding and dissimulating the information comprised by entities. Conversely, natural language is too ambiguous to provide enough and clear definition in engineering applications. Thus, *Knowledge Reuse Scheme* is defined consistent interpretation between different users and different applications. These building blocks are defined as follows:

A. System Element Classification

The classification includes abstract concepts such as work units, stages, work products, model units and people as shown in Fig. 1. Work units are the tasks or activities that software

developers perform, and have a start and end time as well as duration. Stages are major time frames, helps the work units in giving some temporal structure. Work products, such as documents or software, are intangible results of performing work units and have creations and last change times with

status. The status of work product is always one of the enumerated list such as Initial, Complete, Accepted or Approved. Finally, the producers are the people and teams that actually perform the work units in order to create work products.

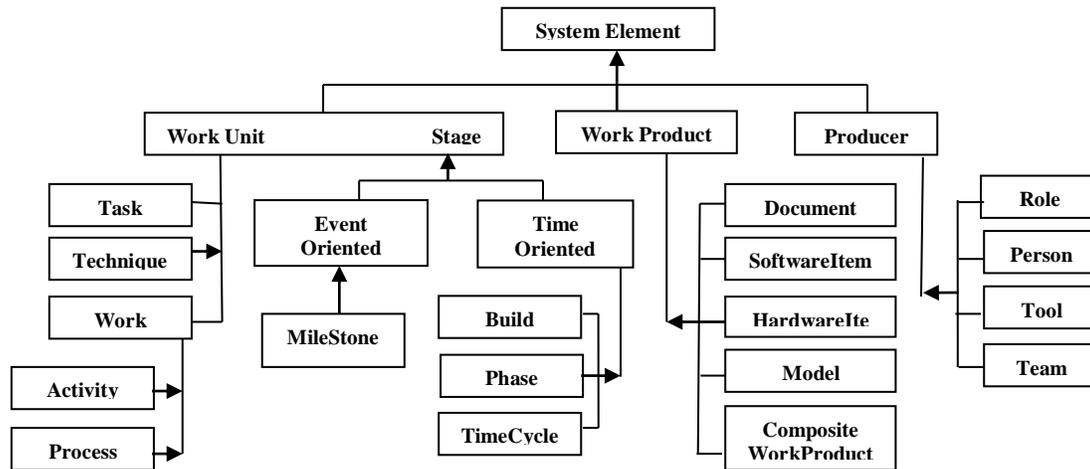


Figure 1: System Element Classification

B. Ontolayering Principle

The principle uses the ontology in a resource usage manner specifically by understanding and dissimilating the information comprised by entities [5]. The three prospects namely: Metamodel, Process and Product prospects have been defined around the communities that network with the ontology as shown in Fig. 2. The meta-model prospect acts as a common standard determining the other prospects. The meta-model is intended to be used as an origin by method engineers so that the methodologies can be developed. Method

engineers typically uses the concepts in the meta-model prospect by sub typing and instantiation, thereby creating new concepts (subtype of existing ones) and entities (instances of concepts). All these new concepts and entities created by method engineers are seized to form a process prospect. Software developers use it by creating the instances of concepts in it and also, by following the guidance explained by entities. Thus, the instances created by software developers are apprehended to form the product prospect.

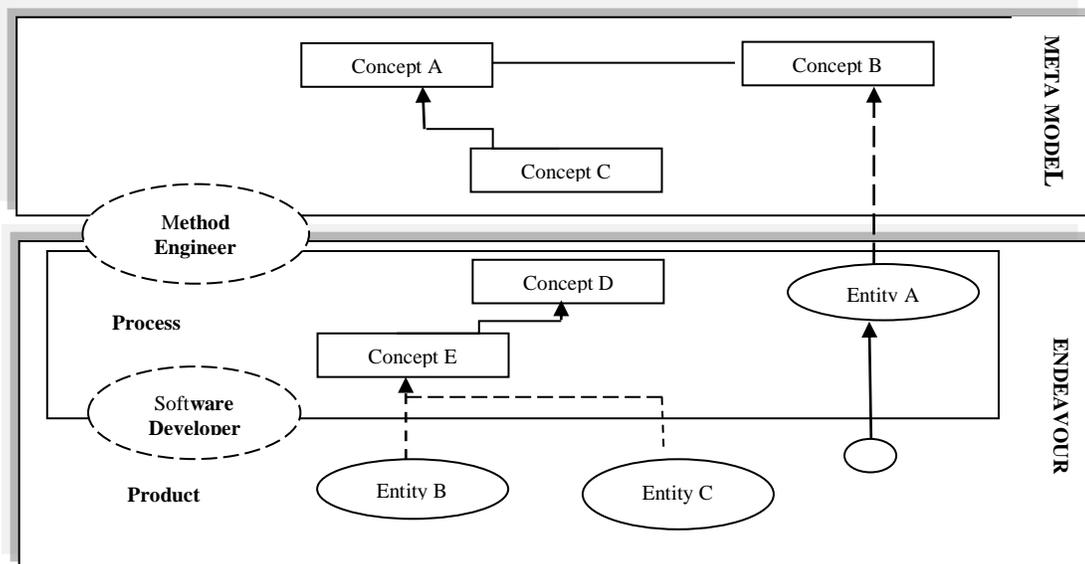


Figure 2: Ontolayering Principle

C. Knowledge Reuse Scheme

The scheme starts with formalizing the system element requirements according to the representation approach of System Element Classification. Subsequently, identification of

the related process concepts and entities that need to be revised according to the *Ontolayering* principle stored in the knowledge base. Consequently, modification of the producer entities, associated concepts based on *Ontolayering* principle

and revision of the influencing product attributes conceded. Finally, simulation of the results is done if the reuse requirements are satisfied; else consider changing a different process concept and entities.

III. ONTOLOGY BASED REUSE ALGORITHM (*ONTOREUSEALGO*)

We have proposed an Ontology Based Reuse Algorithm (*OntoReuseAlgo*) using ontological knowledge modeling approach to aid the product redesign of process plans. The key idea is that a new process plan under new implementation requirements may be obtained by modification of certain concepts and entities of the current process. Ontological knowledge modeling is used to give a uniform representation of the involved information. It starts with understanding the system elements. It includes identifying process concepts and entities that need to change followed by altering with *Ontolayering* principle and modify the producer entity and the associated concepts which help in revising the influencing

product attributes for simulating the final process plan. Fig. 3 shows the overall procedure of the proposed approach and the knowledge reuse strategy in process planning.

Different systems may use different concepts and terminology to express the same thing while the same words may be used to represent different things by different systems. Both situations hinder information communication. Therefore, step 1 uses the *System Element Classification*. Conversely, the process and product concepts that need to change could be properly identified and revised through mappings of related *Ontolayering Principle* to certain process and product prospects in step 2, 3 and 4. Subsequently, natural language is too ambiguous to provide enough and clear definition in engineering applications. In addition, text-based definitions are too rigid and insufficient in information retrieval because it necessitates text-match searching. Thus, *Knowledge Reuse Scheme* is applied to step 5 for providing vocabulary and the specification of the meaning or semiotics of the terminology within this vocabulary.

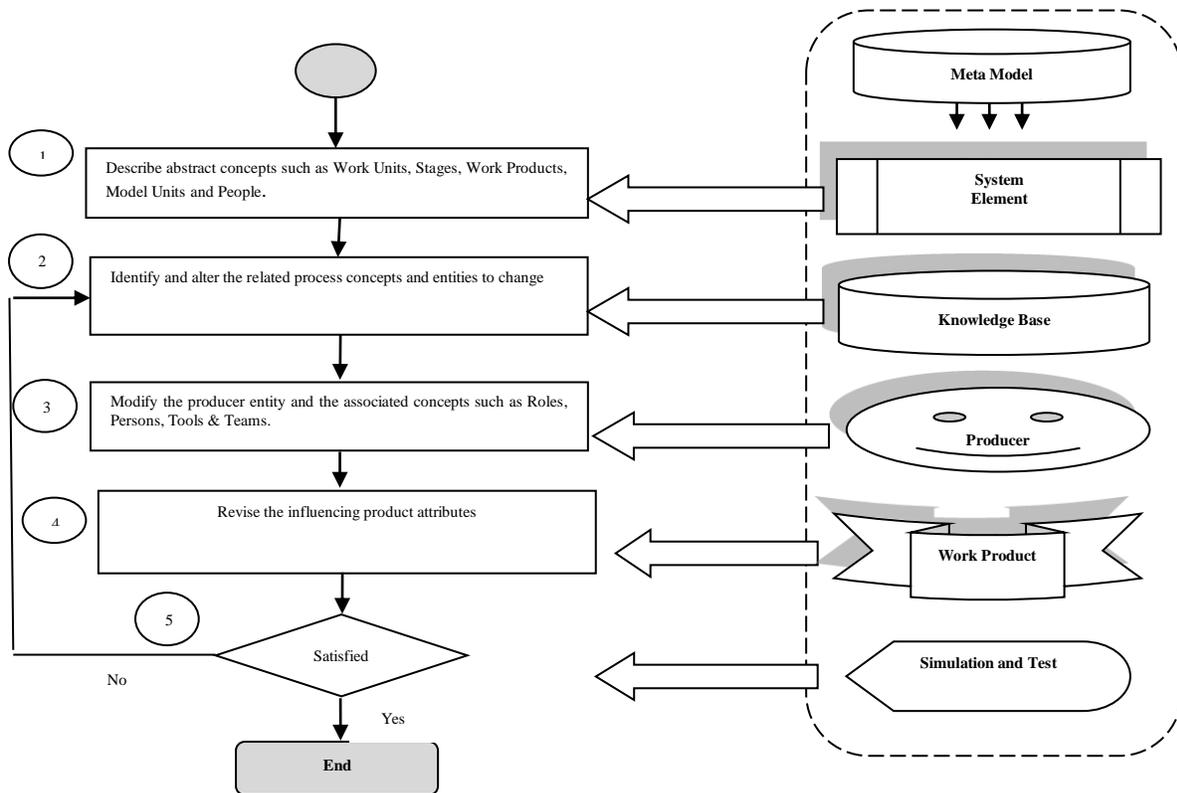


Figure 3: *OntoReuseAlgo*: Reuse Approach for Process Planning

There are 5 steps from the beginning to the end of process planning task, as shown in Figure 1

Step 1: Formalize the system element requirements according to the representation approach of system element classification

Step 2: Identify the related process concepts and entities that need to be revised according to the layering principle stored in the knowledge base

Step 3: Modify the producer entities and associated concepts based on layering principle

Step 4: Revise the influencing product attributes

Step 5: Simulation of the results in step 4 if the reuse requirements are satisfied, then shift to step 5; or else, shift to step 2 and consider changing a different process concept and entities

IV. CASE STUDY

In this section, the design of process planning for redistribution between warehouses is described, based on the proposed reuse approach an *OntoAlgoReuse*. Various people are responsible for carrying out different activities such as Foreman is responsible for one warehouse though Warehouse worker works in a warehouse for loading and unloading subsequently Truck driver is accountable for transportation. In addition, Forklift operator drives a forklift in one warehouse and Official personal receive orders and requests from customers.

Step1- We have analyzed following set of system element requirements, when a foreman performs redistribution of items between the warehouses.

Figure 4: Redistribution between Warehouses Window

TABLE I : SYSTEM ELEMENT REQUIREMENTS OF RETRIEVED CASE

S.No	System element Requirement	Description
1	Initialization (when foreman request for the redistribution)	<ol style="list-style-type: none"> The foreman gives a command for redistribution between warehouses The window in Fig. 4 is presented to the foreman The items can be ordered in a number of ways with ORDER menu such as alphabetical, index, turnover of the items and storing order. In the 'From place' table we may choose to view either all places in the current warehouse or, if we have selected an item, the place where the item exists In the 'To warehouse' table we may select all warehouses or the warehouses that we have to transport to this week The 'Issuer' and 'warehouse' fields are automatically filled when the window pops up. The foreman selects an item by pointing to it and dragging it to the Redistribution form then selects from which place to take the items and to which warehouse to transport them. The foreman then gives the quantity to be moved and the date. It is possible to change the information when the form has been edited. When the foreman EXECUTES the redistribution, the transport is planned. It is also possible to CANCEL the redistribution. Selecting HELP shows window of information about the current window.
2	Loading (when truck fetches the item from the warehouse)	<ol style="list-style-type: none"> A Truck driver asks for a transportation request. The request is marked as ongoing. Give an appropriate request to the Forklift operators to have the items ready when and where the truck is expected. When the Warehouse Worker gets a request to fetch items at appropriate time, orders Forklift operators to move the items to the loading platform When the Truck driver arrives the items are loaded. The Truck driver tells the system when the truck is loaded and when it is expected to be at the new warehouse. Decrease the number of items in this ware house and mark the transport request as on transport
3	Unloading (when a truck delivers the items to the new warehouse)	<ol style="list-style-type: none"> When the truck has arrived at the new warehouse, the items are unloaded The Truck driver tells the system that the transport to this warehouse has been done. The Warehouse workers receive the items and determine a place for them in the warehouse Forklift operators are told to move the items to the new place in the new warehouse When the Truck driver confirms the insertion, the system updates the new place for the items The transportation time is recorded and stored in the system The Redistribution and the transport request are marked as performed.

TABLE II : PROBLEMS WITH RETRIEVED CASE

S.No	Erroneous courses	Description
1	A request is not executable	The execution is interrupted and the Foreman issuing the request is informed
2	Redistribution is wrong	<ol style="list-style-type: none"> The warehouse place does not have enough items to move The destination warehouse is not appropriate to the item
3	No truck available	When performing loading, and unloading, there may not be any truck available at an appropriate time. Then notify the Foreman who should either delete the request or change it.

Step2, 3 and 4- According to the system element requirements mentioned in Table I, the layering principle has identified

following erroneous courses in the service of redistribution between warehouses as shown in Table II.

Step 5: Table III shows the results after executing the knowledge reuse scheme for the task of process planning. Thus, planning endow with new transport requests and

therefore change existing transport requests previously in the system

TABLE III : RESULTS

	Planning	Description
1	To coordinate transports and issue transport requests	<ol style="list-style-type: none">1. When the redistribution is executed the items to be moved are marked as move-pending2. Minimize the use of trucks on condition that all delivery dates should be held and the trucks should be compatible with any delivery requirements for the items.3. The transport requests are connected to a specific truck's transportation plan.

V. CONCLUSION

An *OntoReuseAlgo* aims to improve the knowledge reuse in process planning for software development. It supports the application from three aspects: system element classification, representation on the basis of layering principle and knowledge reuse scheme for process planning. The system element classification and the layering principle are adjoined for the workflow control of the knowledge reuse. We have observed the following significant benefits and are as follows:

- Through organizing and modeling the knowledge towards the characteristics of design processes, unnecessary search time can be avoided on irrelevant knowledge items
- It allows explicit credentials for analysis and comparison of different domain theories
- It describes knowledge acquisition approach to structure the entities and relations that need to be acquired in the domain;
- It provides a metalevel view (vocabulary and structure) on their domain which facilitates adequate system documentation and constructs reusable knowledge-system design
- It can be used to define assumptions that enable knowledge exchange between different agents.

REFERENCES

- [1] Cossentino, M., Burrafato, P., Lombardo, S., Sabatucci, S., Introducing Pattern Reuse in the Design of Multi-Agent Systems, AITA02, Workshop at NODE 02, Erfurt, Germany, pp. 8-9, October 2002.
- [2] De Almeida Falbo, R.; Guizzardi, G. and Duarte, K. C., "An Ontological Approach to Domain Engineering", In Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE '92), Ischia, Italy, July 15-19, 2002.
- [3] Dileo, J., Jacobs, T., Deloach, S. Integrating Ontologies into Multi-Agent Systems Engineering, Proceedings of 4th International Bi-Conference Workshop on Agent Oriented Information Systems (AOIS 2002), pp. 15-16, Bologna (Italy), July 2002.
- [4] DeLoach, S.: Multi-Agent Systems Engineering: A Methodology and

Language for Designing Agent Systems In Proceeding of the 1st Bi-Conf. Workshop on Agent-Oriented Information Systems (AOIS'99), Heidelberg, Germany,1999.

- [5] Faria, C. "A Technique for the Acquisition and Construction of Ontology-based Domain and User Models in Multi-Agent Domain Engineering", Master degree dissertation, Federal University of Maranhão – CPGEE, 2004.
- [6] Ferreira, S. "A Technique and Tool for Domain Design in Multi-Agent Domain Engineering", Master degree dissertation, Federal University of Maranhão – CPGEE, 2004
- [7] Ferreira, S. and Girardi, R. "Specification of a Generic Ontology for Domain Design", Proceedings of III Workshop of Software Engineering (WIS 2003) at Chilean Computing Week 2003 . Chilán, Chile. November 3rd to 8th, 2003. (In Portuguese)
- [8] Fleming, M. and Cohen, R., User Modeling in the Design of Interactive Interface Agents, Proceedings of UM99 (User Modeling), 1999
- [9] International Workshop on Component Engineering Methodology (WCEM'03) at Second International Conference on Generative Programming and Component Engineering, Ed. Sven Overhage and Klaus Turowski, pp. 41-50. Erfurt, Germany. September 22-25, 2003.
- [10] Gruber, T. R. "Towards Principles for the Design of Ontologies used for Knowledge Sharing". International Journal Human-Computer Studies. N° 43, pp. 907-928, 1995
- [11] Guarino, N. "Formal Ontology and Information Systems", Proceedings of FOIS'98, Trento, Italy. Amsterdan, IOS Press, pp. 3-15, 6-8 June, 1998
- [12] Harsu, M. "A Survey on Domain Engineering", Report 31, Institute of Software Systems, Tampere University of Technology, December 2002, 26 pp

AUTHORS PROFILE

Shilpa Sharma, Senior Lecturer at SIMS, Indore, India. She has 5 years of teaching experience. She is pursuing Ph.D. in Computer Science and is MCA from Vaishnav Institute of Management, Rajiv Gandhi Vishwavidhyalaya, Bhopal, India. She has 6 papers published to her name in various National and International journals and conferences.

Dr. Maya Ingle is Professor & Senior System Analyst at School of Computer Science and Information Technology, Devi Ahilya University, Indore since 25 years. Presently, she is Principal and Professor at Indore Institute of Computer Application, Indore on extra ordinary leave. She is also Chair Assessment & Research committee. She is Ph. D. in Computer Science and M. Tech in Computer Science from IIT, Kharagpur. Her research interest includes Software Engineering, NLP, Speech Recognition, Usability Engineering and Agile Computing. She has published more than 80 papers in national and international journals and conferences.