

Structure design of knowledge base of software enterprise based on project development

Xin Cui

School of Economics and Management, Huangshan University,
Huangshan, Anhui 245041, China

Received December 28, 2016

This paper mainly discusses the knowledge management of software project, and designs the innovation knowledge base with the project development as the core.

Keywords: software enterprise; project development; knowledge base; structure

Обсуждается создание программного проекта управления знаниями, а также проектируется база данных об инновациях с разработкой проекта в качестве основного.

Структурний проектування бази даних програмного забезпечення підприємства на стадії розробки проекту. *Синь Цуй*

Обговорюється створення програмного проекту управління знаннями, а також проектується база даних про інновації з розробкою проекту в якості основного.

1. Introduction

At present, the construction of China's software enterprise knowledge base is still in the beginning and the knowledge base structure doesn't have a unified standard and paradigm. In practice, because knowledge base construction is one of the main ways to build the core competitiveness of enterprises, the use of knowledge base for enterprises is very strict, and the public knowledge base of software enterprises is very rare, so it is necessary to discuss about structure of knowledge base, and provides the reference for the knowledge management of software enterprises.

In the process of combing the status qua of knowledge base research at home and abroad, the author focused on the related literature such as SFFU, Melbourne University and European WISE project, and analyzed the system framework of COMMONKADS, VITAL and COMMET. The author visited the knowledge base of Tsinghua Uni-splendor Corporation and obtained suggestions from Tong fang Software and knowledge base builder of Asian-info

Technology, and obtained the support of Beijing Municipal Information Industry Committee, which provided valuable internal information "Guideline of Software Industry Knowledge Base Construction". Based on the above theoretical and practical research, combined with ways of classification for the subject and process of software enterprises, this paper proposes a knowledge base structure for software enterprise, and the main structure of it is "two libraries and one chain". "Two libraries" refers to the two resources. In addition to the traditional public knowledge base, there is a semi-modular project innovation library that targets to project development process. "A chain" refers to the innovation knowledge chain for the specific project.

2. Design of public knowledge base

Public knowledge base refers to the accumulation of experience that supports core technology of software enterprise and knowledge assets. Its subject classification is knowledge classification system, standard, business

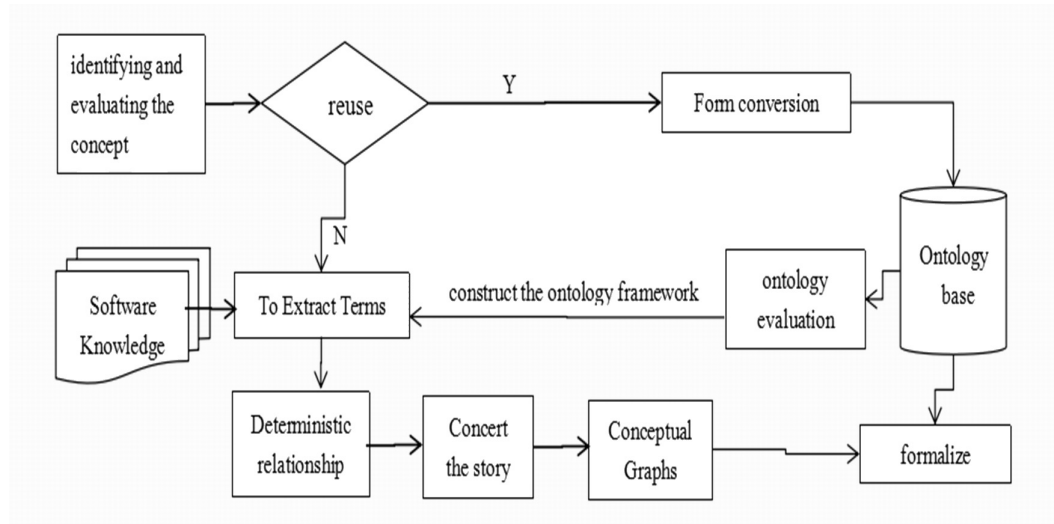


Fig.1 Concrete steps of ontology construction

model, data model, application program part, solution and so on. Software companies use the knowledge base for design of applicable solutions, software development and training. Its purpose is to build the core competitiveness of enterprises through knowledge accumulation, reuse and innovation to build the core competitiveness of companies and promote the long-term development of enterprises. Specific implementation is as follows:

A. Building knowledge ontology

Ontology provides a unified semantic integration between the system of knowledge reuse and diffusion. The ontology building knowledge is very important, and its creation process should be easier to run. They require the creator to treat them with caution. This paper uses the theory and method of knowledge engineering to get the steps of constructing ontology of software enterprise (see Fig. 1).

1) *The important terms in this field and concepts:* In general, [1] identifying and evaluating the concept in these domains requires experts to first identify the most important concepts, find their relevant concepts, and list a list of all concepts. To describe the problem further, this study shows only parts of the core vocabulary: software enterprise analysis report, business model, development process.

2) *List each concept attribute and attribute value:* The concepts given by experts in the field, as well as the concept of attribute and attribute values need to be listed separately, for example, "project background" contains the subject of "doing the research on the customer in field."

3) *The connection and the relationship of different concepts:* To decide the relationship of

ontology, the main contents of this paper are as follows: "Part and whole, inheritance, concept instance and concept, attribute, father and subclass"[2]. **In addition, there are also the concept of the relationship between the opposite, the concept of synonymous relations between concepts.** These two relations are relatively rare in practice.

B. Establish architecture of ontology knowledge base

The architecture is based on the ontology-based knowledge representation method and the knowledge about development process of software projects is the management object (see Fig.2).

1) *The infrastructure layer:* includes the corresponding hardware support platform and software support platform. The main purpose is to consolidate the software enterprise infrastructure for the upper level knowledge management and logical reasoning of the knowledge ontology. It is the main entity part of ontology library construction for the software enterprise. Per the general requirements, in this layer, we need to determine servers, storage and other hardware equipment model and price, as well as the required software, **middle ware and the overall functional requirements;**

2) *Knowledge management support layer:* plays a role in the system structure, which is responsible for the establishment of **meta data, directory system, logical information reasonably arranged in the software system and storage media;**

3) *The knowledge content layer:* It is about combining the actual knowledge base to the above knowledge ontology reasoning layer. Its essence is the original content of the reasoning

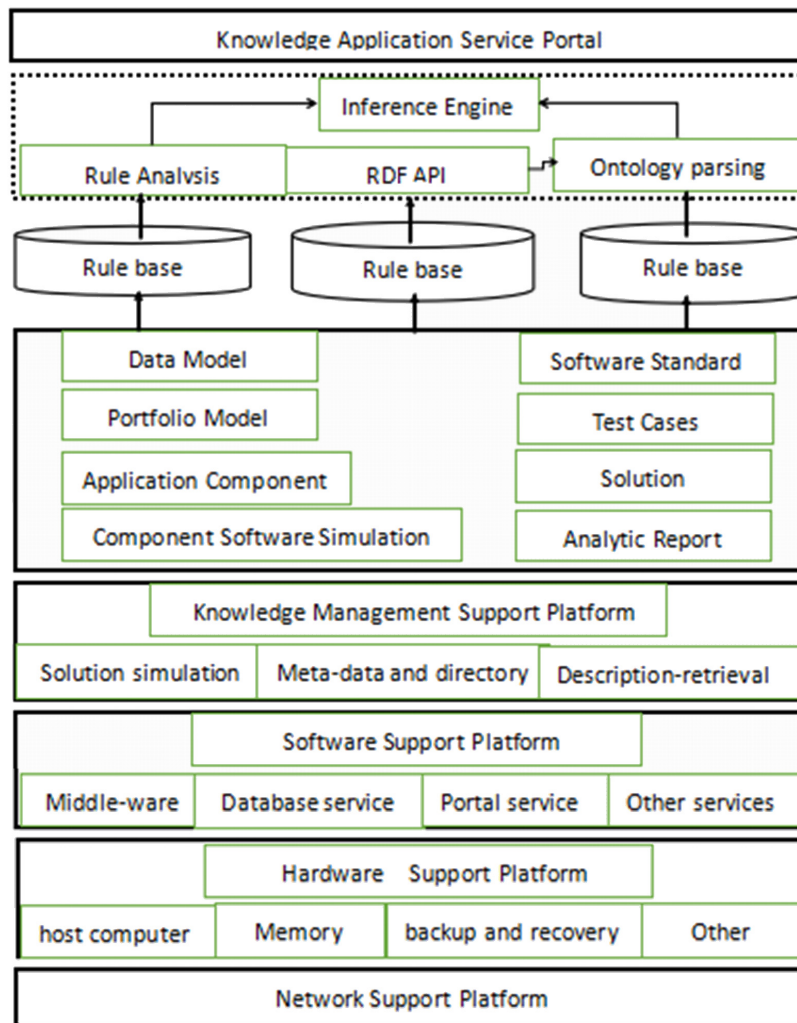


Fig.2 The architecture of ontology knowledge base

layer, including the analysis report, business model, development process.

4) *Knowledge ontology reasoning*: It is based on representation of ontology knowledge, including database which stores instance of ontology, rule base, and knowledge ontology database, which containing knowledge resource ontology and concept ontology.

5) *Knowledge service portal*: It is mainly to provide interactive interface for resources, for user inquiries and landing services. When a registered user accesses the system, the authentication is passed; he/she types in the keyword; the interface accepts the requirements; the server will deliver the knowledge to the user. If a non-registered user accesses the system and the user does not pass the authentication, the server closes its channel and the user cannot query the information.

3. Design the innovation knowledge base

Software companies set project development as the main task; the development process includes parallel, serial and cross-relations. For such a complex development process, to improve the efficiency of knowledge base, the integrated use of knowledge push and navigation technology of knowledge is very important. To facilitate the acquisition of knowledge, knowledge reuse and knowledge sharing, development process will be refined into multiple template path. During the project development, the relevant historical knowledge modules are loaded into the appropriate template, which can greatly improve the development speed, therefore, modular knowledge for project development is inevitable. Based on the above, the knowledge enterprise process classification and path template construction strategy, this paper

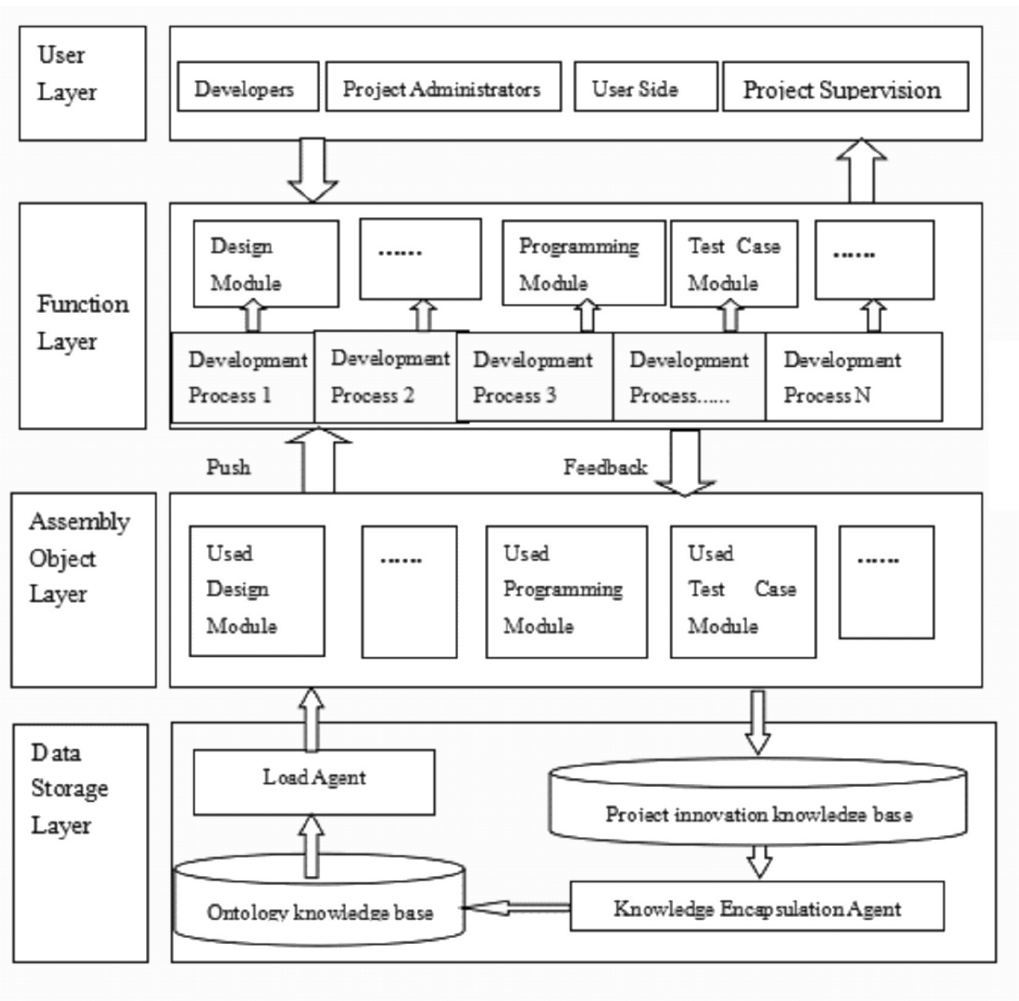


Fig.3 The architecture of the innovation knowledge base

presents the concept of innovation database for software enterprise project. The main steps are as follows:

A. The development of knowledge modular

It divides the project development knowledge into several modules and pushes the historical knowledge module to the knowledge module which the developer prepares to construct. It not only reduces the repetitive development, but also greatly reduces the development time. At the same time, it realizes the validity, timeliness and focus of distributing knowledge. For example: its knowledge module for the development of a teacher training platform project includes interface design, registration, logic, teacher information, announcement, elective, curriculum, self-test, learning assessment, offline test, teaching forum and resource library. By knowledge push, we can find that the offline test module fully meets the current project development needs, so that the offline test module doesn't need to be developed. For this project

development, it not only saves the development time, but also greatly reduces the development costs, while saving staff costs. Therefore, the construction strategy for the development of knowledge modular is of great significance for software development.

The concept of modular is brought by the Japanese scholar Aoki in his book "the Module Age: a new industrial structure," in 2003[3]. Muscularity is the process of building complex products with each of the smaller subsystems that can be designed independently and can play a role in the whole. The modulation of project-development knowledge refers to the muscularity of knowledge encapsulation. The post-packaged knowledge module includes software components, knowledge attributes and knowledge background. Its role is to facilitate the realization of discrete knowledge aggregation, facilitate the sharing, reuse of knowledge and the construction of knowledge base.

B. Development process path template

To develop the software, we need to define the development process of the project. The process path will be template. We will analyze the critical and minor paths, the start and completion times of the critical path, and determine the development cycle, and then determine the type of development. The main development type is as follows: one is the serial relationship, regardless of the critical path, all development path templates have occurred. It is for small project development and team members are few. The advantage is to save labor cost. the disadvantage is that it may lead to a long development cycle. The second is the parallel relationship. After determining the critical path, the secondary path can be carried out at the same time or be postponed. Each path template is independent of each other, with the critical path in parallel relationship, input and loading knowledge can choose the method that the key path is first and the secondary path is the next. It is for medium-sized project development, and team members are not too many, The advantage is that developers can give full play to their technical advantages. The disadvantage is that it is likely to cause increased labor costs. The third is cross-relationship. Critical path and the secondary path are run at the same time, and after the development task of the secondary path is completed, developers could choose the certain module in the critical path and continue to develop until the completion of all tasks. This cross-development method is used more for large-scale project development. The advantage is to save labor costs and shorten development time; the disadvantage is that it likely to result in employee conflict and cannot give full play to the technical advantages of staff.

In software development, the development of each individual template can be understood as a serial development. Parallel and cross-development type can also be understood as the composition of multiple serial development. If we connect path templates, knowledge modules and development process chains, developers can touch a certain path template in the process chain of the development project to obtain the knowledge module of the template link.

The advantage of the development process template is that the knowledge acquisition time is saved. It also makes it easy for the developer to select the template associated with his specialty and use their advantages in technology. Meanwhile, the diversification of combination of templates makes the development process from complex to simple and lies the foundation for software development towards standard-

ized and fast. The disadvantage is that the template may bring attention to detail and ignore the overall; developers may be more concerned about the construction of a single template, and the overall understanding of the development project is not enough. This condition is not conducive to the management of unity of the development project and leads to different languages using in the software development, the heterogeneity of knowledge, and ultimately leads to instability of running software system. Therefore, the construction of process development chain and the establishment of a unified knowledge-sharing platform is necessary.

C. Establish architecture of the innovation knowledge base

The innovation knowledge base needs to meet the needs of project development of software enterprise. Its users are mainly project developers, and clients, third-party supervision and business managers. The link of function layer is variable, mainly include path template and knowledge module. Path template is the link of knowledge module, and it is relatively stable. Knowledge push is mainly realized through agent; it has the function of knowledge caching and plays a role in linking public knowledge base. The data layer mainly consists of public knowledge base and innovation library, which is the storage place of knowledge. Based on the above considerations, this paper proposes the following structure of project innovation knowledge base (see Fig. 3).

1) *The first layer is the data layer:* the layer of knowledge acquisition, organization and storage. It is divided into two parts: ontology library and project innovation library. The ontology library provides the knowledge needed for the project development, and through the encapsulation agent filters, packs and modular the knowledge into public library to store it permanently.

2) *The second layer is the knowledge assembly layer:* to fully meet the requirements of knowledge search; the history of knowledge and knowledge related background need to be shown to the user, the knowledge engine needs to accept the access from the upper, analyze the user needs and the relationship of knowledge units, the dynamic connection of the knowledge unit in the ontology library is needed and the corresponding knowledge module in the development step layer needs to be submitted.

3) *The third layer is the functional layer:* the development process is displayed in the form of the path template and each path template is relatively fixed, which is to facilitate the continuity of historical memory storage. To facilitate the flexible development process management,

project management personnel can add the path template per the actual demand, trigger the AGENT, load the corresponding knowledge module, realize the integration of the knowledge module and the path template, and form the knowledge module containing the path template.

4) *The fourth layer is the user layer:* The clients of this library are developers, project managers, and this structure needs to meet the standard for the inspection from project development commissioners and project supervisors.

IV. Formation Process of Knowledge Innovation Chain

Knowledge innovation increases the knowledge stock of knowledge base and improves the core competitiveness of enterprises. Building knowledge chain can activate the flow of knowledge to achieve effective access to knowledge, increase knowledge assets of enterprises to provide knowledge support for project development.

A. The main role of knowledge innovation chain

the main role of knowledge innovation chain is reflected in the following areas:

1) *Enhance the core competitiveness of software enterprises:* Software enterprises need to participate in the competition, and the strength is mainly from the core competitiveness of enterprises. Well-known economist Gary Hamel in the "Core Competitiveness of Enterprises," stated: "The core competitiveness of enterprises is actually implied in the core products of knowledge and skills." The long-term and sustained competitive advantage of software enterprises is derived from the capacity of obtaining knowledge, knowledge accumulation and capability of knowledge innovation. These capabilities have promoted the competitive advantage of enterprises. Software development is the source and inner power of the development of software enterprises and is basic guarantee to obtain long-term competitive advantage. The construction of knowledge innovation chain is good for the discovery, accumulation and application of innovative knowledge in project development process. Because the knowledge of software development is extensively linked, the individual and organizational knowledge, explicit and implicit knowledge can be reflected, recognized, integrated, and provided feedback. The knowledge can provide the strong support to further development and research. It is shown that [4], expert learners are better at solving real and complex problems than novices. Therefore, the integration of software development knowledge

through the innovation chain can transform the advantages of separated knowledge of the software enterprises into general knowledge, improve the software Development efficiency, and enhance the core competitiveness of software enterprises.

2) *Increase the accumulation of innovative knowledge of software enterprises:* In the software project development process, most people are concerned with such a superficial process: demand research, module design, code writing, testing and handover. That is from the thought to applications and the economic chain runs through but there is less attention for innovation chain. Since the nature of software project development is knowledge, this process is essentially the application of historical knowledge and the generation of innovative knowledge. Each activity of it is accompanied by the flow of innovative knowledge. Software project development is the process of acquiring, organizing, spreading and reintegrating relevant software knowledge. Innovation chain runs through the whole process of software development. The rapid recognition, collation and integration of innovative knowledge has promoted the accumulation of innovative knowledge in software enterprises.

B. The construction of knowledge innovation chain

The knowledge innovation chain takes the project development path template provided by the project innovation knowledge base as the carrier, carries on the knowledge innovation, causes the knowledge acquisition, the reuse and the proliferation all to occur in the path template. Its movement process (see Fig. 4), mainly includes the initial stage, refinement phase, construction phase, evaluation stage and transfer phase.

1) *Initial phase:* After landing a development project, the project developer selects the push template or retrieves the corresponding class path template based on the requirements and tasks of the project development. It triggers a new instance to load the required knowledge, load the required knowledge to the path template, and save the instance in the path template.

2) *Refinement phase:* To understand the path and process of project development, load the corresponding knowledge module, extract the path of the implementation of the required development knowledge, and load it to the path module. The path to complete the starting time, marking who completed and development tools and development languages needs to be cleared.

3) *Construction stage:* According to the selected development knowledge, based on the ref-

erence and understanding of historical knowledge module, the developers need to complete the development tasks in this path, combining with the problems encountered, encapsulate its attributes, characteristics and knowledge background into a knowledge module, and form new knowledge modules.

4) *Review stage*: knowledge management team needs to review, test, analyze the formation of new knowledge module for each path module to determine whether there is an exception. If there is, then the module needs to be returned to the delivers. If not, the new knowledge module will move into the next stage.

5) *Transfer phase*: **The knowledge management team** test the normal new knowledge module again. If there is no problem, then the module can be stored in the public knowledge base, become a historical module to be permanently stored.

References

1. Ina Fourie and Theo Bothma. *The Electronic Library*, **24**(4), 469, 2009.
2. Peter Holdt Christensen. *J. Knowledge Management*, **11**(23), 36, 2007.
3. Schilling M. A. Steensma H. K. *Acad. Management J.*, **44**(6), 1149, 2001.
4. Amrit Tiwana. *Inform. Software Techn.*, **6**, 899, 2014.