The Architecture for Solving the Cross-Domain Keywords During New Product Development

Article *in* Advanced Materials Research · December 2012

DOI: 10.4028/www.scientific.net/AMR.605-607.497

3 authors:

Yung-Chung Tsao
Providence University
254 PUBLICATIONS **76** CITATIONS
SEE PROFILE

Kevin Hsu
National Central University
69 PUBLICATIONS **60** CITATIONS
SEE PROFILE

Yin Te Tsai
Providence University
115 PUBLICATIONS **451** CITATIONS
SEE PROFILE

Advanced Materials Research Vols. 605-607 (2013) pp 497-500
© (2013) Trans Tech Publications, Switzerland
doi:10.4028/www.scientific.net/AMR.605-607.497

The Architecture for Solving the Cross-domain Keywords during New Product Development

Yung-Chung Tsao[1,a], Kevin Chihcheng Hsu[1,b], Yin-Te Tsai[2,c]

[1]No.300, Jhongda Rd., Jhongli City, Taoyuan County 32001, Taiwan(R.O.C.)

[2]200 Chung-Chi Rd., Salu Dist., Taichung City 43301, Taiwan(R.O.C.)

[a]tyc6095@ms1.hinet.net, [b]khsu@mgt.ncu.edu.tw, [c]yttsai@pu.edu.tw

Keywords: New Product Development (NPD), Information Systems (IS), Knowledge Management System (KMS), Product Data Management (PDM), Product Lifecycle Management (PLM).

Abstract. The team-members for the new product development (NPD) are recruited from different departments in the enterprises. So the team-members in the design teams range from novices to experts in NPD. The juniors developer in the design project team lack more successful product-design experiences as their seniority and skills. Therefore, those developers always query and search their problems with the limited terminology via the information systems or websites. So the results of the querying and searching always are limited to the similar domain-knowledge.

The new product development (NPD) involves multidisciplinary knowledge such as accumulated experiences of knowledge-workers, and different technical and legal documents. The information systems (IS) facilitating the NPD processes often include document-based knowledge management system (KMS), Product Data Management (PDM), Product Lifecycle Management (PLM) systems etc. With these different systems, novices at various stages of NPD processes often have problems to use the exact and suitable keywords to query the problems from those information systems.

The study proposes a case-based reasoning to construct a hierarchical knowledge model to record knowledge-workers' experiences and to store the information of experiences and the recommendation of experts. The aim of the study is that the proposed architecture can query the information scattered in different information systems by using their individual-domain terminology and retrieve the better fitted results of the querying.

Introduction

Background and Motive

The processes of New Product Development (NPD) involve multidisciplinary knowledge. These knowledge scattered over design team members and various information systems including document-based knowledge management system (KMS), Product Data Management (PDM), and Product Lifecycle Management (PLM) systems. Therefore, facing the scattered knowledge during the NPD processes, the novices usually have lower success rate to acquire the exact informative documents due to the lack of the suitable cross-domains' terms as the parameters for retrieval in the involved information systems. Even the experienced users on the less-explored domains will face these difficulties.

The keyword-based searching requires the deep insight & the knowledge to use the correct keywords, so the novices search and query the documents and information with their known keywords limited to their inexperienced skills and knowledge. If the novices don't know the exact and suitable keywords, they get the querying results limited to their individual domain indeed. How to help those novices to share the past experiences becomes the critical issues in the study.

The Aims of the Study

As the problems mentioned above, the study propose an architecture to help novices by using the keywords based on their individual-domain to acquire the querying results of cross-domain

Literature References

Tasks in New Product Development Processes

The Cooper (1994) had recommended the staged-based approaches in NPD processes into the seven stages: (1) Product Conceptual Design (2) Product Detailed Design (3) Prototyping (4) Design Validation & Verification (5) Pilot Run (6) Mass Production (7) Customer Service[1-2] as shown in Figure 1.

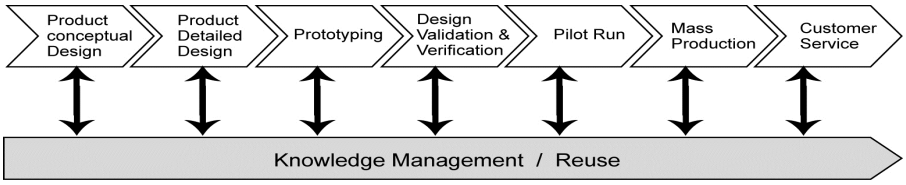

Figure 1 Stage-based processes in the NPD

The required knowledge in the different stage may have specific documents as shown in Table 1. therefore, the sharing of multi-styled documents become important issues in NPD.

Table 1 Types of Documents in Each Stage

Stages	Types of Document
Product Conceptual Design	Marketing research, Requirements, Analysis reports, Conceptual specification
Product Detailed Design	Detailed specification, Evaluation reports, BOM
Prototyping	Configuration manual, Specification, CAD, Alternative BOM, E-BOM, Test plan
Design Validation & Verification	Testing report, E-BOM, M-BOM
Pilot Run	Testing report, M-BOM, S-BOM, Quality assurance report
Mass Production	M-BOM, Engineering change records, Sales reports
Customer Service	Product manual, Maintain Feedback, S-BOM

Case-based Approach

The researchers proposed the Case-Based Reasoning (CBR)[3-6] to help solve the new problem based on the previous situations similar to the current one. Case-based reasoning can adapt an old solutions to match new problems by using old cases to explain new situations and to retrieve suitable solutions; or to interpret a new problem-situation or create an equitable solution-sets to a new problem from precedents.

The Proposed Architecture

The Proposed Architecture

The study constructs a conceptual architecture based on the stages' approaches with the tasks related to the roles in exact stages as shown in Figure 2.

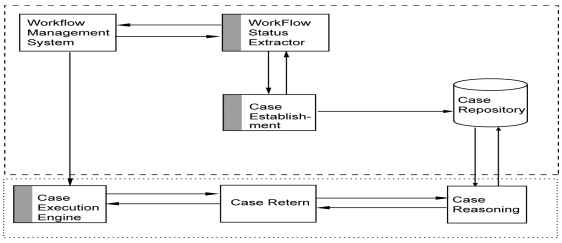

Figure 2 The Proposed Architecture

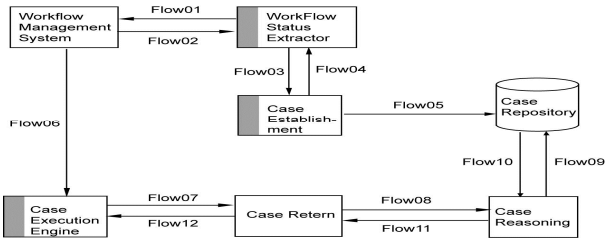

Figure 3 The System Flow of the Architecture

The System Flow of the Proposed Architecture

Case establishment

The system flow of the proposed architecture as shown in Figure 3 execute a case establishment after the querying requests. The full system flow loop each time when a new case establishment.

- Flow01: Novices or users query the problems via Workflow Management System.
- Flow02: The Work Status Extractor extracts all status for the users, including tasks, stages, classifications, roles, and/or components of product development.
- Flow03: The Work Status Extractor send all status into the Case Establishment as new case.
- Flow05: The Case Establishment retain this new case into Case Repository.
- Flow04: The Case Establishment notifies the Workflow Management System after the case established.

Query Execution

The system flow of the proposed architecture as shown in Figure 3 execute a query plan with the past cases from case reasoning. The full system flow loop each time when a new querying requests.

- Flow06: Novices or users wish to query the problems via Workflow Management System. So the Workflow Management System send all status for the users, including tasks, stages, classifications, roles, and/or components of product development into the Query Execution Engine.
- Flow07: The Query Execution Engine need the past cases' experience, so send all case data to the Case Retriever for past cases reuse.
- Flow08: The Case Retriever send the current case to the Case Reasoning to prepare the retrieving the past cases from the Case Repository.
- Flow09: The Case Reasoning search the Case Repository for retrieving the better-fit cases.
- Flow10: The Case Repository returns the better-fit past cases to the Case Reasoning.
- Flow11: The Case Reasoning returns the better-fit past cases to the Case Retriever.
- Flow12: The Case Retriever returns the better-fit past cases to the Query Execution Engine.
- Query Execution Engine: The Query Execution Engine combine the current cases and the returned past cases into the Query Dispatch Plan.

The Architecture for Knowledge

Knowledge Hierarchy

The Necib (2003) indicated the human being, were able to understand the fuzzy & chaos concepts with the representations of the object' instances abstractly. The users with the lower levels of skill, however, were difficult to query the exact answering precisely as the same as the experts. Therefore, how to bring out the types abstractive hierarchy and the cooperative querying answering for enlarging and shrinking the querying scopes[7-8].

One perspective of the product' structures is the hierarchical knowledge representations of the products' perspective as shown in Figure 4.

For examples, the queries are the parameters for searching the reference of the part Y_1, the retrieval via the parameter Y_1 and A_3 results only the assembly relationship. But the loss information of Y_1 with A_3 lacks the brother relationship to the part Y_2, Y_2. Due to the workflow approaches, the systems still offer more the relationship information of the part Y_1, Y_2, Y_3 and the assembly A_3 in the aspects of the attributes including stages, tasks, roles, etc. .

The stage information constructs the hierarchy of the product development processes for each stage with the attributes as shown in Figure 5.The roles' information constructs the hierarchy of role' relationships for each team members with the attributes & the roles additionally as shown in Figure 6.

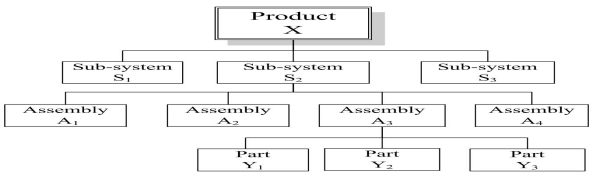

Figure 4 Products' Structures

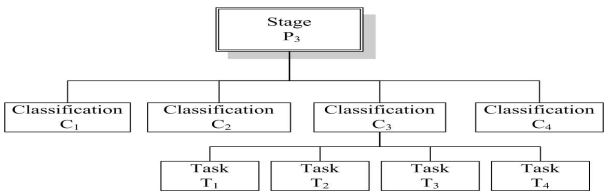

Figure 5 Product Development Processes

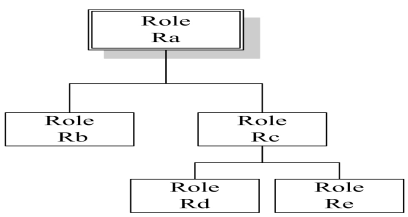

Figure 6 Organizational Structure

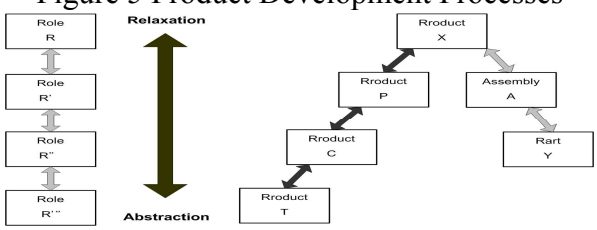

Figure 7 The Abstraction and Relaxation of Knowledge Hierarchy

The Abstraction and Relaxation of Knowledge Hierarchy

In a view of the integrated approaches mentioned above, the knowledge hierarchy in multi-domains constructs the conceptual model including the bi-direction of the abstraction and the relaxation, even more the attributes added to the relationships based on the individual attributes scattered in the cross-domains as shown in Figure 7.

Conclusions

The study proposes an architecture to solve the cross-domains keywords in NPD as shown in Figure 3 with case-base reasoning approaches and methodology. Based on the hierarchy of knowledge, the study can provide better relaxation as shown in Figure 7 to match single-domain keywords to multi-domains terms for different roles in their stages as shown in Figure 1.

Therefore, the proposed architecture can help novices to use their familiar terms to query the problems as before, and the results of querying return can exceed the old results.

Acknowledgements

The study was partially sponsored by NSC100-2218-E126-001, NSC100-2410-H-008-051-, NSC99-2632-E126-001-MY3, NSC99-2218-E-126-001, NSC99-2632-E-126-001-MY3, NSC99-2632-E-126 -005 -MY3, NSC98-2511-008-003-MY2, NSC97-2410-H-008-039

References

[1].Cooper, R. G. (1994). Perspective third-generation new product processes. Journal of Product Innovation Management, 11(1), 3-14.

[2].Cooper, R. G. (2008). Perspective: The Stage-Gate® Idea-to-Launch Process—Update, What's New, and NexGen Systems*. Journal of Product Innovation Management, 25(3), 213-232.

[3].Kolodner, J. L. (1983a). Maintaining organization in a dynamic long-term memory. Cognitive Science, 7(4), 243-280.

[4].Kolodner, J. L. (1983b). Reconstructive memory: A computer model. Cognitive Science, 7(4), 281-328.

[5].Kolodner, J. L. (1988). DARPA Workshop on Case-Based Reasoning: Proceedings: Morgan Kaufmann Publishers Inc.

[6].Kolodner, J. L. (1992). An introduction to case-based reasoning. Artificial Intelligence Review, 6(1), 3-34.

[7].Necib, C., & Freytag, J. C. (2003). Ontology based query processing in database management systems. On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, 839-857.

[8].Necib, C., & Freytag, J. C. (2005). Query processing using ontologies.