Towards the Definition of Domain Concepts and Knowledge through the Application of the User Story Mapping Method

Ana Milicic, Apostolos Perdikakis, Soumaya Kadiri, Dimitris Kiritsis, Petko Ivanov

HAL Id: hal-01526136
https://hal.inria.fr/hal-01526136

Submitted on 22 May 2017

Towards the Definition of Domain Concepts and Knowledge through the Application of the User Story Mapping Method

Ana Milicic[1], Apostolos Perdikakis[1], Soumaya El Kadiri[1], Dimitris Kiritsis[1] and Petko Ivanov[2]

[1] LICP laboratory, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
{ana.milicic}{apostolos.perdikakis}{soumaya.elkadiri}
{dimitris.kiritsis}@epfl.ch

[2] SAP Research, Dresden, Germany
p.ivanov@sap.com

Abstract. The context and problem of identifying and thereafter representing, analyzing and managing information and knowledge about an organization has always been very crucial to achieve business goals in an efficient and flexible way. Particularly in a PLM context, the issue of information overload is growing in importance. An emergent challenge consists in providing a context-driven access to federated information and knowledge and fostering cross-discipline collaborations between actors to improve quality in product development. This paper highlights key issues for knowledge definition and representation. We propose a bottom-up approach based on the User Story Mapping method (USM). This method is user-centric and leads to the definition of current and/or expected scenarios and processes along with a collaboratively agreed vision. Common concepts and viewpoints are therefore derived and generalized through a process of merging defined roles, activities and usages sequences with a focus on the product content. This bottom-up approach provides a federated and common understanding of information throughout the industrial product and process lifecycle; which combined with appropriate tools and methods, such as questionnaires, standards specifications, knowledge based approaches, etc. results in the definition of the knowledge network and domain and therefore improves capabilities for sharing and reusing this knowledge in collaborative product development. The proposed approach is applied in the context of the FP7 European project LinkedDesign (Linked Knowledge in Manufacturing, Engineering and Design for Next-Generation Production) based on three application scenarios.

Keywords: Knowledge representation and definition, PLM, Bottom-Up approach, User Story Mapping Method

1   Introduction

Innovation is the application of knowledge to produce new knowledge [1]. It requires systematic efforts and a high degree of organization. As we enter the knowledge society, ownership of knowledge and information as a source of competitive advantage is becoming increasingly important. In other words, organizations depend more on the development, use and distribution of knowledge based competencies. This is particularly relevant in knowledge intensive processes such as product innovation. Consequently, research and development (R&D) organizations are paying more attention to the concept of managing their knowledge base in order to increase competitive advantage, through effective decision making and innovation [2][3][4]. Knowledge is a key resource that must be managed if improvement efforts are to succeed and businesses are to remain competitive in a networked environment [5]. In particular, the two major challenges that face organizations are: (a) ensuring that they have the knowledge to support their operations and (b) ensuring that they optimize the knowledge resources available to them. Managing knowledge is about creating an environment that fosters the continuous creation, aggregation, use and reuse of both organizational and personal knowledge in the pursuit of new business value. Knowledge management can be considered to be a systematic and organized attempt to use knowledge within a company to transform its ability to generate, store and use knowledge in order to improve performance. In short, the overriding purpose of enterprise knowledge management is to make knowledge accessible and reusable to the organization.

Capturing domain specific knowledge is one of the main challenges in the field of Knowledge Based Engineering (KBE) [6]. Several methodologies have been elaborated to guide knowledge acquisition activities and thus avoid omitting essential knowledge [7] but they usually require a time-consuming collection and analysis of (often implicit) knowledge about the product and its design process, respectively [8]. Thus, most approaches to designing KBE-Tools address especially repetitive engineering tasks [9][10], since the potential to reduce time and cost by means of such approaches has to be balanced against the effort needed to gather and formalize the required knowledge in a scheme (e.g. an ontology) [11][12]. The User Story Mapping method comes to address the previously stated challenges by providing an efficient, time saving, bottom up requirements analysis for the design of KBE-Tools. It is a user centric method which allows the designers of the software to learn what the future users expect from this KBE-Tool, as well as it helps the users to express their over-all demands in functional view which is close to them.

The rest of the paper is organized as follows: Section 2 discusses knowledge definition and representation issues; Section 3 presents the USM method and the proposed approach and; Section 5 concludes the research findings.

2   Knowledge definition and representation

Knowledge is an elusive concept and therefore it is important to define it in context in order to understand it. The term is used in several different ways in the literature. For example, Nonaka and Takuechi in [13], two of the early researchers in this field, adopt a philosophical angle and define knowledge as "justified true belief". In this view, knowledge is an opinion, idea or theory that has been verified empirically and agreed upon by a community. According to Wilson in [14], knowledge at the most basic level is "that which is known". Quinn et al in [15] liken knowledge with professional intellect where professional intellect in organizations centers on know-what, know-why, know-how and self-motivated creativity. Stewart in [16] also considers knowledge in terms of intellectual capital. On the other hand, Bohn in [17] examines knowledge in terms of a company's processes. He believes that an organization's knowledge about its processes may range from total ignorance about how they work to very complex and formal mathematical models. According to Davenport et al in [18], knowledge is information combined with experience, context, interpretation and reflection. It is a high value form of information that is ready to apply to decisions and actions. Simply put, knowledge can be defined as the integration of ideas, experience, intuition, assertions, skills and lessons learned that have the potential to create value for a business by informing decisions and improving performance. In this view, knowledge is a key enabler to organizational success. However, in order for knowledge to be useful it must be available, accurate, effective and accessible.

   Question of knowledge representation first emerged in the field of artificial intelligence, where the experts were working on representing the knowledge using predefined set of symbols. Application of knowledge engineering in PLM context required that the format used for representing the knowledge is understandable by both, humans and machines. For this reason, number of methods was developed, including relational diagrams and linked tables but lately, ontologies are shown to be preferable choice. Ontologies proved to be very convenient for organizing and storing the data. They enable automatic reasoning and inference which means that beside the knowledge gathered in the time of modeling the ontology, additional relations will be automatically built up in time.

   Specification and conceptualization of ontologies lean on the identification of the relevant concepts of a particular domain, their type, and the constraints on their use. However, existing methodologies (Diligent, Methontholgy, On-To-Knowledge) lack detailed and clear guidelines for building the concepts. On the one hand, the process of concepts definition represents a key issue for knowledge gathering as it has to cover in an optimal way the whole domain. On the other hand, several knowledge resources may exist and their concepts reuse can be of a key importance.
The NeOn Methdology [20] comes to deal with the aforementioned issues and provides some methodological guidelines for performing the ontology requirements specification activity, to obtain the requirements that the ontology should fulfill. Particularly, it consists of elaborating an ORSD (Ontology Requirements Specification Document) which aims to list, among others, the intended uses, the end-

users and a set of questions describing the requirements that the ontology should fulfill.

Nevertheless, this approach of listing intended uses and questions that the ontology should respond to may appear as a flat structure, in the sense that it doesn't lead to study and analyze the domain mainly in terms of interactions that link the end-users and usages, before going in deep into the questions that the concepts should be able to answer. The research conducted in this paper has a focus on the first phase of concepts definition as it represents the main basis for knowledge definition and conceptualization. The following section discusses the proposed approach for dealing with concepts definition based on one of the agile methods, called the User Story Mapping.

3 A bottom-up approach based on USM

The problem of information overload exists in many different domains. It is usually difficult to find the most suitable way of documenting, describing, and transferring gathered knowledge, be it for software, products, services, etc. In this work we propose the usage of an approach from the agile software development called the User Story Mapping (USM) [19].

When a new software product is being developed, one of the first steps of the process is to document the idea. This usually results in a description of key features that the developed product will have, optionally including a short abstract, called the "elevator pitch" that will be used to advertise the product and show its value to the customer. After shortly documenting the idea, the next step is to develop a concrete list of action items or tasks, also called backlog, that need to be implemented in order to transform the idea into a concrete product. Unfortunately, such backlogs, event arranged in a priority order are usually flat structures. They help the team members to understand what needs to be done next, but unfortunately do not explain why it needs to be done and what the whole system or product does. Such approach can be compared with having the puzzle pieces, but not knowing what the whole picture should look like, not knowing what the final goal is.

An approach to overcome the problem of how to create a good backlog is the user story mapping. It is not only a way of structuring the backlog items, it is a way of visually documenting ideas, a way of sharing a concrete product vision, functionalities, users that can benefit from the product, and how they use the product. It is a way to communicate not only within and among teams, but also with users and customers.

3.1 User Story Mapping Method

A user story map is a user centric approach and organizes the backlog along scenarios and users. It answers the question how a user uses the product and consists of several structure blocks as it can be seen in Figure 1:

- Usage dimension – It describes how a user would use the product. It shows the sequence of steps that a user would perform when using the product. It is very important that usage steps cover the whole scope of the product usage.
- User dimension – This dimension defines the types of users that will use the developed product. It helps to identify different users and the aspects of the product that will be interesting for those users.
- Backbone – This section describes the activities that a user performs within a usage step. The backbone describes the activities that a user performs when using the developed product. This section is called backbone as it often represents the essentials of the product and suits as a guideline for the definition of the user stories, which are actually a refinement of the backbone.
- User stories as backlog items – This is the actual placeholder for the user stories. The user stories are ordered vertically under each activity and represent a refined version of an activity. It is recommended that user stories follow the pattern "As <user> I want to <feature> so that <value>". After all user stories have been defined, they also need to be prioritized, taking into account the value that this user story brings, the technical risk of implementing it and the effort that will be needed to implement the story.

Although the USM method was originally proposed and used in the area of agile software development, it is a generic method for structuring and sharing information about a product. It visualizes in a simple way several aspects of a product looking from the user perspective. It is both means of communication and means of documenting knowledge in PLM.
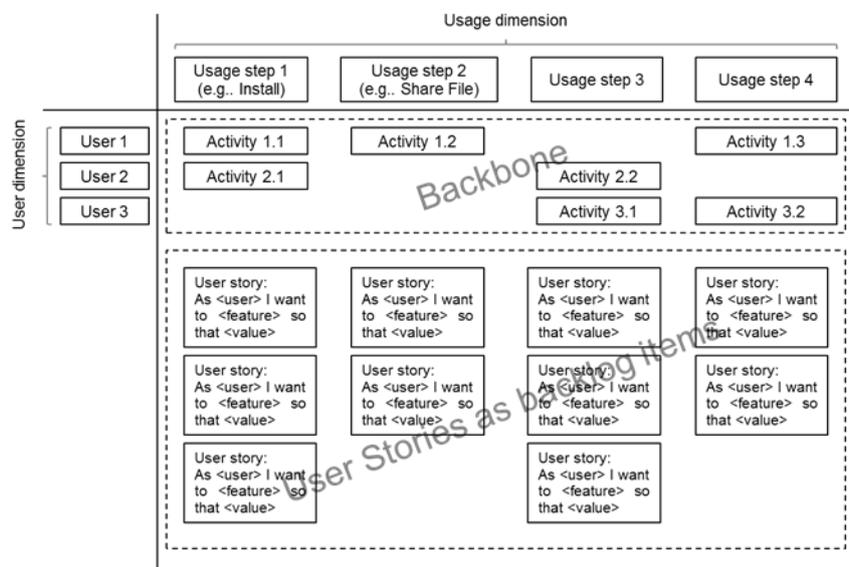

Fig. 1. User story map

3.2 A bottom-up approach for the definition of domain concepts

If we consider the USM to be the first step of gathering information regarding the domain of interest than we can create an algorithm for building a complete and structured knowledge base, described in Figure 2:

- Step 1: Apply the USM method, based on the business requirements and the project vision.
- Step 2: Gather other sources of information (standards, past experience...) in order to collect some generic and specific concepts, with respect to the scope resulted from the application of USM (Step 1).
- Step 3: Create an unique list of concepts that covers entire domain based on usages, roles and activities resulted through the application of USM and the generic concepts collected within the previous step.
- Step 4: Define relations and dependencies among the list of concepts.
- Step 5: Create a dynamic knowledge base covering the domain, expressed in some of the standard formats like relational data base, ontology, semantic model.
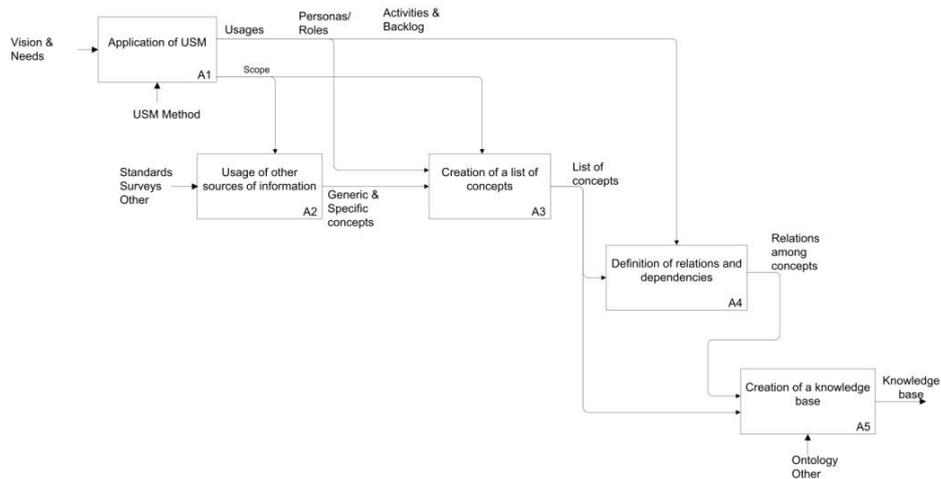

Fig. 2. Diagram displaying the creation of knowledge base

Obviously, in this scenario, USM is vital part as it will create a base view of the domain in question. Switching from USM to list of concepts is relatively straight-forward step. Functionalities required by user stories are described in form of sets of functional modules and each module is translated into concept of the domain. Next, the list is extended with additional concepts coming from other sources of information like industrial standards or similar projects. Finally, concepts are described using relations and expressed in some of the usual knowledge base formats.

4 Application to the LinkedDesign Project

USM method gives a great approach on "end user" request specification. In a real-life scenarios, when creating a new software, there is always a dilemma between creating a generic product which can be used by everybody, but doesn't really cover anybody's needs completely or creating a strictly end-user dictated custom product. USM method gives a solution for such problems since it allows a controlled generalization of user requests. This method for request specifications is developed for the scenario where some of the "end users" are already known and the product is developed according to their specifications, but since the controlled generalization of these requirements is done, it is possible and quite straight forward for other future clients to use the software.

LinkedDesign[1] is a project of a FP7 framework which strives to create a platform which will store and enable sharing of data, knowledge management and fast and efficient querying for information for all the actors in project lifecycle phases of interest. In LinkedDesign project we are dealing with three industrial partners as use-cases whose fields of engagement are greatly diverse but can all benefit from a platform to be designed.

One of the questions that necessarily emerges is to what degree should some product be generic. The more generic it is, the wider is the specter of "end users" that will be able to use it. On the other hand, generic models require more work on implementing them for specific companies. One of the approaches is to gather USMs from all interested users and then merge them into one single USM by generalizing them only to such extend, that the final model is simple enough. For example, when developing a software for different kinds of manufacturing companies, user roles such as "Part designer" and "Designer" coming from two different companies should be merged into one simple role "Design engineer" and both users should be able to identify the connection. In LinkedDesign specifically this example is shown in Figure 3.

In some other cases generalizing is not that simple. For example, if only one of all users wants to have the role "Supplier", it has to be carefully considered should such role be included in the generic model, or should it be taken under "Manufacturing personnel" which everybody requires. The later choice comes as better solution since it will keep our list of generalized roles simple and USM method gives us opportunity to note the work of "Suppliers" through a user story of more generic role. These examples lead us to a conclusion that during this merging process, the most important thing is to have constant communication with all "end users". As long as they are able to recognize all their user stories in these merged, more generic roles, the model is not too generic.

---

[1] http://www.linkeddesign.eu/
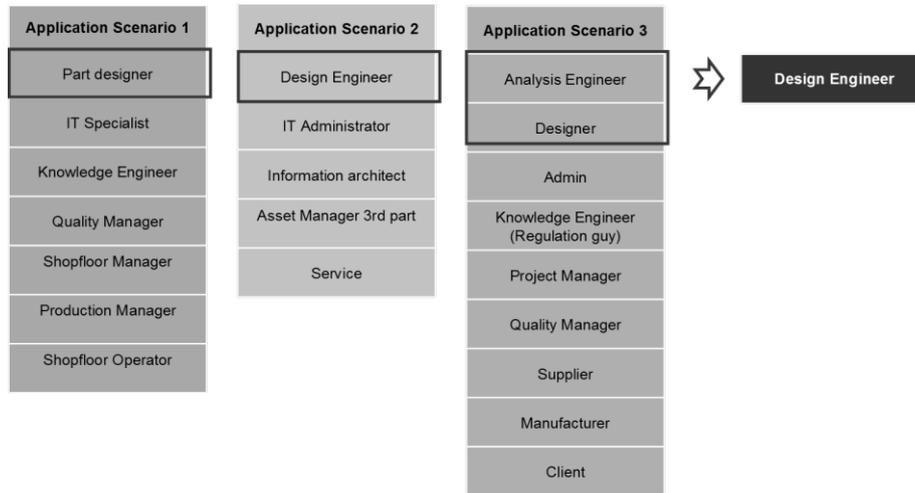
Fig. 3. User roles merging

Same guidelines stand for creating an optimal set of generalized functions, though in this case situation gets a bit more complicated as a consequence of difference in the nature of manufacturing that different clients implement. For example function "Install" can be required by almost all clients but since there is no standardized terminology, "Install" can be used to describe installation of a knowledge base platform, overlooking the manufacturing, or installation of a sensors overlooking the physical machinery. Function "Report" of a project manager is another one that can be expected from all clients but depending on a case, it can be a request to be able to report feedback about product to a manufacturing department for adjusting the protocols or it can be a request to be able to report final statistics to a executives department for a future planning. This would be a problem of using the same word for different actions but also the problem of using different names for one action has to be considered. For example simple data processing can be described as "Analytics" or "Knowledge extraction" depending on a client. The only way to deal with these issues is again to have a continuous communication with all the clients and to make sure that terminology is well defined and understood.

General guidelines on how this process should be conducted are given in Figure 4. Still, it is important to remember that this is a manual procedure and that human perception and creativity are crucial for success.
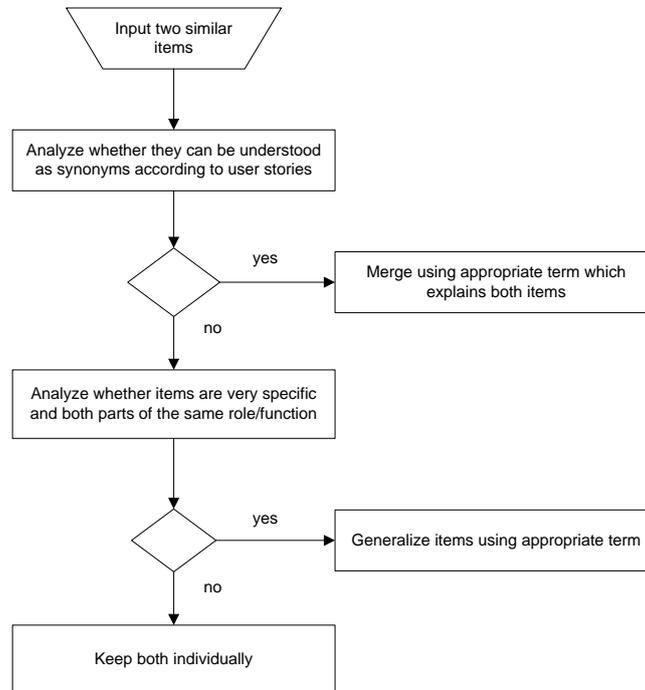
Fig. 4. Algorithm for USM merging

Another important issue to consider is that all actors have to fully understand what they are being asked for, by this USM method. In other words, they have to be able to describe their demands fully and in a detail manner and to be encouraged to consider all their employees and their activities. Close communication is again answer for this matter, since it will also lead to creating a more synchronized vocabulary and better understanding of terms used to describe different activities. Finally, the best results should be achieved through iterative procedure of merging which is finished when final model is simple enough and it covers all demands for details of different users.

Once the generalized USM map is created (shown in figure 5), the procedure toward creating the LinkedDesign ontology follows the proposed "five step" procedure presented in Section 3.2. Currently, it is in the stage 3 where the list of concepts is being created, some of them are given in the following table.

Top Level Concepts	Description
Actor	Groups all the personnel involved
Task	Groups scheduled actions
Process	Groups all processes
Product	Groups all the details of a products

Resource	Groups all the required elements for product manufacturing
LCP	Groups life-cycle phases of the product
Factor	Groups relevant issues related to product which need to be examined
Component	Groups the components of a product

Table 1. Top level concepts

5   Conclusion

Identification and representation of knowledge in a product related domain is still a great challenge. In this paper we propose the USM method for knowledge extraction as well as formal guideline for how to apply this method. On the LinkedDesign use-case we showed that it is convenient and efficient, successful in giving the expected results. The USM method gives us a tool which can directly translate raw data into list of relevant concepts that covers entire functional profile of the software in question and hus it gives us a detail image of the domain this software operates on. It is simple and straight-forward and it enables end-users to express their descriptions of the domain in a common everyday language, rather than using technical terms, which is more probable to lead to gathering of more detailed information. Beside formalization of the information, it is shown to be an excellent tool for generalization of "end-user" requests and a vital step toward the creation of the knowledge base.

	LEAP Env. setup	Data Collection & Knowledge Capture	Analysis & Control	Knowledge Exploitation & Management	Reports & Decision Making
Design Engineer		Gather requirements and load data; Gather information about reusability within LEAP of previous models	Analyze and update CAD models	Identify causes of critical points and define solutions; Consult recommendations engine (on experts, standards....)	Make and use reports
IT Specialist	Setup & Configuration of LEAP system; Document requirements for interoperability; Maintain the system (upgrade); Install Leap Adapters to existing systems	Manage data collection			Make and use reports
Knowledge Engineer	Maintain & Update Rules basis; Document knowledge	Connect data sources; Update rules basis	Maintain rules/documents	Maintain Knowledge; Show rules usage	Make and use reports
Project Manager		Define data sources	Perform an analysis of current data; Control changes	Get an overview of the overall status	Make and use reports
Manufacturing Member			Give feedback to LEAP		Make and use reports
Supplier		Share specifications of components			
End Product Annotator		Collect field/user data		Review end product	Record users feedback

Fig. 5. The generalized USM

References

1. P. F. Drucker, *Post-capitalist society*. Harper Paperbacks, 1994.
2. I. Nonaka, "The knowledge creating company," pp. 96-104., 1991.
3. T. H. Davenport, S. L. Jarvenpaa, and M. C. Beers, "Improving knowledge work processes," *Sloan management review*, vol. 37, no. 4, pp. 53–65, 1996.
4. K. E. Sveiby, *The new organizational wealth: Managing & measuring knowledge-based assets*. Berrett-Koehler Pub, 1997.
5. A. Gunasekaran, "Agile manufacturing: a framework for research and development," *International Journal of Production Economics*, vol. 62, no. 1-2, pp. 87–105, 1999.
6. R. D. Sriram, "Artificial intelligence in engineering: Personal reflections," *Advanced Engineering Informatics*, vol. 20, no. 1, pp. 3–5, 2006.
7. W. Skarka, "Application of MOKA methodology in generative model creation using CATIA," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 5, pp. 677–690, 2007.
8. C. Van der Velden, C. Bil, X. Yu, and A. Smith, "An intelligent system for automatic layout routing in aerospace design," *Innovations in Systems and Software Engineering*, vol. 3, no. 2, pp. 117–128, 2007.
9. G. Colombo, A. Mosca, and F. Sartori, "Towards the design of intelligent CAD systems: An ontological approach," *Advanced Engineering Informatics*, vol. 21, no. 2, pp. 153–168, 2007.
10. J. Kulon, D. Mynors, and P. Broomhead, "A knowledge-based engineering design tool for metal forging," *Journal of materials processing technology*, vol. 177, no. 1, pp. 331–335, 2006.
11. M. Franke, P. Klein, L. Schröder, and K. D. Thoben, "Ontological semantics of standards and plm repositories in the product development phase," in *Global Product Development. Proc. 20th CIRP Design Conference*, 2010, pp. 473–484.
12. I. Nonaka and H. Takeuchi, *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. Oxford University Press, USA, 1995.
13. D. A. Wilson, *Managing knowledge*. Butterworth-Heinemann, 1996.
14. J. B. Quinn, P. Anderson, and S. Finkelstein, "Managing professional intellect: making the most of the best," *Harvard Business Review*, vol. 74, no. 2, pp. 71–80, 1996.
15. T. A. Stewart, *Intellectual Capital: The New Wealth of Organizations, New York: Bantam Doubleday Dell Publishing Group*. Inc, 1997.
16. R. E. Bohn, "Measuring and managing technological knowledge," *Sloan Management Review*, vol. 36, no. 1, pp. 61–73, 1994.
17. T. H. Davenport and L. Prusak, *Working knowledge: How organizations manage what they know*. Harvard Business Press, 2000.
18. K. M. Wiig, "Knowledge management: where did it come from and where will it go?," *Expert systems with applications*, vol. 13, no. 1, pp. 1–14, 1997.
19. *Jeff Patton, " It's All In How You Slice It ", Better Software 2005.*
20. *M. del Carmen Suárez, "NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse", Thesis, 2010*