Chapter
Methods and tools for knowledge sharing in product development

Marco Bertoni[1], Christian Johansson[2], Tobias C. Larsson[3]

Abstract. The emerging industrial business partnerships, which feature cross-functional and cross-company development efforts, raise the barrier for the establishment of effective knowledge sharing practices in the larger organization. This chapter aims to highlight the role of knowledge as a key enabler for effective engineering activities in the light of such emerging enterprise collaboration models. Knowledge Enabled Engineering (KEE) is presented as an approach to enhance the extended organization's capability to establish effective collaboration among its parts, in spite of different organizational structures, technologies or processes. KEE is analysed in its constituent parts, highlighting areas, methods and tools that are particularly interesting for leveraging companies' knowledge sharing capabilities.

1.1 Introduction

The development process, as all work processes, involves some combination of four types of "work" (Figure 1), which can be thought about as a continuum [1]. At one end of this continuum we find the entirely independent effort of an individual to produce some

---

[1] M. Bertoni
Luleå University of Technology, SE-97187, Luleå, Sweden
e-mail: marco.bertoni@ltu.se

[2] C. Johansson
Luleå University of Technology, SE-97187, Luleå, Sweden
e-mail: christian.johansson@ltu.se

[3] T.C. Larsson
Luleå University of Technology, SE-97187, Luleå, Sweden
e-mail: tobias.c.larsson@ltu.se

kind of product or service. Next, we find dependent work, where one applies some level of effort to make use of someone else's product or service. Along this continuum, participants need to share information across disciplines, and other barriers, to achieve a mutual objective. Eventually we find the full integration of knowledge acquired through specialists in a range of disciplines, which enables a team to optimize the work to be performed.

Collaboration is a term mainly referring to the third stage of the above continuum, i.e. to "people working together on common tasks, or to the aid extended to a thing or a person to produce or create something new" [2], just as a blacksmith and a glassblower can collaborate on a piece of art for mutual benefit. Fundamentally, it is based on the idea that working together will allow collaborating partners to create something superior to what any one entity could have created alone.

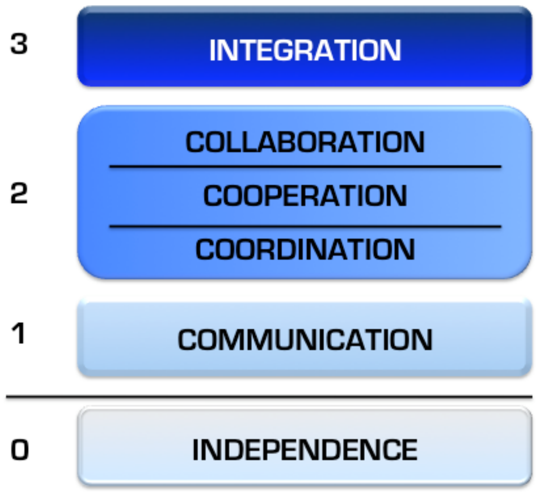


Figure 1: The four levels of interaction

This concept is of high interest both in the business-to-business and business-to consumer industry [2]. In the latter, customer service manufacturers may try to develop their collaborative skills to better reach out to their customers and to maintain constant contact with them. On the other hand, collaborative product development processes, involving people, processes and technologies across multiple organizations working in the same line of business, is becoming the industry standard in light of globalization and outsourcing. True collaboration is still a pipe dream for large product development projects. The difficulties recently encountered by, for example, the aeronautical industry [3][4] clearly highlight that effective collaborative development processes requires a quantum jump in the way information and knowledge is shared in the extended organization.

1.2 Motivation and objectives

The ultimate goal of any enterprise collaboration means is to support a more effective and trustworthy decision-making. In essence, "*it's all about getting the right information to the right people at the right time so they can do their jobs more effectively*" [5]. To enable conscious decisions, enhanced knowledge sharing methods and tools are needed to make sure that all design stakeholders contribute with their experience and skills in such decision-making processes, while protecting intellectual ownership.

The main objective of this chapter is to highlight the role of knowledge as a key enabler for effective engineering activities, in the light of the emerging enterprise collaboration trends.

The chapter initially describes the different levels of interaction that are featured in modern business partnerships, highlighting the difficulties in establishing effective collaboration, and consequently knowledge sharing practices, when dealing with cross-functional and cross-company development efforts.

The chapter further introduces the concept of Knowledge Enabled Engineering (KEE), an acronym that summarizes the intent of providing the extended organization with the capabilities to establish effective knowledge collaboration among its parts, in spite of different organizational structures, technologies or processes. KEE is eventually analysed in its constituent parts, highlighting areas, methods and tools that are particularly interesting for leveraging companies' knowledge sharing capabilities.

1.3 Four Levels of Interaction

Product development processes can be differentiated on the basis of the interaction required for addressing the common development target. Due to cost-benefit considerations the intensity and design of integration have to be based on the requirements of the existing development situation [6]. It is possible to outline four major levels of interaction between organizations and teams (Figure 1):

1. Level 0: Independence means that companies do not interact in any form during the development tasks and do not share any kind of data or information during the process.
2. Level 1: Communication refers to the process of transferring information from one source to another, and is at the basis of any collaborative task. It simply refers to the act of exchanging information by the use of some kind of media and does not imply working towards a common goal.
3. Level 2: Coordination, Cooperation and Collaboration refer to the association of a number of persons for their common benefit and to collective action in pursuit of a common goal. Autonomous, independent, federated and loosely coupled systems can exchange services while continuing their own logic of operation, similarly to virtual enterprise [7] agreements.
4. Level 3: Integration refers to the concepts of coherence and standardisation indicating a 'tightly coupled' system where components are interdependent and cannot be separated. This level points to the more traditional single-OEM structure; a homogeneous organization, composed by interdependent parts and standard languages, methods and tools.

It is possible to define three levels of collaboration intensity for product development: coordination, cooperation and collaboration [6]. Which type should be chosen depends on the required interaction intensity caused by the interaction need of a given development situation, and typically increases with tighter process-related interdependencies between the subtasks, and with the necessity of integrating the knowledge of the development partners.

- Coordination is the act of managing interdependencies between activities [8] and the support of interdependencies among actors [9], regulating diverse elements into an integrated and harmonious operation to accomplish a collective set of tasks.
- Cooperation, similarly, refers to the association of a number of persons in the pursuit of common goals [10], able to adhere to some kind of dialogue rules [11], i.e., setting out maxims (of quantity, quality, relevance and manner) to which it can be assumed parties in a dialogue are adhering on an utterance-by-utterance basis [12].
- Collaboration is a special case of cooperation and implies the joint execution of one task where the individual, private aims of the participants are more closely aligned with each other. People need to work together to reach the desired outcome rather than that outcome being achieved through 'selfish' participation constrained by contextual factors [11]. Collaboration may be seen as the highest degree of interdependency between autonomous and independent parties.

1.4 Emerging enterprise collaboration models

In those areas where products are complex and require large investments it is common for companies to collaborate in joint ventures where risks, costs, revenues and profits are shared among the partners. In such a context, two different types of partnerships are of particular interest with a focus on collaboration: the Extended Enterprise (EE) [13] and the Virtual Enterprise (VE) [6]:

The EE is a form of joint venture where one company, usually named Original Equipment Manufacturer (OEM) is in charge of integrating the parts of a product, produced and designed by satellite companies, into the whole [14]. The objective with forming an EE is building long term relationships across the value chain, both to share risks, knowledge, and resources [15] and to reduce costs and time-to-market [16].

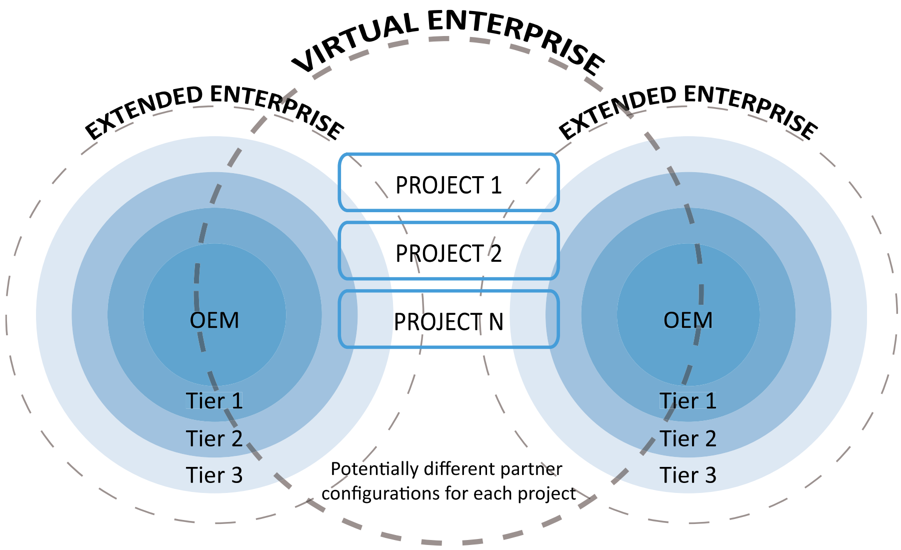


Figure 2: Virtual Enterprise vs. Extended Enterprise [17]

The VE spans over multiple EEs and involves several different OEMs, which are asked to temporarily collaborate to design and manufacture new products. The VE partners share costs, skills and core competencies to deliver solutions that none of the individual partner could have done on their own [18]. Figure 2 visualizes the relationship between the EE and the VE.

While the EE focuses more on long-term partnerships, the VE is a very short-term and flexible endeavour [19], where companies participate for a shorter time span to satisfy a business opportunity relatively quickly [6][13]. The cross-organisational nature of the VE allows companies to be flexible with the use of resources [13], moving people in and out of the individual organisations as the conditions changes and different competencies are in demand.

1.5 Barriers and inhibitors for effective knowledge sharing

Knowledge sharing is a key aspect to any strategic alliance, since fundamentally, companies enter these partnerships to tap into the knowledge bases of other companies to expand the offerings they can make. Managing such knowledge is one of the most crucial aspects for the successful implementation of a virtual working paradigm. However, a number of collaboration barriers, which are often not visible in a co-located and "physical" organization, inhibit the EE and VE effectiveness to share knowledge across teams, functions and companies. Considering Allen's [20] observations that people are likely to seldom collaborate if they are more than 50 feet apart, the challenges of globally distributed organizations are immense.

As Brown and Duguid [21] point out, *"knowledge usually entails a knower"* (p.119), i.e. knowledge sharing relies on humans taking on the role as creator, carrier, conveyor and user. This distinguishes knowledge and information sharing, which can usually happen "outside" humans and without the direct influence of human beings. Information is normally treated as independent and more-or-less self-sufficient, whereas knowledge is usually associated with someone (e.g. "where is that information" vs. "who knows that?"). In a virtual organization the right people to ask (knowledge owners, hidden experts, lead users, etc.) are even more difficult to locate than in a traditional (co-located) engineering situation [17].

Furthermore, work practices may differ substantially between different parts of an organization, which generate strong needs to share knowledge-of-practices and be able to locate people with credibility [17]. In co-located teams, knowledge-of-expertise is usually built up naturally and quickly, through the interactions with others. People gradually learn about the strengths and weaknesses of their colleagues, and when faced with a problem they often have a very good idea of whom to approach for help.

Also, engineers must be able to trust the information they receive – so called trust-in-expertise [17]. This trust fundamentally boils down to a trust in capacities and abilities, and people need to make continuous assessments about whether or not they can trust on recommendations, advice etcetera that they colleagues offer. Another common problem of cross-site collaboration is the delay in resolving work issues [22], due to the difficulties of finding the right person to talk to, initiating contact, and discussing possible ways to deal with the issue. As physical distance increases, tacit knowledge (e.g. intuition, individual perception, rules of thumb) is not spread as easily through informal communication channels, and must therefore be made clearer to be used throughout the company. Figure 3, adapted from Lipnack and Stamps' [23], illustrates some of the barriers related to "virtual distance".

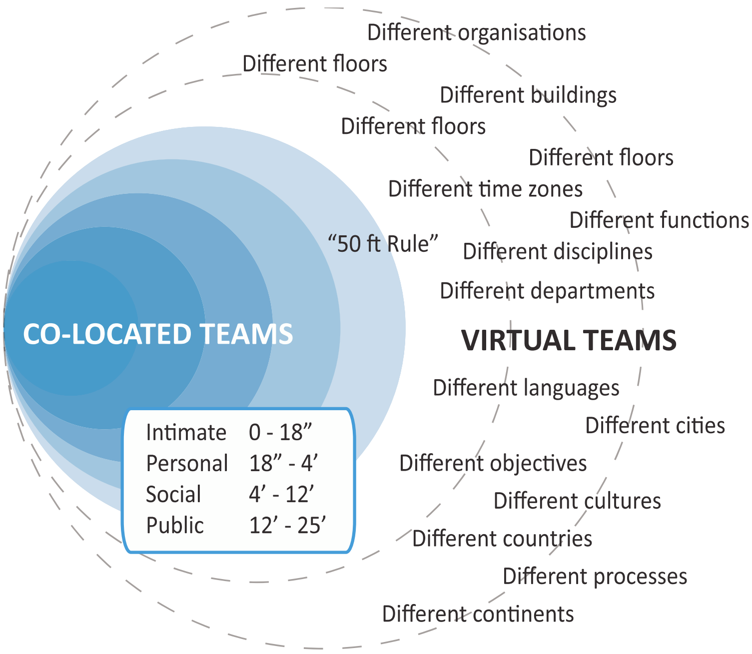


Figure 3: Virtual Distance [17]

Moving to company perspective, a main issue is finding the best trade-off between sharing and shielding core knowledge assets. The "coopetitive" [24] nature of the VE arrangements makes knowledge sharing problematic, because knowledge used for cooperation may also be used for competition in other projects [25]. Companies feel hesitant to share knowledge if they feel that what they gain is less than what they give away [26], so the problem is to understand what has to be shared and in what form to sustain the project without losing the main source of competitive advantage.

Last, but not least, the loss of face-to-face interaction have an inhibiting effect on morale, loyalty and performances of employees [27], who may "interact" but not collaborate. If on one side technology can cross boundaries, people often may not. Plenty of social and behavioural issues have yet to be solved in order to implement effectively the virtual working paradigm across organizations.

1.6 KEE: Enabling engineering activities through knowledge

In an ideal world, companies should be able to use their collective knowledge to make better and faster decisions, reducing lead-time and improving robustness of strategic engineering activities. Engineers should concentrate on the more intellectual parts of engineering work, rather than spending time doing dull and cumbersome routine work.

The term Knowledge Enabled Engineering (KEE) [28] collects a wide range of methods and tools that intend to enable such ideal scenario, supporting a knowledge sharing activity that spans long product lifecycles within multi-disciplinary, multi-company and multi-cultural environments, across supply chain relationships [29]. The main purpose of KEE is to make as much knowledge as possible available early in the product development process, to allow more design iterations via simulation [30], reducing the lead-time for tedious analysis model generation and thus allowing more design concepts to be studied [31].

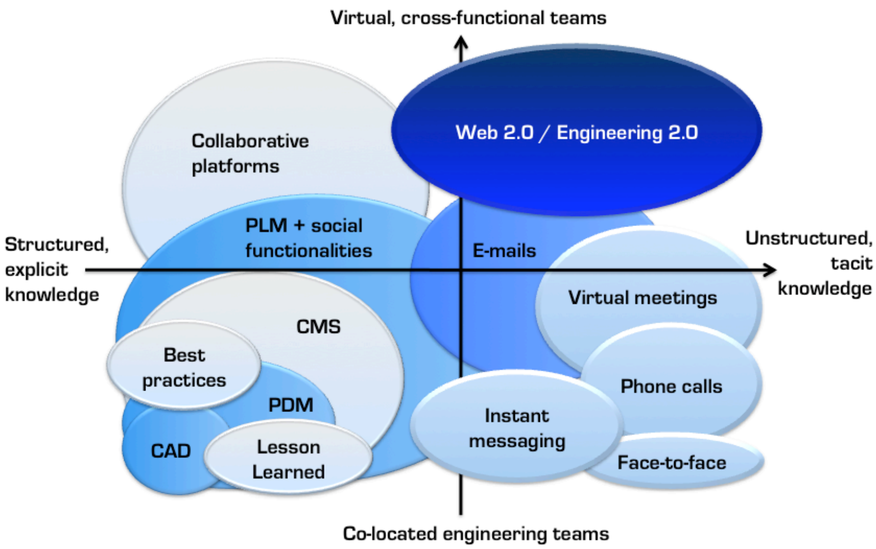
KEE originates from the work done in the area of Knowledge Engineering (KE) and Knowledge Based Engineering (KBE). However, while KBE is often associated with commercial systems providing demand driven, object oriented programming languages for rule-based execution, KEE is intended as a more generic and methodological-oriented term [32] with emphasis on context and rationale. KEE systems are intended to support three main tasks: 1) promoting the use of fitting-for-purpose models to support the capturing of engineering knowledge in different engineering activities, 2) supporting an iterative process where the capturing of engineering knowledge and the automation of the engineering activities is done simultaneously, and 3) capturing the underlying process of generating and evaluating solutions concepts in a computerized system to ensure repeatability and improve the quality of the output.

The KEE research aims to develop, validate and exploit methods to identify, model, store, retrieve, reuse and share product/service relevant knowledge spanning long product life cycles. The following sections aims to spotlight the most crucial areas to leverage the company's capability to enable engineering activities through a better use of their knowledge assets.

1.6.1 Enabling informal knowledge sharing between dislocated design teams

The advent of Web 2.0 technologies has brought a new culture of sharing information on the Web where users can actively create, store, edit, access, share and distribute the content to larger audiences. This ‘bottom-up’ sharing mood is opposed to the predefined, ‘top-down’ structure of most of the existing knowledge management tools, pushing each individual to maintain their own space for which they have complete control over the information they likes to share.

Moving from McAfee’s [33] concept of Enterprise 2.0, Larsson et al. have coined the term “Engineering 2.0” [34], which borrows the more general Web 2.0 concept and translates it into a cross-functional engineering context. Engineering 2.0 promotes the use of lightweight (compared with traditional CAD/PDM/PLM) and bottom-up tools to support informal knowledge sharing across functions



and companies in a VE setting (Figure 4).

Figure 4: Engineering 2.0 positioning

Lightweight because the purpose is to develop and implement solutions that require little time and effort to setup, use and maintain. Bottom-up because they do not impose a pre-defined structure, but rather lets structures evolve over time as an almost organic response to the activities, practices and interests of the knowledge workers. Weblogs, wikis, social networks, RSS feeds, tagging, microblogs, instant messaging, discussion forums, social bookmarking, and mash-ups are few examples of Web 2.0 technologies that can be successfully adopted in an engineering setting.

A recent McKinsey survey [35] has shown that 2/3 of 1700 companies interviewed worldwide have investigated or deployed Web 2.0 tools to support their product development activities. However, their use is limited today to a comparably small part of the entire development process, mainly with the intent of gathering ideas and feedback on the product from the customer base [36].

Web 2.0 tools are used mainly to enable effective communication within dislocated teams, particularly in the initial stages of a software development project [37]. The Microsoft Quest internal communications system [38], the wiki-like environment proposed by Ciavola et al. [39] and IBM Dogear [40] are a few examples of social software supporting learning, sharing and collaboration between researchers and professionals in the design activity.

Lightweight approaches have been also proposed to support product development projects' documentation [41], e.g. using wiki-style collaboration tools to create assessment reports [42] or to maintain rule-based systems as they grow [36]. Furthermore, several groups are proposing ways to complement CAD/PDM/PLM tools with social functionalities, leveraging social interaction and collaborative features, among global design teams, Vuuch [43], for instance, is a plug-in for Pro/ENGINEER or Dassault Systemes' SolidWork, to initiate, monitors, and manages design discussions directly from the CAD environment to organize design discussions by associating them to the product Bill of Material (BOM).

1.6.3 Enabling fit-for-purpose reuse of past design knowledge

The key element to successful knowledge reuse is to understand a designer's reuse intention [44]. The designer's information seeking behaviour, in fact, depends on task-dependent procedures [44]. It means that the usual search queries alone are not sufficient to address the designers' information needs. Rather, if the context for the query is known, it is possible to anticipate the type of result that will be useful, and refine the query accordingly, providing more tailored knowledge to people with similar profiles [45]. Context-aware applications may help to increase the relevance of the knowledge they retrieve to support a given task.

Context-awareness can be defined as "*the ability of a computing device to detect and sense, interpret and respond to aspects of a user's local environment and the computing devices themselves*" [67]. Context-aware system have gained popularity in the 1990s to address problems raised by the usage of mobile terminals [47], with the purpose of 1) retrieving information and execute commands for the user manually based on available context, 2) automatically executing a service based on the current and available context, 3) tagging the context to information for later retrieval. Context may refer, therefore, both to a physical dimension (time, space, device) and internal/logical dimension, i.e. the user's goals, tasks, work context, business processes, the user's emotional state [48].

One of the most notable examples of Context-aware applications supporting collaborative product development is the Knowledge-Enabled Solution Platform (KESP) [49] developed within the EU FP6 VIVACE project [28]. The KESP is an intelligent knowledge assistant that automatically provides the engineer with the contextualized Knowledge Elements (K-El) that s/he needs during his/her daily design activity. The KESP is both a multi-source context-based application and a self-learning software system that enables the user to perform multi-source searches for all the K-Elements applicable to his or her engineering context. It describes the engineers' profile by using six contextual dimensions (Product, Process, Project, Gate, Role, Discipline) and enables to manage links between knowledge and contexts of applicability, relying on similarities between contexts to provide the knowledge workers K-El relevant for their task. The driving idea behind the KESP is the possibility to associate a K-Element with the most suitable context in which it should be applied. For some K-El, such as norms for instance, it may be quite straightforward to define beforehand the domain of applicability. For others it may be more complex. To overcome this obstacle, the KESP provides a way to learn the relationship between a K-Element and the description of the context in which it has been applied, so making easier for engineers to understand what documents/information is more relevant for their job.

1.6.4 Enabling effortless management of design rationale

A recent study [50], presenting the results from a UK survey about requirements of managers and engineers in design and service, has shown that "rationale" (34.9%) is the most mentioned category in terms knowledge and information needs in engineering. Understanding the reasons why a system has been designed in a certain way, or the information about what design options have been considered but rejected is necessary to understand, recreate, or modify a design, although rarely adequately captured in a systematic and usable format, [51], but rather scattered throughout a collection of paper documents, notebook entries, and the memory of the designers [52].

Design intent capture systems aim to capture and store information about past design solutions and their rationale in a repository that is independent of human memory [53]. They can be divided into solutions aiming to "communicate" or "document" design knowledge [51]. The first category [54] aims to capture all communications among team members during design meetings, but do not encourage the synthesis and retrieval of high-level design decisions (assumptions, constraints, philosophies) [51]. The second category documents design decisions to enable people outside the project group to understand, supervise, and regulate what is done by the team [55], although not usually capturing why a particular design was not chosen [51]. Furthermore, automatic approaches assume aim to capture the communication among the designers and design teams or between the designer and the design support system without asking user intervention [56]. Manual tools, instead, requires the direct intervention of a user/designer to record the rationale.

It is also possible to distinguish between "process-oriented" approaches [57], supporting complex system designs and based on Issue Based Information System (IBIS) [58] framework, and "feature-oriented" approaches, used to support automated reasoning [59].

A plethora of approaches, from ontology definition languages [60], to operation-mining algorithms [61] to soft computing techniques [62] has been proposed for enabling the effective capturing of design rationale. The most recent developments in this area includes the Design Rationale editor (DRed) [54], derived from the IBIS concept, which proposes a simple and unobtrusive approach that allows engineering designers to record their rationale as the design proceeds. A different approach is proposed by CoPe_it! [63], which support argumentative collaboration on the Web. The use of automated design rationale capture systems to improve collaborative co-design CAD environments has been also extensively studied [64] and several attempt have been made to capture design knowledge and to represent it in a single-user CAD environment [65].

Knowledge Enhanced Notes (KEN) [66] is a system enhancing the capturing of discursive and collaborative aspects of synchronous design activities. COHERE [67] emerges from work in issue mapping and design rationale, to propose a social, semantic annotation tool focused on the construction and management of connections — between data, knowledge resources (such as documents), ideas (including issues, options and arguments), and people.

1.6.5 Enabling downstream knowledge capturing

When designing a new product or service, it is increasingly important to have a clear picture of a wide variety of customer needs to identify opportunities to improve the current offer both from a product and service perspective. Enhancing the interaction with the end users to use their knowledge to drive the preliminary design activities is a dream for many manufacturers, which the latest achievements in the area of communication technologies can make true.

Crowdsourcing [68] represents nowadays one of the most promising approaches to capture downstream knowledge for the benefit of product development. Crowdsourcing essentially means outsourcing a task to a large group of people or to a community (a crowd) through an open call. The basic principle is that online consumer communities greatly add value to new product development [69], thus Web 2.0 can leverage the critical role that customers and the crowd play in the innovation process [70]. Crowdsourcing is hugely popular in the software development domain. Dell IdeaStorm [71] represents an example of how idea crowdsourcing may be exploited in an early design stage of new products. Several top companies, such as Microsoft, Apple or IBM, have made extensive use of social media to share ideas with lead users ahead of beta testing [72]. In manufacturing, Web 2.0 applications have been used to gather innovations for both products an services, sometime even by means of virtual prototypes online [73] or SecondLife® extensions [74] to analyse patterns of usability and preferences.

1.6.6 Enabling knowledge maturity assessments

In product development, many problems are intrinsically wicked [75] or ill-defined [76], hence making a decision is often about settling for what is good enough rather than waiting for the optimal solution to emerge, e.g. because of flawed or missing information [77] [78]. Rational decision making [79] is de-facto a rare occurrence. Methods and tools are needed, therefore, to provide decision makers with a deeper understanding of the status of the knowledge base on which they draw decisions. The Knowledge maturity concept is interesting to boost decision makers' confidence when information and knowledge is in limited supply. Grebici et al. [80] denote maturity as the distance between the level of completeness relative to what should be the level of completeness, i.e. as-is status versus to-be status, thus being the relative state of the development of a piece of information with respect to achieving a purpose. For example, a preliminary analysis result concerning the heat tolerance of an aero engine component may be good enough in the feasibility stage of designing the component, whereas the same numbers may be too inaccurate to be of value in the detail design stage.

The concept of knowledge maturity [81] is explored to leverage KEE practices, providing a practical decision support that increases decision makers' awareness of the knowledge base on which the decisions are based, and to support cross-boundary discussions on the perceived maturity of available knowledge. The knowledge maturity model [81] computes the state of readiness of a knowledge asset using a narrative scale over three dimensions: input, method (tool), and expertise (experience) on a scale from 1 to 5. A rank as 5 indicates an excellent knowledge maturity, meaning that content and rationale have been tested and proven and reflect a known confidence, that the procedure to produce the content and rationale reflects an approach where proven methods are used, where workers continually reflect and improve and where lessons learned are recorded. On the other end, a knowledge maturity level ranked with 1 means that the con-tent and rationale is characterized by instability (e.g. poor/non understanding of knowledge base) and the procedure to produce the content and rationale is dependent on individuals and formalized methods are non-existent. In between these levels there is a continuous increase in detailing, documentation and standardization.

The knowledge maturity allows worldwide-located teams to have a shared artefact around which they can identify and discuss issues of concern, visualize the current status of the knowledge base, and negotiate a shared understanding of the advantages and drawbacks with the available knowledge base.

1.7 Conclusion

Engineering work is little-by-little changing to include a larger part of the value creation activities in virtual and cross-functional development processes. At the same time, the definition of an engineer is changing as well. Engineering organizations are hiring younger engineers who think and work differently, and who must learn to utilize their knowledge, capabilities and talent in more open and collaborative way.

These aspects raise inevitable questions on the role of humans in product development. Are they engineers? Are they product developers? Are they function providers? Are they knowledge workers? The engineering culture, which changes slowly in established firms, is struggling to adapt to the relatively rapid changes of targets, methods and workforce.

In the new context, engineers are explicitly interested in avoiding redundancy, and instead seek novelty and innovation rather than well-known knowledge. Essentially they are looking for new ways to deal with new problems they are likely to face when moving into the development of complex, lifecycle-oriented products in a distributed context. What appears evident from the research is that these new problems are not likely to be adequately addressed by the "traditional" information and knowledge sharing systems alone. In spite of the hugely important work being done in the CAD/PDM/PLM arena, a quantum leap in methods and tools is required to satisfy the need for knowledge of the new, distributed, digital, cultural diverse, participatory and social responsible generation of engineers.

The research in the area of Knowledge Enabled Engineering aims to address such need. The purpose of KEE is twofold: facilitating the work of the "knowledge engineer", the person in a company who is responsible for 'engineering' knowledge bases, as well as the work of the "knowledge worker', of who use such knowledge for accomplishing his/her everyday tasks. In the end, KEE is all about enabling more effective and efficient engineering activities, in an increasingly challenging product development environment, through providing engineers the knowledge they need, in the form they need and at the time they need it.

References

1. http://www.di.net/articles/archive/2437. Accessed September 2009
2. Mathew G.E.: The State-of-the-Art Technologies and Practices for Enterprise Business Collaboration – An Overview. An Infosys Technologies ltd report (2002)
3. Greising D., Johnsson J: Behind Boeing's 787 Delays. Chicago Tribune, December 8th (2009).
4. http://www.airbus.com/en/presscentre. Accessed December 2010
5. McElroy M.W.: The New Knowledge Management: Complexity, Learning, and Sustainable Innovation. KMCI Press/Butterworth-Heinemann, Burlington (2003)
6. Kern E.M., Kersten W.: Framework for internet-supported interorganizational product development collaboration. Journal of Enterprise Information Management Vol. 20 (5), pp. 562-577 (2007)
7. Davidow W.H., Malone M.S.: The Virtual Corporation: structuring and revitalizing the corporation for the 21st century. Harper Business, New York (1993)
8. Malone T.W., Crowston, K.: The interdisciplinary study of coordination. ACM Computing Surveys, Vol. 26(1), pp. 88–119 (1994)
9. Bordeau J., Wasson B.: Orchestrating collaboration in collaborative telelearning. In Boulay, B. and Mizoguchi, R. (Eds), Artificial Intelligence in Education, IOS Press, Amsterdam, pp. 565-567 (1997)
10. Reed C.A., Long, D.P.: Collaboration, Cooperation and Dialogue Classification. Proc. IJCAI 1997, Nagoya, Japan (1997)

11. Walton D.N., Krabbe, E.C.W.: Commitment in Dialogue: Basic Concepts, Concepts of Interpersonal Reasoning. State University of New York Press, New York (1995)
12. Grice H.P.: Logic and conversation. In Cole, P., Morgan, J.L., (eds): Studies in Syntax. Vol. III, Seminar Press, New York (1975)
13. Browne J., Zhang, J.: Extended and Virtual Enterprises - similarities and differences, International Journal of Agile Management Systems. Vol. 1(1), pp. 30-36 (1999)
14. Ericksen P.D., Suri R.: Managing the Extended Enterprise. Purchasing Today, Vol. 12 (2), (2001)
15. Boardman J.T., Clegg, B.T.: Structured Engagement in the Extended Enterprise. International Journal of Operations & Production Management. Vol. 21(5/6), pp. 795-811 (2001)
16. Pardessus T.: The Multi-Site Extended Enterprise Concept in the Aeronautical Industry. Air & Space Europe. Vol. 3(4). pp. 46-48 (2001)
17. Larsson A.: Engineering Know-Who: Why Social Connectedness Matters to Global Design Teams. Doctoral Thesis, Luleå University of Technology, Luleå, Sweden (2005)
18. Hardwick M., Bolton, R.: The Industrial Virtual Enterprise. Communications of the ACM. Vol. 40(9). pp. 59-60 (1997)
19. Katzy B.R., Dissel, M. 2001. A toolset for building the virtual enterprise. Journal of Intelligent Manufacturing, Vol. 12. pp. 121-131 (2001)
20. Allen T.: Managing the Flow of Technology. MIT Press, Boston (1977)
21. Brown J.S., Duguid P.: The Social Life of Information. Harvard Business School Press, Boston (2000)
22. Herbsleb J.D., Mockus A., Finholt T.A. Grinter R.E.: Distance, Dependencies, and Delay in a Global Collaboration. Proc. ACM CSCW 2000, Philadelphia, PA (2000)
23. Lipnack J., Stamps J.: Virtual Teams. John Wiley & Sons, New York (1997)
24. Brandenburger A.M., Nalebuff B.J. Kaplan R.S.: The Right Game: Use Game Theory to Shape Strategy. Harvard Business Review, (July-August), pp 57-71 (1995).
25. Loebecke C., Van Fenema P.C., Powell, P.: Co-Opetition and Knowledge Transfer. The DATA BASE for Advances in Information Systems. Vol. 30(2). pp. 14-25 (1999)
26. Appleyard M.M.: How Does Knowledge Flow? Interfirm Patterns in the Semiconductor Industry. Strategic Management Journal. Vol. 17., pp. 137-154 (1996)
27. www.smuts.uct.ac.za/~dliang/essays. Accessed September 2009.
28. http://www.vivaceproject.com. Accessed January 2011.
29. Sellini F., Cloonan J., Carver E., Williams, P.: Collaboration across the extended enterprise: barriers and opportunities to develop your knowledge assets?. Proc. TMCE 2006 Conference, Ljubljana, Slovenia (2006)
30. Sandberg M.: Knowledge enabled engineering design tools for manufacturability evaluation of jet engine components. Licentiate thesis 2005:16, Luleå University of Technology, Sweden (2005)
31. Isaksson O.: A Generative Modeling Approach To Engineering Design. Proc. ICED 2003, Stockholm, Sweden (2003)
32. Nergård H.: Knowledge Enabled Engineering Systems in Product Development: towards Cross Company Collaboration. Licentiate thesis 2006:37, Luleå University of Technology, Sweden (2006)
33. McAfee A.: Enterprise 2.0: The Dawn of Emergent Collaboration. MIT Sloan Management Review, Vol. 47(3), pp. 21-28 (2006)
34. Larsson A., Ericson Å., Larsson T., Randall D.: Engineering 2.0: Exploring Lightweight Technologies for the Virtual Enterprise. Proc. COOP 2008, Carry-le-Rouet, France (2008)
35. Bughin J.: How Companies are benefiting from Web 2.0. McKinsey Global Survey (2009)
36. Richards D.: Social Software/Web 2.0 Approach to Collaborative Knowledge Engineering. International Journal of Information Sciences, Vol. 179, pp. 2515-2523 (2009)
37. Maalej W., Happel H.J.: A Lightweight Approach for Knowledge Sharing in Distributed Software Teams. Proc. PAKM 2008, Yokohama, Japan (2008)

38. Patrick K., Dotsika F.: Knowledge Sharing: Developing from within. The Learning Organizations Journal, Vol. 14(5), pp. 395-406 (2007)
39. Ciavola B.T., Gershenson, J.K.: Establishment of an Open, Wiki-Based Online Resource for Collaboration in the Field of Product Family Design. Proc. ICED 2009, Stanford, CA (2009)
40. www.research.ibm.com/jam/601/p28-millen.pd, 2006. Accessed January 2011
41. Høimyr N., Jones P.L.: Wikis supporting PLM and Technical Documentation, Proc. PDT Europe Conference 2007, Geneva, Switzerland (2007)
42. Hawryszkiewycz I.T.: Lightweight Technologies for Knowledge Based Collaborative Applications. Proc. IEEE CEC 2006 (2007)
43. http://www.vuuch.com. Accessed January 2011
44. Sanghee K., Bracewell R.H., Wallace, K.M.: Improving design reuse using context. Proc. ICED 2007, Paris, France (2007)
45. Kirsch-Pinheiro M., Villanova-Oliver M., Gensel J., Martin H.: Context-Aware filtering for Collaborative Web Systems: adapting the awareness information to the User's Context. Proc. ACM symposium on Applied computing 2005, Santa Fe, NM (2005)
46. Hull R., Neaves P., Bedford-Roberts, J: Towards Situated Computing, Proc. IEEE ISWC 1997 (1997)
47. Prekop P., Burnett, M.: Activities, context and ubiquitous computing. Computer Communications, Vol. 26(11), pp.1168–1176 (2003)
48. Gustavsen R.M.: Condor – an application framework for mobility-based Context-Aware applications. Proc. UBICOMP 2002, Goteborg, Sweden (2002)
49. Redon R., Larsson A., Leblond R., Longueville B.: VIVACE Context Based Search Platform. Proc. CONTEXT'07, Roskilde, Denmark (2007)
50. Heisig P, Caldwell N.H.M., Grebici K., Clarkson P.J.: Exploring knowledge and information needs in engineering from the past and for the future e results from a survey. Design Studies, doi:10.1016/j.destud.2010.05.001 (2010)
51. Hooey B.L., Foyle D.C.: Requirements for a Design Rationale Capture Tool to Support NASA's Complex Systems. International Workshop on Managing Knowledge for Space Missions. Pasadena, CA (2007)
52. Klein M.: Capturing design rationale in concurrent engineering teams. IEEE Computer Journal, Vol.26(1), pp.39-47 (1993)
53. Bracewell R., Gourtovaia M., Moss M., Knott D., Wallace K., Clarkson P.J.: DRed 2.0: a method and tool for capture and communication of design knowledge deliberated in the creation of technical products. Proc. ICED 2009, Stanford, CA (2009)
56. Gross M.D.: The electronic cocktail napkin – Computer support for working with diagrams. Design Studies, Vol.17(1) (1996)
57. Shipman F.M., McCall R.J.: Integrating different perspectives on design rationale: Supporting the emergence of design rationale from design communication. AIEDAM, Vol.11(2), pp.141-154 (1997)
56. Hu X., Pang J., Pang Y., Atwood M., Sun W., Regli W.C.: A Survey on Design Rationale: Representation, Capture and Retrieval. Proc. ASME DETC/CIE 2000, Baltimore, MD (2000)
57. Conklin E.J., Yakemovic K.C.B.: A process-oriented approach to design rationale. Human-Computer Interaction, Vol.6(3-4), pp. 357–391 (1991)
58. Goodwin R., Chung P.W.H.: An Integrated Framework for Representing Design History. Journal of Applied Intelligence, Vol.7(2), pp.167-181 (1997)
59. Chandrasekaran B., Goel A.K., Iwasaki Y.: Functional representation as design rationale. IEEE Computer Society Press, Vol. 26(1), pp.48–56 (1993)
60. Medeiros A, Schwabe D., Feijó B.: Kuaba Ontology: Design Rationale Representation and Reuse in Model-Based Designs. Lecture Notes in Computer Science, pp. 241-255 (2005)
61. Jin Y., Ishino Y.: DAKA: design activity knowledge acquisition through data-mining. International Journal of Production Research, Vol. 44(14), pp. 2813-2837 (2006)

62. Saridakis K.M., Dentsoras A.J,: Soft computing in engineering design - A review. Advanced Engineering Informatics, Vol. 22(2), pp. 202-221 (2007)
63. http://copeit.cti.gr/site/index.html, Accessed January 2011
64. Ryskamp J.D., Jensen CG., Mix K., Red E.: Leveraging design rationale to improve collaborative co-design CAD environments. Proc. TMCE 2010 Ancona, Italy (2010)
65. Sung R., Ritchiea J.M., Reab H.J. Corneyc J.: Automated design knowledge capture and representation in single-user CAD environments. Journal of Engineering Design (2010)
66. Conway A., William I., Andrew L.: Knowledge Enahnched Notes (KEN), Proc. ICED 2009, Stanford, CA (2009)
67. De Liddo A., Buckingham Shum, S.: Cohere: A Prototype for Contested Collective Intelligence. Workshop on Collective Intelligence in Organizations. Proc. CSCW 2010, Savannah, GE (2010)
68. http://www.wired.com/wired/archive/14.06/crowds_pr.html, 2006. Accessed January 2011.
69. Pitta D.A., Fowler D.: Online consumer communities and their value to new product developers. Journal of Product & Brand Management, Vol. 14(5), pp. 283–291 (2005)
70. Ribiere M.V, Tuggle F.D.: Fostering innovation with KM 2.0. VINE: The journal of information and knowledge management systems, Vol. 40(1), pp. 90-101 (2009)
71. Di Gangi P.M. and Wasko M.: Steal my idea! Organizational adoption of user innovations from a user innovation community: A case study of Dell IdeaStorm. Decision Support Systems, Vol. 48(1), pp. 303-312 (2009)
72. Smith D., Valdes R.: Web 2.0: Get ready for the next old thing. Gartner Research Paper. Stamford, CT (2005)
73. Füller J., Bartl M., Ernst H. Mühlbacher, H.: Community based innovation: How to integrate members of virtual communities into new product development. Electronic Commerce Research, Vol. 6, pp. 57–73 (2006)
74. Tseng M.M., Jiao R.J. Wang C.: Design for mass personalization. CIRP Annals - Manufacturing Technology, Vol. 59, pp. 175–178 (2010)
75. Rittel H.W.J., Webber M.M: Dilemmas in a General Theory of Planning. Policy Sciences, Vol. 4, pp. 155-169 (1973)
76. Cross N.: Designerly ways of knowing. Design Studies, Vol. 3(4), pp. 221-227 (1982)
77. Flanagan T., Eckert C., Clarkson P.J.: Externalizing tacit overview knowledge: A model-based approach to supporting design teams. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 21, pp. 227–242 (2007)
78. Rosenzweig P.: Misunderstanding the nature of company performance: The halo effect and other business delusions. California Management Review, Vol. 49(4), pp. 6–20 (2007)
79. Simon H.A.: Rational Decision Making in Business Organizations. The American Economic Review, Vol. 69(4), pp. 493-513 (1997)
80. Grebici K., Goh Y.M., Zhao S., Blanco E. McMahon C.: Information Maturity Approach for the Handling of Uncertainty within a Collaborative Design Team. Proc. CSCW 2007, Limerick, Ireland (2007)
81. Johansson C.: Knowledge Maturity as Decision Support in Stage-Gate Product Development: A Case From the Aerospace Industry. Doctoral Thesis, Luleå University of Technology, Luleå, Sweden (2009)