This is the published version of a paper presented at 24th ISPE International Conference on Transdisciplinary Engineering, Singapore, 10 July to 14 July, 2017.

Citation for the original published paper:


N.B. When citing this work, cite the original published paper.

Open Access

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:hj:diva-36827
Applying connectivism to engineering knowledge to support the automated business

Joel JOHANSSON a,1 and Fredrik ELGH b

a,b Product Development, School of Engineering, Jönköping University, Sweden

Abstract. Maintaining products in an automated business includes digitalization and automation of engineering knowledge. When new products are to be developed, and introduced not only has the production processes be automated but also the knowledge regarding how the product should be constituted depending on customer requirements. One big challenge that companies of this kind face is how to make sure that the knowledge automated still can be understood by its stakeholders during the development project and after product release and through the whole product life-cycle, which might last for decades.

In this paper, we present a method to navigate and share vast amount of knowledge in businesses with high degree of automated engineering. The method is based on the connectivistic view of knowledge were network formation and filtering are two corner stones which implies the utilization of graph theory together with electronic publishing functionality.

Keywords. Engineering Knowledge, Engineer to order, Knowledge Management, Connectivism

Introduction

Engineering knowledge refers to the knowledge engineers apply when developing products and corresponding production systems. This is a wide definition with emphasis on applying, which means that the knowledge is part of decision making processes and hence excluding curiosities. Engineering knowledge refers to any reason for why, how, when, where, what, by whom something is to be done or be constituted.

Knowledge management and knowledge based engineering have for decades strived to digitalize and automatically utilize this kind of knowledge within product and production development. Knowledge based engineering (KBE) includes tools and methods to digitize and automate engineering knowledge. Even if KBE has gained much attention through the last three decades it is still found hard to industries to develop and even harder to maintain them over time. Here it is suggested to take the connectivistic view on knowledge to see what can be achieved to help industries continuously grow their product knowledge. The paper is organized as follows: first the research method is described, then a frame of reference is given where KBE and connectivism is briefed. Thereafter is the case study described in connection with the

1 Corresponding Author.
connectivistic view on knowledge. Finally, a discussion is given followed by conclusions.

1. Research method

This paper reports a research project entitled *Efficient Implementation and Management of Systems for Design and Manufacture of Custom Engineered Products* (IMPACT) that runs 2015 through 2017 aiming at developing tools and methods to make increasing and more effective use of information and knowledge rich systems supporting customization of engineered products. Four large companies are engaged in the research project and serve as testing areas for the ideas. This paper presents suggested working approach for one of the companies.

2. Frame of reference

This paper deals with a KBE system that has been developed for decades and is very mature within the company. The KBE system is in fact a part of the business case and a key factor to the company to stay competitive. In this section, we will shortly review what KBE is and then review a new perspective on knowledge called connectivism and that has served as guidance through the development of the suggested working approach. The connectivistic perspective is described in the context of knowledge and learning in general.

2.1. Knowledge based engineering

La Rocca [1] defines knowledge based engineering as a technology based on the use of dedicated software tools called KBE systems, which are able to capture and systematically reuse product and process engineering knowledge, with the final goal of reducing time and costs of product development by means of 1) automation of repetitive and non-creative design tasks and 2) support of multidisciplinary design optimization in all the phases of the design process.

![components of knowledge management](image)

*Figure 1: Components of knowledge management. Adapted from [1].*
Knowledge based engineering can be said to be the integration between artificial intelligence and computer aided engineering. Artificial intelligence is a set of methods and models from the computer science research field that support flexible modelling of concepts and methods for logical reasoning while computer aided design includes methods and models to model geometry and product structures. La Rocca [1] further puts knowledge base engineering in the context of knowledge management saying that KBE is a subset of knowledge engineering which is a sub set of knowledge management, see Figure 1.

2.2. Connectivism

Connectivism is a philosophy of knowledge described by Siemens [2] which addresses learning that is located within technology and organizations, a type of learning that KBE ultimately is intended to support. Connectivism is based on nine principles (not numbered in the reference): 1) Learning and knowledge require diversity of opinions to present the whole. 2) Learning is a network formation process of connecting specialized nodes or information sources. 3) Knowledge rests in networks. 3) Knowledge may reside in non-human appliances, and learning is enabled/facilitated by technology. 4) Capacity to know more is more critical than what is currently known. 5) Learning and knowing are constant, ongoing processes (not end states or products). 6) Ability to see connections and recognize patterns and make sense between fields, ideas, and concepts is the core skill for individuals today. 7) Currency (accurate, up-to-date knowledge) is the intent of all connectivist learning activities. 9) Decision-making is learning and the incoming information is seen through the lens of a shifting reality.

Five components are identified within connectivism. Networks is where knowledge resides. Conduit, context and content together shape the meaning of knowledge and individualized filters to help focusing.

Central in the connectivistic view on knowledge is that learning is a network formation process [2]. In the knowledge technologies as seen in Figure 1 this is realized through community of practices, systems for computer-supported collaborative work, ontologies and knowledge webs. Interestingly, these technologies are not considered to be a part of knowledge based engineering system.

Context in the connectivistic view includes elements like emotions, recent experiences, beliefs, and the surrounding environment. Each element possesses attributes, that when considered in a certain light, informs what is possible in the discussion. The object is tied to the nature of the discussion, framework or network of thought. The context-game is the formulation and negotiation of what will be permissible, valued, and the standards to which we will appeal in situations of dispute. The context-game of implementing a new corporate strategy involves individuals, politics, permissible ways of seeing and perceiving, recent events, corporate history, and a multitude of other factors [2]. Context in this broad definition is not normally considered in theories for knowledge management, knowledge engineering and KBE.

Conduits is the medium through which knower (i.e. experts) and seeker (i.e. knowledge consumers) communicate and through which the known entity finds expression [2]. Conduit is the facilities making the knowledge relevant, current, and available.

Siemens [2] briefly reviews the history of how information has been consumed and concludes that we used to go to one source of information to get a thousand points of information (for instance new papers). Now, we go to a thousand sources of
information to create our own view. He continues by saying that we have become the filter, mediator, and the weaver of the networks. A statement that indicates how intervened the concepts in connectivism are.

Since we as humans have a limited possibility to focus our attention (we can only do one or a few things at a time and we only have a limited time per day) and since the amount of information and knowledge is ever increasing there is a great need for filter the content based on individualized filters and current context.

Content is of course of central importance (even if it is told that the capacity of learning is more important than what we already know). Relevance, however, is not only about the nature of content. The process of ensuring currency of content/information is critical to manage knowledge growth and function effectively. Content has to blend together with conduit and context [2].

3. Case study

The studied company is the world’s leading supplier of tools, tooling solutions and knowhow to the metalworking industry. The company is active in an internationally very competitive market and needs to constantly cut development lead time by seeking means to improve their processes and system maintenance. The company has a long-standing tradition in automation of quotation and order processes and has adopted an engineer-to-order business model supported by systems for automated design and production preparation of customized product. A request for quotation of a custom engineered product is replied within hours including detailed design drawings and a final price. All the necessary documents and manufacturing programs are automatically generated when the bid is accepted by the customer. On order acceptance, production operations including CNC and CMM routines are sent to the production site. That in turn initiates the automatic trucks or other material handling systems to feed the manufacturing machinery with working material. When the individual product is finally ready it is automatically packed, and shipped to the customer, sometimes just a few hours after order acceptance.

During the first step in the research method two research areas were found important to the company [3]: 1) Models which enables companies to formalize their knowledge to facilitate multidomain utilization. 2) Documentation of relations between produced products specific system versions, used in the products creation, to connect it to the knowledge of which it was derived from. The two areas are targeted with the connectivistic approach.

The company has been working with knowledge based engineering, or rule based design since the mid 1980’s and is a very mature company in this area. The fourth generation of their design automation system is based on a domain specific language (DSL) which means they developed a unique programming language to be able to capture all aspects or their products in a way that suits the engineers. The DSL is a text-based programming language that comes with an in-house developed integrated development environment (IDE). The DSL captures not only the concepts and logics of the products but also the geometrical models which makes the DSL a KBE-system as seen in the frame of reference, Figure 1.

The knowledge of the product is captured and digitized as class hierarchies and logical expressions in a format and with terms that are crafted to fit the everyday terminology of engineers at the company. The IDE is crafted so that product
knowledge is stored close to the product geometry, albeit not within the CAD-system. User defined features (UDFs) have served as the baseline to achieve this. UDFs are composite features, i.e. a combination of geometrical elements packed with parameters and references so that they can be used in many parts. The design engineers develop the UDFs necessary to make complete products and the design programmers uses the IDE to write programs in the DSL so that the UDFs are automatically combined into products based on user enquires.

As seen in Figure 1 knowledge management and knowledge engineering includes networking activities, which is in line with the connectivistic principle that learning is a network formation activity. Since KBE systems are results of these activities inherently they will form networks too. To visualize these networks, it is possible to apply graph theory. The visualization and reflection upon such graphs guides the continual refinement and knowledge development as marked in the bottom left arrow in Figure 1.

To try this out a plugin was made to the IDE that scans the knowledge base to connect its various elements and how they are related through logical and structural expressions in the code.

The plugin was applied to a sub-set of the knowledge base for the simplest product at the company with the result as shown in Figure 2. The nodes of the graph represent both geometrical and conceptual components such as parts, UDFs, surfaces, coordinate systems, class declarations, enumerations, and base type parameters. The graph contains both “kind-of” and “part-of” connections.

![Figure 2: 1366 nodes connected through 1458 edges](image)

There are many ways of automatically layout graphs. The following methods were tested: Fruchterman-Reingold [4], Force Atlas 2 [5] and Yifan Hu [6]. The graph in Figure 2 is a result of combining the two latter layout methods.

The graph in Figure 2 is hard to grasp, it simply contains too much information to consume at once. Therefore, filters are so important. Filters should make it possible for
the engineers, individually and based on their context, to control: What to see (or not to see). When to see. How much to see. In what format to see. It should also be possible for engineers to manipulate and statically store views of the product knowledge.

Another add-in was developed for the IDE that continuously scan the neighbouring context of the code block currently selected to create, layout and broadcast smaller graphs. This gives an instant and focused view of context. The graph creation as aligned with the IDE abstract structure tree so that whenever the programmer changes the code, the graphs are automatically updated within milliseconds.

It is possible to change the levels of neighbouring code block to view at a time, and it is possible to change whether to see kind-of or part-of relations or all together. It is possible to change at what access level to generate the graph (private or public members). It is also possible to tell what types of content to see, parts, UDFs, geometrical elements, classes or base type parameters (if all is shown at once there would be 4620 nodes and 4846 connections in the graph corresponding to Figure 2).

When making this close integration of graph visualization and IDE it was also possible to capture content in forms of code-comments into the graphs. Since the IDE supports the commenting of code blocks in the web-page format (xhtml-fragments that supports rich text and pictures) it was found possible to add the so important content into the programming code in a format that fits the content itself: text and pictures (movies would be possible as well but was not relevant). When hovering the mouse over or right clicking on a node a pop-up window show the comment in web-page format.

According to the connectivism, conduits are the pipes where knowledge continuously flow. It is said that conduits should be blended with content and context. This has been achieved by the integration of graph-visualization and web-page-comments of the code, which forms the conduits to the engineers. Through the connections made by the logics of product knowledge it is possible to navigate the knowledge and through the comments it is possible to change it (code blocks also need to be changed in such case, not just the comment). The dynamically generated graphs were made interactive so that when clicking on a node the IDE focused on that code block.

Still there was a need for another type of conduit. That was to provide knowledge to stakeholders that make use of the KBE system after product release. These stakeholders include aftermarket and engineers making special adjustments or modifications of the products. They need to know why the products are constituted the way they are, still they do not need all the information. The information provided to them also should be of the same version as when the product specification (CAD-models and other documents) where generated through the KBE system.

To achieve this, methods were developed to extract the web-page-comments from the code for each executed code block when running the KBE system. These web-page-comments are then compiled into an electronic book (ePub or PDF) containing all the information for that specific instance of the product family, in an interlinked way based on the network. So, that for instance when viewing a page describing a class definition it is possible to click on links to pages describing its members and other classes where it is used.
4. Discussion

To apply the connectivistic view to knowledge based engineering five areas were covered: network, filters, context, content and conduits. Even if connectivism is an abstract philosophy of knowledge this research proves that it is possible to adapt such a mindset to further enhance KBE systems and processes to keep them alive and up-to-date. The methods developed during the research project are far from ready and much work must be done to make them readily available to the engineers in the global company.

When reviewing the nine principles of connectivism it can be concluded that they are, or can be realized at the company by the following (same numbering as in section 2.2): 1) Since the IDE support web-page like comments it is possible to let other than programmers make use of it to encourage an interdisciplinary work setting. Design engineers, production engineers, marketing and programmers would be able to continuously contribute to the knowledge flow. User access functionalities should be added to the IDE for this to be secure though. 2) The KBE system serves as the infrastructure to connect nodes of information sources, to form the network. 3) The digitized knowledge stored in the KBE system inherits its network from processes in knowledge management and knowledge engineering. The visualization of the network makes this more obvious. 4) The KBE system facilitates storage of digitized engineering knowledge. 5) Through the visualization of networks, interactive retrieval and navigation amongst pieces of knowledge and the automated compilation of electronic books the capacity of knowing more has increased. 6) Viewing knowledge development as a continuously ongoing process can be enabled through extensive versioning control of the knowledge. It would change the picture in Figure 1 so that there would be no barriers between knowledge management, knowledge engineering and knowledge based systems, they would be continuously sub processes of the organizational learning. 7) The visualisation of network and context through graphs makes it literally possible to see the connections and patterns of the knowledge. 8) The close integration of geometrical models, concepts, logics and web-page-like comments within a single IDE makes the knowledge content up-to-date. To keep the accuracy a continuous review of the knowledge is important. 9) Viewing decision-making as a learning process is in line with viewing product development as learning, in which engineers indeed find themselves in a very shifting reality. Through the IDE with the visualisation of network and context, interactive navigation of knowledge with easy to add knowledge content an agile KBE development platform is at place supporting the flow of information.

One thing that has emerged from this research is the absence of content, within code comments, but also outside of the IDE. When a product development project is ready and product is released in form of a KBE-system, it lacks comments and the background knowledge used to develop the product still resides in the minds of the engineers. Further work will be put to develop methods to make it more natural to programmers, but more important possible to non-programmers, to add comments into the IDE already at the beginning of product development projects, even before adding code to the KBE. This means that comments will be added first and code thereafter.

The low amount of knowledge re-use at the company is thought to be a result caused by low standardization in the formalization of knowledge created by the design engineers. Report content varies a lot from engineer to engineer and it is not certain that the Company can make use of them in new projects. A more standardized way to
formalize this knowledge could result in a higher re-use of the knowledge which could both save time and ensure quality of the produced products [3].

The broad definition of context within the connectivism was covered at the company through the development of a DSL, the common language suited to the engineers. During that process, it was agreed on basic terminology making up the DSL. Even if released the DSL is general enough to engineers to develop and share their own concepts and models which reflects their own context. When other engineers change these models, it will in turn be from their context point of view. It can further be said that the KBE treats the elements of knowledge stored in the knowledge based on context as well. Sometimes a UDF is treated as a logical element, another time it is viewed as a geometrical element being part of a drawing, and yet another time it is used in a CNC or CMM process. The term polymorphism in computer programming reflects this very technical view of context, even if the connectivistic term is much broader.

5. Conclusion

This paper is a starting point of applying the connectivistic view of knowledge [2] to knowledge based engineering. It was shown that by scanning the elements within a knowledge base in a KBE system it is possible to visualize and navigate its content through graphs. It was also shown that it is necessary to enable individualized and contextualized filtering of the vast amount of information. Not least has the importance of adding content been shown. No matter how good the conduits are and no matter how many nodes and connections the network has, it still useless if there is no content. With the increased ability for the company to learn this challenge can now be well met.

Acknowledgement

The authors express their gratitude towards the participating company in the study as well as The Knowledge Foundation who partly funds the project.

References