Automated Design Rationale Capture within the CAx Environment

Kenneth J. Mix[1], C. Greg Jensen[2] and Jordan Ryskamp[3]

[1]Brigham Young University, kmix@byu.edu
[2]Brigham Young University, cjensen@byu.edu
[3]Brigham Young University, jryskamp@gmail.com

ABSTRACT

The reuse of the design rationale behind company products has become an increasingly important practice for improving designs.This paper proposes a methodology for the automated capture of design rationale.Current industry design rationale capture methods focus on file storage and management through version control while the proposed method provides the additional ability to capture the entire design process rather than only versions of files.The method requires three primary elements: a CAx tool, a communication package, and a storage medium.An implementation that utilizes Siemens NX 6 CAD software, Skype communications software, and Microsoft SQL server is also presented.The additional functionality provided by this implementation is discussed, and methodology for testing and comparing this implementation to other design rationale capture systems is also presented.

Keywords:knowledge capture, design rationale, knowledge reuse.
DOI:10.3722/cadaps.2010.361-375

1 INTRODUCTION

Today's product development firms face greater competition than ever before.Worldwide, companies are attempting to shrink design cycle times, release new products faster, all while improving quality and increasing supply.This paper presents a methodthat attempts to address these needs by utilizing existing software to effectively store and retrieve company knowledge in a centralized storage system that allows for the fluid storage and retrieval of data. This storage system is termed the "knowledge cloud." The benefits and value provided by this method allow companies to stay ahead in this competitive market.

To stay ahead of the curve, companies must re-use their company resources.Gone are the days of one-off designs.Today's product development firms build from previous designs to iterate and create new products. This reuse of resources prevents companies from having to "reinvent the wheel."Specifically, companies utilize product lifecycle (PLM) or product data management (PDM) packages to store and reuse previous CAD and other product definition data, requirements lists, bill of materials lists, and other product lifecycle data.This data, paired with parametric design techniques allow companies to develop new products much more quickly.

Perhaps the most important resource companies have is their employees.The ideas and intents of an employee during the product development process, along with the decisions they make can be termed “design rationale.”Seasoned engineers and designers know the ins and outs of a particular product line, and can assist newer employees.But what happens when these seasoned engineers leave the company?Companies may attempt to capture their knowledge as they leave in a final documentation package, but for the most part, the rationale behind designs is lost.

This paper proposes a tool methodology for capturing design rationale from these employees not at the end of their employment, but every day as the employee works. The purpose of capturing more information from the process is to ensure that every decision and intent on the project is captured, rather than having a limited set of knowledge.The storage of this design rationale can then beused in the future to determine how products were developed.

2 BACKGROUND

This research builds upon the work of other research that has focused on the capture and re-use of design rationale.A brief literature review is presented in order to lay a foundation for the principles used in this method and to show how this work builds upon other research.

2.1 Design Rationale

The combination of specifications, motivations, and actions for the purpose of creating designs is called design rationale [13].The primary motivation for much of this work is the inability of data management packages to collect and store the intents and motivations that occur throughout a project [10].

Most of the research in this area has been focused on providing frameworks and tools for representing or entering design rationale data [19].Some examples of tools for accomplishing this are gIBIS[3], PHIDIAS [18] and Compendium [4]. The capture of DR is subject to many barriers [7,9]. Some of the limitations and problems with DR capture are discussed by Dutoit, et al. [6].

While the majority of these tools capture data, they do not directly integrate themselves with the engineer’s daily routine.Because of this, their adoption into the industrial realm has largely failed[1].This is likely because design rationale capture is often viewed as being disruptive and time-consuming [2]. One solution to this issue is to avoid as much as possible explicit interaction with the designer, but rather to passively record their actions with the CAx tool.This method was first shown by the Rationale Construction Framework (RCF) designed at SRI International [14].This research builds on the SRI approach.

Additionally, most of the tools and methods researched do provide at least some ability to reuse the data.Lee has summarized the capability of many design rationale systems to reuse data [11].

This tool builds directly upon the work of these researchers.It improves on the failure of other tools to integrate with an engineer’s daily routine. The method requires tight integration between the DR capture and retrieval tool andengineering tools that user’s work in daily. As stated above, this more direct integration encourages the use of the system and will assist with adoption, both on the corporate and individual level. This paper builds directly on previous work performed by the authors[12,17].

2.2 Foundational Tools

In order to fully understand the implementation of this method, a basic understanding of some foundational tools is necessary.A basic overview of some of the more essential tools used is presented in order to supplement the readers current knowledge of these tools.

2.2.1 Programming

A variety of programming methods were used to create the tools.The primary programming language was Microsoft’s C# language.C# was chosen because it is a simple, modern, object-oriented approach to creating robust applications [8].Additionally, this work extensively uses Application Programming

Interfaces (API) in order to integrate with the user's daily workflow.APIs allow a programmer to access functionality from 3rd party programs.

2.2.2 Database Management Tools

The tool presented in this paperalso uses a relational database management system, or RDBMS as the storage system.This system was pioneered by Dr. E.F. Codd[15].The power of the RDBMS lies in its simplicity and reliability [16].SQL is a query language that allows interaction with RDBMS systems [5].

3 METHODS

This section presents the general method for the improvement of future product design through the automated capture of design rationale. The reader should note that the method described is intended to be universally applicable to any set of engineering, communication and data storage tools. The general method can be applied to many different types of implementations, since it is intended to be a general statement of steps that should be taken to capture and reuse design rationale.
The steps of the method will be explained verbally and by using set builder notation.A series of examples will also be used to illustrate the method, however for the full details of the implementation please refer to section four.

3.1 Overall Method Summary

The method proposed here for improving future design by automating the capture of design rationale and providing for retrieval consists of two primary operations, each consisting of multiple steps. The operations with their steps are shown in Figure 1.

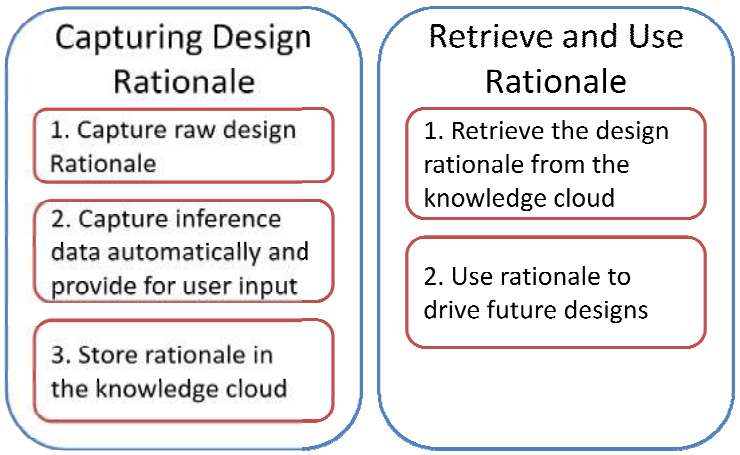

Fig. 1: Method showing main operations with each sub-step.

While simplistic in its structure, it will be shown that this method provides for more steps of the design process to be captured and stored as compared to what is currently possible with existing tools.

Additionally, there are performance gains over current tools due to the fact that each of these steps is readily automated in a computer environment.By leveraging the power of computers to gather and store information, this method provides new capabilities and the ability to understand product development processes better.

3.2 Capturing the Design Rationale

The first operation of the method concerns itself with capturing the design rationale.A major key to the effective capture of design rationale is to minimize the amount of interaction the engineer must

have with the system, while maximizing the amount of data captured.Efforts were made to ensure that the process is automated wherever possible in order to remove unnecessary interaction.However, some minimal human interaction is unavoidable, as user verification is essential to ensure integrity.

Additionally, it is essential that the capture step be tightly integrated with the daily routine. The user should not be required to use another tool to capture design rationale. All capture should take place from within the engineering tool that is used by the individual.

3.2.1 Capture Raw Design Rationale

The first step of capturing design rationale is to capture the raw data.A is the set of all actions taken by the user during a design session.

$$\{A: A_1, A_2, A_3, A_4,\ldots,A_n\} \tag{3.1}$$

An example to illustrate this step of the method is to consider an engineer who is working on a design for a new gear.A gear is a relatively simple part that could easily be modeled in a single day by an experienced engineer.Figure 2 shows the steps that are captured by traditional PLM tools, as well as those captured by the proposed method.

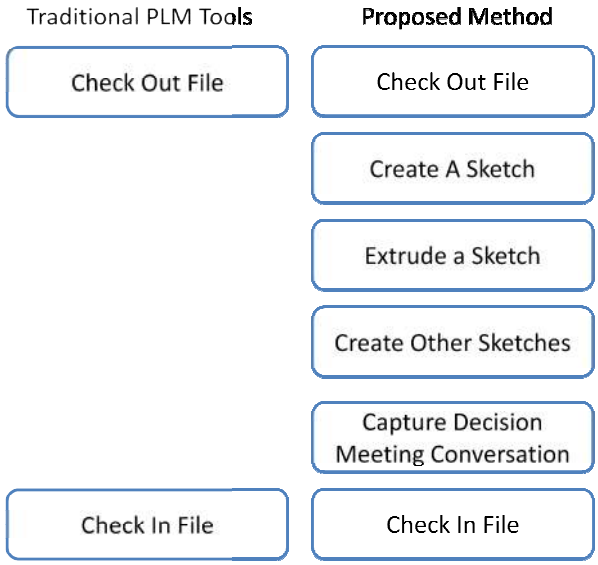

Fig. 2: Data captured from PLM vs. proposed method.

Notice that in traditional version-control packages, the only data that would be captured would be the file state before and after it has been checked in.Alternatively, the proposed method would capture each stage of the process, which will ultimately lead to a greater understanding of the process by which the part was created.

3.2.2 Capture Inference Data and Provide for User Input

Of course, the raw design rationale that is captured would be of little use if not for the coincidental capture of the user's intentions with each action.These intentions are termed inferences, or what the user infers as they take an action.The capture of inference data gives meaning to the raw data.It allows the data to be stored effectively and provides additional information about the data.Inference data can be shown using set builder notation:

$$\{N: I_1, I_2, I_3, I_4,\ldots,I_n\} \tag{3.2}$$

N is the set of all user intentions and other meta-data during the process. The combination of the raw data with the user's inferences is what we term design rationale and can be written as:

$$\{DR: N \cup A\} \tag{3.3}$$

As an example, again consider the engineer who begins modeling a gear.His actions are listed in the left column of Fig. 3.

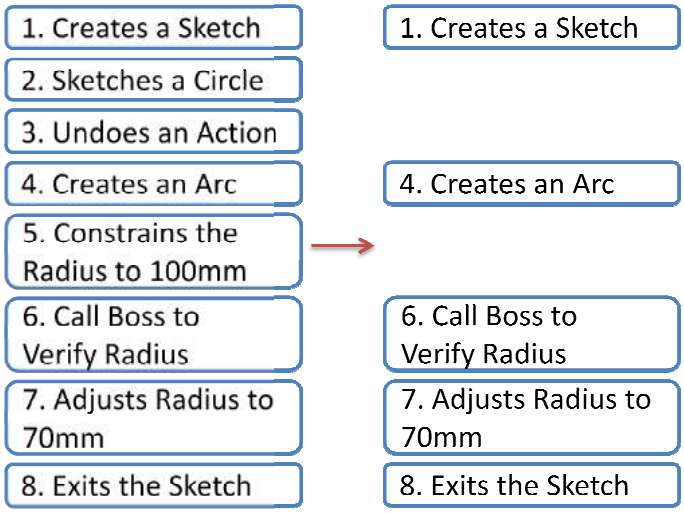

Fig. 3: Raw actions taken by the user versus actions captured by the method.

Steps 2, 3 and 5 should not be captured, since they are intermediate steps made by the engineer, which are corrected in later steps.The method denotes that these types of actions need not be recorded.This has two advantages.First, it does not clog valuable computing resources with unnecessary data, and second, it allows a person looking back on this data in the future to quickly assess the data rather than having to sift through unnecessary and confusing steps.

In addition to inferences, meta-data that relates objects of data is also captured.An example of this would be time stamps, taken at key moments in the CAD package and correlated to time stamps in a conversation.In this way not only can a future user refer to the CAD design process, they can review the communication that went along with that design. Because a primary goal of the method is to minimize user interaction, the implementer should note that both the capture of raw data and meta-data can be readily automated.

3.2.3 Store Rationale in the Knowledge Cloud

The last step of capturing this data is to record it in the knowledge cloud.The storage method and medium is not specific, since there are many equally effective methods for storing and relating data. The only requirements for the storage system are that it has the ability to be automated.This is essential, since the method draws heavily on the ability of computers to process data automatically.Data storage and correlation of this magnitude would simply not be reasonable in any manual method.

3.3 Retrieve and Use Design Rationale

The second operation of the method is to retrieve and use the stored design rationale.Again, minimizing the amount of time the user spends using the system is advantageous.This attempts to give the user the ability to search and parse the data effectively so that it can be easily reused.

3.3.1 Retrieve the Design Rationale from Knowledge Cloud

The goal of the knowledge cloud is to allow for the capture and storage of more steps of the design process than is currently possible with existing tools. Rather than being used as only a legacy data storage medium, the knowledge cloud also includes explanatory data, which canhelp a user understand what was done on a project and why.

It is required that a subset of data be returned to the user from the larger set of data.This action can be written as follows:

$$\{S \subseteq DR \mid S = R\} \tag{3.4}$$

Where DR is the set of all design rationale, and S is a subset of data that matches the search parameters R.The defined algorithm for retrieving a subset of data from the database will of course vary depending on the implementation. Some implementations may choose to use a simple text-based search system, while others may elect to create a more robust search system which utilizes more advanced search technologies.

This can be illustrated by continuing the previous example involving the Acme Gear Company.Suppose that an engineer inside Acme wishes to review tooth designs from some recent projects, since he is having issues selecting the appropriate size and mesh angle.It is necessary to create an algorithm that enables this engineer to find the subset of data that will help him determine the correct actions to take.This scenario can be represented by Fig. 4.

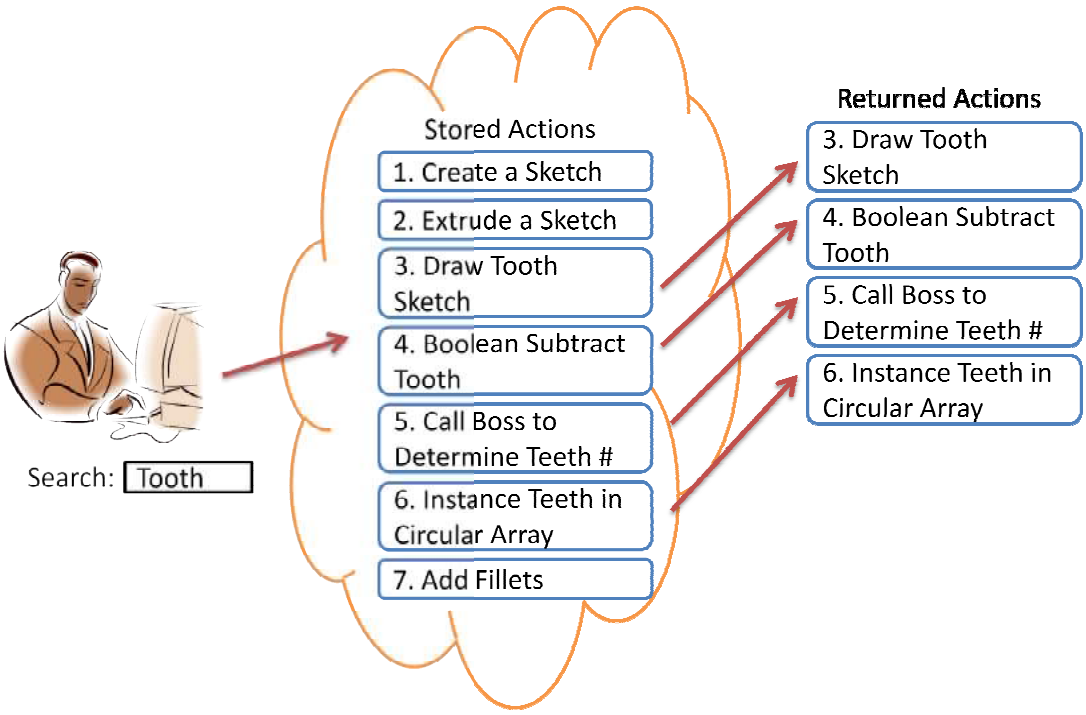

Fig. 4: User is retrieving a subset of data from a larger set.

Notice that only a small subset of actions is returned to the user, specifically those that involve tooth design. Note that with current version control systems, the only method by which the user could get this information would be by reviewing the actual CAD data. This would tell them little about the process and the reasons behind the decisions.

3.3.2 Using Design Rationale to Improve Future Designs

The data can now be used to improve future designs. Because of the additional information recorded, the knowledge cloud can help to explain the logic behind designs, and can be effectively used to create future iterations of the product.

The methods for reusing this data are as numerous as the implementations, and will depend heavily on the specific needs of the company and the individual.Regardless of implementation, the method requires that the design rationale that is captured have to ability to be reused.This will be explained in the continuation of the previous example below.

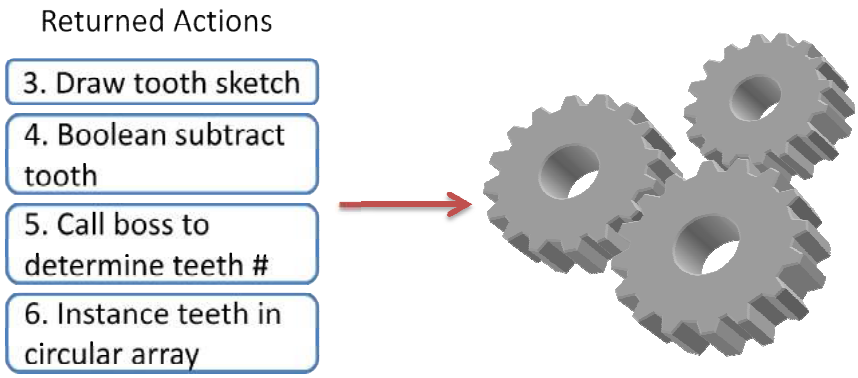

Fig. 5: User can now use the actions that were returned to drive the creation of a new gear.

Using the subset of the data, the engineer can now review the rationale behind how previous gear teeth were designed.He then quickly makes the teeth for his part and continues his work.He has not lost valuable time tracking down the information from other engineers, or bothering his superior.

While this example may appear too simplistic, it nonetheless illustrates how the presented method for storage and retrieval of design rationale in the knowledge improves on the traditional version control-basedenvironment.The relatively little time it takes to retrieve the rationale behind designs allows future designers to quickly build upon the shoulders of their predecessors and produce designs more quickly and with superior design.

4 IMPLEMENTATION

This section discusses how the methodology described was applied to create a tool that would be capable of capturing both CAD and communication data from the design process. This tool was designed with the intent of assessing the method, and so thefunctionality of the tool is limited to that purpose.

The intention of this paper is to present a method for capturing and reusing design rationale. While the researchers have selected Siemens NX 6, Skype, and Microsoft SQL server as the primary implementation software set, this is only one possible implementation. The method could be equally applied to any set of tools that includes a CAx tool, a communications package, and a storage system.

4.1 Architecture Overview

The implementation uses Siemens' NX 6 CAD tool as a base.It also uses Microsoft SQL server heavily which serves as the knowledge cloud storage medium.There are two additional sub-systems, a

communication sub-system, and a CAD sub-system, that have been added in the form of custom NX plug-ins.Each of the sub-systems consists of multiple modules.The sub-systems use a database system to store and manage the data.An illustration of this structure is shown in Fig. 6.

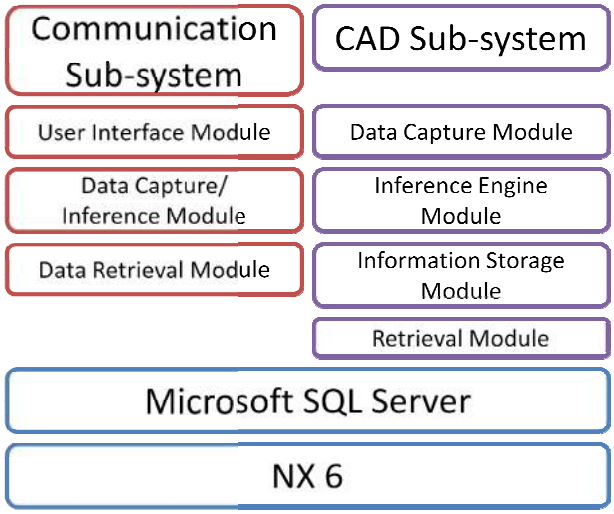

Fig. 6: Implementation architecture.

Both sub-systems of the tool will be discussed in greater detail below.

4.2 Communication Sub-system

The communication sub-system consists of three modules:

- User Interface Module
- Data Capture/Inference Module
- Data Retrieval Module

4.2.1 The User Interface Module

The user interface module (UIM) is a custom built application that resides inside the NX window.This module can be called directly from the NX 6 interface, through the use of an integrated button, as shown below:

Fig. 7: NX 6 Menu Bar showing integrated Skype button (highlighted in red).

The UIM heavily uses the Skype API to drive content and supply reliable voice and text-based communication.As shown in Fig. 8, the first window presented by the UIM presents the user with a “buddy list” that allows them to see presence information for individuals in their organization.

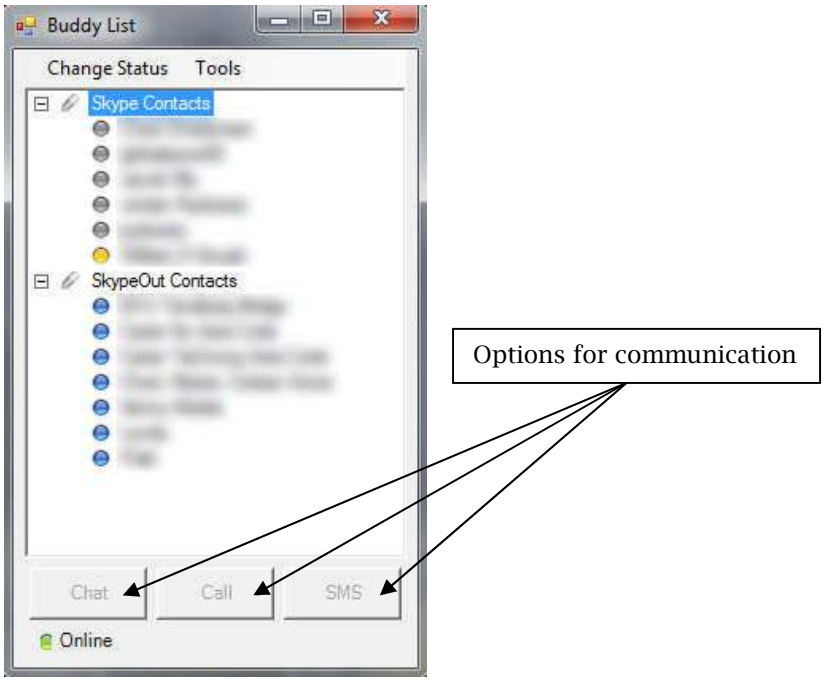

Fig. 8: Buddy List containing the usernames of people that can be communicated with.Usernames have been blurred out for privacy reasons.

When a name is clicked, options are given based on the type of user that was selected.For instance if that user has a cell phone on file in the system, options to either call or SMS text message that person are available, and if they have a computer account on the system they can also be instant messaged.Offline messaging and voicemail are also available.All communication windows reside in the NX interface, but are transparent as to allow design work to continue as the conversations are taking place.

Upon completing a conversation and closing the corresponding window, the user is asked if they would like to save the conversation.The nature of communication in the workplace necessitates the ability to choose whether conversations should be saved.Sensitive content, ITAR or IP controlled conversations, and other reasons may cause the user to neglect saving the conversation.

4.2.2 Data Capture/Inference Module

If the user selects to save the conversation, the raw conversation data is stored.This can include the audio from a phone call, or the text from an instant messaging or SMS conversation. The raw data is useful, since reviewing the communication that takes place during a project can be a useful way to answer questions about what was done. However, raw data independently can be difficult to search, especially in the case of audio files. Because of this, meta-data about the environment is automatically stored, including a time-stamp, the conversation type, the usernames of the people who were involved in the conversation, and the file paths of currently open NX parts.This last piece of meta-data allows for the linking of communication data with files for future retrieval.

In addition to the automated capture of this data, the user is given the opportunity to add a title for the conversation as well as some additional tags, which are keywords that describe the data.This allows the conversation to be further categorized for future retrieval.Giving the user the option to specify this information adds a human element to the data that can be invaluable for reuse on future designs.However, note that excessive interaction with the system reduces use of the system, this implementation attempts to minimize this interaction as much as possible.

The raw data along with the meta-data is then stored in an enterprise class RDBMS, Microsoft SQL Server, which serves as the knowledge cloud storage medium.A focus of this methodology is to encourage data reuse on future designs.The power of Microsoft SQL Server is the ability to quickly and efficiently store and relate data which will ultimately lead to faster retrieval.

4.2.3 Data Retrieval Module

The final module of the communication sub-system is the retrieval module.This implementation focused on the ability of the user to quickly retrieve conversations from the system.A simple search system was developed and is shown below, with results shown.

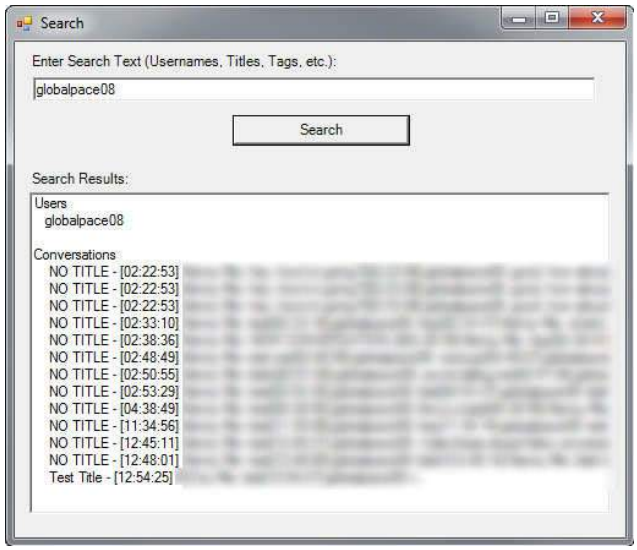

Fig. 9: A list of results from a search for the user “globalpace08”.Conversations have been blurred out for privacy reasons.

There is a single field where keywords, dates, person’s names, etc. can be entered into the system.The database then returns conversations based on those keywords.

While this search system is simple, it shows that the rapid retrieval of data allows a user to quickly gather the information they need from communication based data.Future implementations will include the ability to view data using a number of organization formats, such as chronologically, by project, etc.

In summary, the communication sub-system captures and stores many types of communication data directly from Siemens’ NX 6 environment.This tight integration allows the user to communicate and capture this data without leaving the familiar design environment.

4.3 CAD Sub-system

The CAD sub-system consists of four modules:

- Data Capture Module
- Inference Engine Module
- Information Storage Module
- Retrieval Module

4.3.1 Data Capture Module

The first step of the method is to record data from the design process.This step is implemented in CAD with the CAD Data Capture Module (CDCM).The CDCM unobtrusively gathers information about the user’s design session.This information includes feature information, information about the user,

mouse location and actions, screenshots, etc.In order to accomplish this, the NX API was used to gather information directly from the NX session, such as the feature information.Additionally, C# and .NET framework methods were also used to gather data such as screenshots.

One of the main functions for recording user actions is a programming method termed getCurScreenShot().This method takes screenshots from the CAD system when key actions are performed.These screenshots can then be correlated with data that comes from the NX API about the part and its features.

The number of screenshots that are taken is determined by a variable timer.This timer is used to reduce the load on system resources which can ultimately slow down the system.Another method for limiting this load is to selectively capture information.This is done using the inference engine module, which will be discussed next.

4.3.2 Inference Engine Module

Once the data has been captured, the Inference Engine Module (IEM) can then draw inferences from the data.The IEM has two primary functions, first to draw inferences for future use, and to limit the amount of information that is recorded by the CDCM.

The first primary function is to draw inferences from a set of data.To illustrate this, an example is included.IEM utilizes an algorithm which is encapsulated in the getVisActionsInfers() that parses a set of data, such as a collection of CAD actions in order to find a predefined set ofactions, such as any time a user clicks on a menu, for instance..When these actions are found, they are returned by the function as a list.Many similar inference algorithms were used in this implementation.

The second primary function is to limit the amount of information recorded.As noted above, the gathering of vast amounts of data can be taxing on system resources.This implementation utilizes IEM to capture only the minimum amount of information necessary to describe an action.To explain using an example, screenshots are taken only at key moments in the design process.The moving of a mouse cursor and other trivial actions are not recorded.This allows the system to run much more efficiently, even in environments with limited resources.

4.3.3 Information Storage Module

The information storage module is primarily responsible for storing the data along with any related inference data.A major requirement of the ISM is to be able to quickly and safely store data.

Again, storage was implemented using Microsoft SQL server.Tables were created in the database to store actions, inference information, session data, and user information.These tables can then be accessed using SQL query language.

4.3.4 Data Retrieval Module

In order to retrieve the data as outlined in the method, a data retrieval module (DRM) was created.The DRM does this by allowing a user to search out a particular part, and provides screenshots taken at key moments in the production of that part.The user can review the process by which a part was created by viewing a video feedback of these screenshots. This enables the user to see what actions were performed, and why these actions were performed.

The ability to review the actions by which a part was created adds significant value over current systems, which typically only give the user versions of a part at distinct moments in time. It will be shown in section 5 that when a user can see each step of the design of a part, it allows them to understand the process of designing that part much more clearly.

5 RESULTS

The implementation of this method was created in order to show how this tool can augment current knowledge capture toolsets.A brief discussion of current knowledge capture toolsets is provided in order to establish what additional features the proposed knowledge cloud provides.Following this the additional functionality will be discussed.

5.1 Current Industry Standard DR Capture Methods

Most modern PLM packages include knowledge capture functionality.These packages integrate closely with CAD packages in order to manage product definition data.CAD, Analysis, and other product data can be easily version controlled in the PLM system.

The knowledge capture tools provided with these PLM tools do not fully capture the design process however.PLM systems have evolved from product data management (PDM) systems.PDM systems have traditionally focused on storing product data and controlling the data through revision tracking.PLM systems add some additional functionality, but their knowledge capture functionality is primarily raw data storage and management, with the basic ability to store meta-data through the use of attributes attached to the data.

5.2 Additional Functionality

Current PLM systems are not designed to capture detailed data and designer intent from a CAD session, nor are they designed with integrated communication capture.This tool augments the current PLM toolset by adding these functionalities.

5.2.1 CAD Session Functionality

This method adds the ability to capture data from more steps of the design process.Rather than collecting only the file versions as evidence of the process, it captures the actions that took place in order to create the part.Fig.2 is repeated in Fig. 10 to illustrate this point:

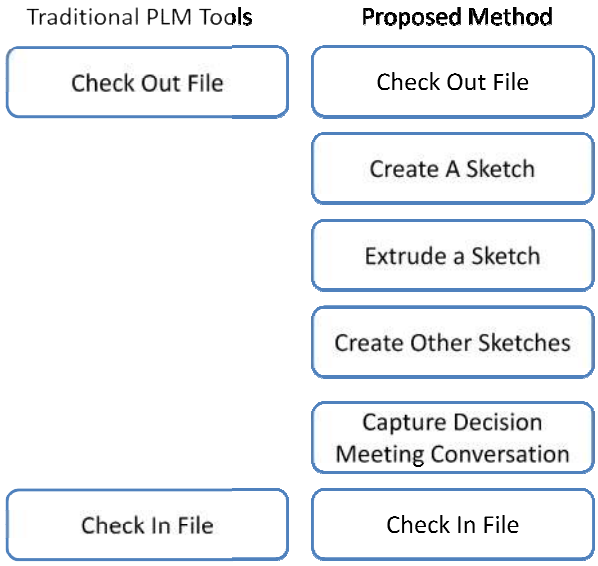

Fig. 10: Data captured by PLM tools vs. proposed method implementation.

Notice the additional steps that were captured in this method as compared to traditional PLM tools.Each step of the design process was captured, rather than just the versions of the file.This way the entire process by which the part was created was captured.

5.2.2 Communication Functionality

In addition to CAD functionality, integrated communication functionality is also added.This integrated system allows the user to communicate with their coworkers without leaving their daily workflow environment.

The communication system is also capable of capturing any conversation that takes place within the tool and correlate this data with the CAD data to more effectively capture the entire design process.

5.3 Experimentation Results

An experiment was developed and performed in order to determine whether this approach to capturing design rationale did indeed improve an engineer's ability to recognize and understand the process of creating a particular part.

The engineer is presented with a hypothetical scenario in which they work for a company that develops brackets. The company has offices in both India and in the United States of America. The engineer participant, located in the U.S.A. is to review the process by which the engineer in India created a part, and then to complete the part. The engineer is shown figure 11.a as guidance, along with drawings of the bolt pattern.

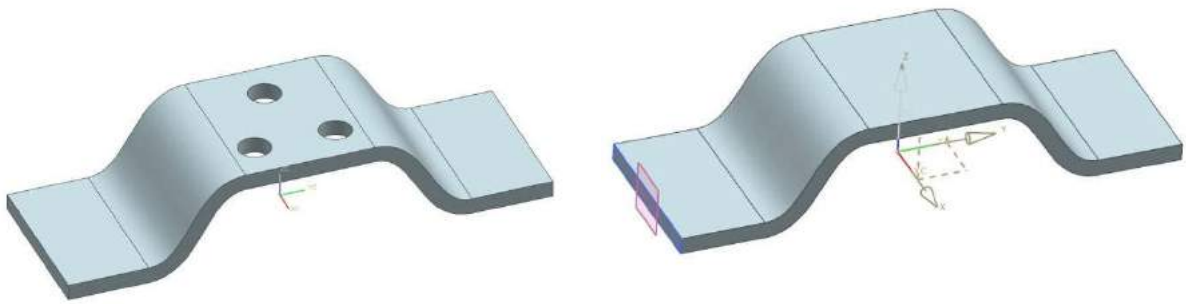

Fig. 11: Bracket image shown to each experiment participant. Brackets: (a) A graphical depiction of the completed bracket, (b) The bracket part as it looked upon opening it.

Upon opening the bracket CAD file that was supposedly created by the engineer in India, they find that the part is almost completed, as shown in figure 11.b. The only missing feature is the bolt patternfeature, which still needs to be added. The key to finishing the model is recognizing that bolt pattern exists, but is hidden by the solid model. The sketch can easily be seen in the wireframe depiction of the part below:

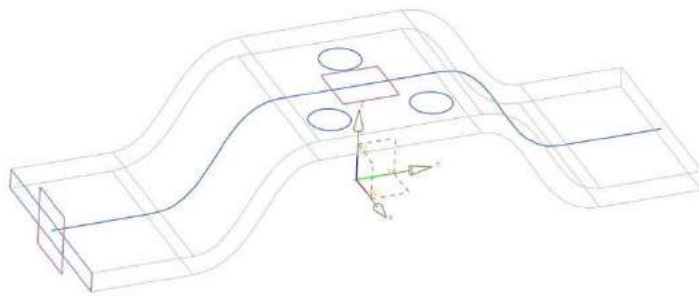

Fig. 12: Wireframe depiction of bracket, showing the hidden sketch.

Because the flange was created by sweeping the rectangle on the left of the diagram, and the guide curve is defined on the same plane as the sketch, the bolt pattern sketch is obstructed from view. The point of the experiment was to determine whether the tool presented in this paper allowed the engineer to see the actions that took place, and then successfully recreate the part.

For the experiment, the sample population was divided into two groups, A and B. Each group was given the same scenario with all of the same information. The single difference was that Group A was also allowed to use the tool presented in this paper to view a brief dynamically generated video of the key events from engineer in India's design session. They were asked to identify the next step of the design and complete it. They were observed to determine whether they were able to identify the existence of the hidden sketch.

The experiment was replicated with 10 different engineering students with a variety of skills and experience. Each student was randomly assigned to either Group A or Group B, leaving 5 in each group. Of the five students in Group B who did not use the tool presented, none identified the existence of the bolt pattern. Of the five students in Group A, four identified the existence of the sketch. A two-sample unequal variance T-test was performed on these results to determine if the tool had significant impact on the user's ability to identify the sketch. The results indicate that with a 95% confidence level we can reject the null hypothesis and state that the tool does significantly impact the user's ability to identify the sketch. These preliminary results show that the tool gives the user the ability to identify actions and rationale in a more effective way than with version-controlled files alone.

6 SUMMARY

This paper has proposed a method for capturing design rationale during a design session and storing that information in a knowledge cloud.An implementation of this method was also presented which provides an opportunity to test this method against current industry-standard PLM tools that include knowledge capture capabilities.Results from experimentation show with 95% confidence that the implementation significantly impacts an engineer's ability to see and understand the process by which parts are created.

Future work will include a more robust study of the efficacy of this tool, especially the further impact of communication captureand its ability to augment the understanding of the inquiring engineer.

REFERENCES

[1] Bracewell, R.; Wallace, K.; Moss, M.; Knott, D.: Capturing Design Rationale. Computer-Aided Design, 2008: 1-14.

[2] Burge, J. E.; Brown, D.: Rationale-Based Support for Software Maintenance, In A. H. Dutoit, R. McCall, I. Mistrik, B. Paech, Rationale Management in Software Engineering, The Netherlands: Springer-Verlag Berlin Heidelberg, 2006

[3] Conklin, J.; Begeman, M. L.: gIBIS: A Hypertext Tool for Team Design Deliberation, Proceedings of the ACM conference on Hypertext. Chapel Hill, 1987, 247-251.

[4] Conklin, J.; Selvin, A.; Shum, S. B.; Sierhuis, M.: Facilitated hypertext for collective sensemaking: 15 years on from gIBIS, Proceedings of the 12th ACM conference on Hypertext and Hypermedia, Arhus: ACM, 2001, 123-124.

[5] Connolly, T.; Begg, C.: Database Systems: A Practical Approach to Design, Implementation and Management, Boston: Addison-Wesley, 2009.

[6] Dutoit, A. H.; McCall, R.; Mistrik, I.; Peach, B.: Rationale Management in Software Engineering: Concepts and Techniques, In A. H. Dutoit, R. McCall, I. Mistrik, & B. Peach, Rationale Management in Software Engineering, Netherlands: Springer-Verlag Berlin Heidelberg, 2006.

[7] Grudin, J.: Evaluating opportunities, In T. Moran and J. Carroll, Design rationale: concepts, techniques, and use, Hillsdale, NJ: L. Erlbaum Associates Inc, 1996.

[8] Hejlsberg, A.; Wiltamuth, S.; Golde, P.: The C# Programming Language, Boston: Pearson Education, 2006.

[9] Horner, J.; Atwood, M.: Effective Design Rationale: Understanding the Barriers, In A. Dutoit, R. McCall, I. Mistrik and B. Paech, Rationale Management in Software Engineering, The Netherlands: Springer Berlin Heidelberg, 2006

[10] Klein, M.: Capturing Geometry Rationale for Collaborative Design, Proceedings From the Sixth IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Cambridge, 1997, 24-28.

[11] Lee, J.: Design Rationale Systems: Understanding the Issues, IEE Expert, 1997, 78-85.
[12] Mix, K. J.; Jensen, C. G.; Ryskamp, J.: Using Global Communication Trends for Automated Design Rationale Capture, Accepted for publication in Tools and Methods for Competitive Engineering. 2010.
[13] Moran, T. P.; Carroll, J. M.: Design Rationale: Concepts, Techniques, and Use, Mahwah: Lawrence Erlbaum Associates, 1996.
[14] Myers, K. L.; Zumel, N. B.; Garcia, P.: Automated Capture of Rationale for the Detailed Design Process, Proceedings of the Eleventh Conference on Innovative Applications of Artificial Intelligence, AIAA, 1999, 876-883.
[15] Powell, G.: Beginning Database Design. Indianapolis: Wiley Publishing, 2006.
[16] Rob, P.; Coronel, C.: Database Systems: Design Implementation, and Management, Danvers: Boyd and Fraser Publishing Company, 1995.
[17] Ryskamp, J.; Mix, K. J.; Jensen, C. G.: Leveraging Design Rationale to Improve Collaboration in Multi-user CAD, Accepted for publication in Tools and Methods for Competitive Engineering. 2010.
[18] Shipman, F. M.; McCall, R. J.: Integrating Different Perspectives on Design Rationale: Supporting the Emergence of Design Rationale from Design Communication, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 1997, 11, 2.
[19] Zdrahal, Z.; Mulholland, P.; Valasek, M.; Bernardi, A.: Worlds and Transformations: Supporting the Sharing and Reuse of Engineering Design Knowledge, International Journal Human-Computer Studies, 2007, 959-982.