Transdisciplinary Engineering for Complex Socio-technical Systems
K. Hiekata et al. (Eds.)

doi:10.3233/ATDE190164

Integrating Knowledge Objects and Design Descriptions to Support Maintenance of Design Automation Systems

Morteza POORKIANY[1] and Joel JOHANSSON
Jönköping University, Sweden

Abstract. Design automation systems are implemented by many manufacturing companies to automate the repetitive and time-consuming design tasks. By automating such tasks, the designers have more time to focus on creativity and offer more customized solutions to the customers.

To automate a design task, first, the design knowledge should be captured from designers. This type of knowledge which is usually understandable by humans should be structured and formalized. Next, computer codes and scripts (that are mostly understandable by computers/expert persons) are created to execute the knowledge and provide the desired output.

To support maintenance of computer codes and scripts in a design automation system, it is necessary to know what, how and why about that piece of code/script. In order to support maintenance of the systems, we represent the system's knowledge in form of knowledge objects. Knowledge objects are executed in run time and consist of two parts: computer readable and human readable. The focus in this paper is on the human readable which we call it "design description". A MOKA-based framework is provided to create design descriptions for the computer readable parts. The design descriptions help engineers to understand and if needed update the computer readable parts, which in a wider aspect support maintenance of the whole system. E-books were used as a way to represent the design descriptions and a case study is provided to explore the results of the research.

Keywords. Design Description, Knowledge Object, Design Automation, Design Rationale, MOKA.

Introduction

Intense competition on the global market forces manufacturing companies to continuously work on improving their development processes to shorten the lead time. To achieve that goal, some companies focus on automation of design tasks. The targeted design tasks to be automated are usually the tasks that do not require so much creativity and can be performed by the computer systems. The objective of the automation in design is mainly to reduce the time for design and development by performing time consuming and repetitive tasks such as adapting the product according to new specifications.

To run a design automation system, the design knowledge should be represented in a suitable format for execution. To make the design knowledge suitable for machines,

[1] Corresponding Author, Email: morteza.poorkiany@ju.se.

first the knowledge should be captured from designers. This type of knowledge is usually recorded in form of loose text. Then the knowledge should be structured to provide a more formalized representation that is easier to understand and interpret. Finally, the formalized knowledge will be classified into smaller objects to be inferred and executed by computer. Computer codes will be required for execution.

Usually the knowledge that is executed by the automation system is the result of the design. For example, a CAD file, a design table or design rules are the results of activities that were performed during the design process and are executed by the system. To maintain the system, these CAD files, design rules and tables along with the computer codes should be updated. However, updating computer codes and rules and files without any explanation about them is a challenging task specially if the system was developed long time ago or if the knowledge expert who was involved in development of the system is no longer available in the company for consulting.

Design knowledge can be divided into two parts: One is design definition describing the results of the design, without any explanation concerning the reasons and argumentations behind the design. Such information can more often answer the "what" question. A CAD model or design table are some examples of design definition which are mostly based upon insights, experience, trade-offs, calculation, simulations, etc. The other type of knowledge is design rationale explaining the purpose and reasons behind the design in more details. Design rationale provides a better understanding for design definition and often aims at explaining the product in the way it is designed answering the "why" and "how" questions. For instance, "why a CAD model looks like as it is?" or "how a design concept was evaluated?"

The importance of design rationale in supporting maintenance of design automation systems was addressed in [1]. Further, the concept of "design description" as a way to document product knowledge was introduced by Elgh [2]. Design description includes both design definition and design rationale.

In this paper, the concept of "knowledge object" as a way to categorize the design knowledge into small pieces that the execution of pieces can be scheduled by computer is used. A knowledge object consists of a computer readable part (codes and scripts) and a human readable part. Design descriptions are provided for the human readable part to document both design definition and design rationale corresponding to the computer codes and scripts in each knowledge object (more detail about knowledge objects in chapter 1). The design descriptions provide the most important information that is necessary for maintaining the computer readable parts.

A design automation system that automates a set of design tasks in the design process is discussed in this paper to explore the use of design descriptions for system maintenance. The system is developed based on the concept of knowledge object. The design activity is divided into design tasks and design knowledge is represented in form of knowledge objects for each design task. A framework inspired from MOKA (a methodology for developing knowledge-based engineering systems) [3] is developed. The framework supports documentation of the design descriptions. The framework has been tested through the design automation system.

1. Frame of reference

1.1. Knowledge Objects

Knowledge objects have been defined as " bundles of human comprehensible knowledge representations and computer routines for the automated application of the represented knowledge" [4]. Knowledge object consists of two parts: automation part and design description part (see figure 1). The content of the automation part is represented in a way that is readable by machine, while the intention with the content of the design description part is to be human readable.

Design descriptions in the knowledge objects should be documented in a way to support the engineers to understand know-what (design definition); know-how and know-why (design rationale). Such an understanding helps the engineers when they need to perform changes in the process, help them when they search for errors or help training new engineers to understand the design process of the tooling. The required information, in addition to design definition, can be for example, explanation about the objective of the knowledge object, input and output parameters, failure mode (what could go wrong when executing the knowledge object), explanation about the rules and logics behind them, and or a contact person in case of need for more detail information.

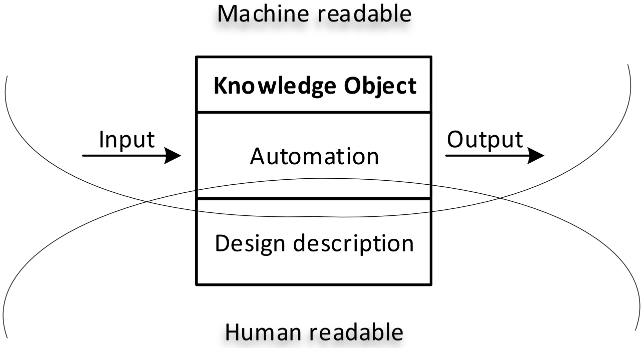


Figure 1. A schematic view of knowledge object adapted from [4].

1.2. Methods for formalizing and structuring design knowledge

Knowledge Based Engineering (KBE) studies methodologies and technologies for capture and reuse of product and process knowledge and aims to reduce time and cost of the development process by automating the repetitive design tasks [5]. The major benefit of KBE is to save time and cost. Another benefit of KBE is its integrated modeling approach, where the design knowledge is maintained in a central representation. This allows leverage of a shared knowledge base and offers domain-specific views of a design problem [5].

Research in KBE has led to introducing methodologies for supporting development of KBE systems. The KBE methodologies mainly provide frameworks for formally capturing the design knowledge within a system that can infer and act on the captured knowledge [6]. A well-known methodology in KBE is MOKA (methodology and software tools oriented to knowledge based engineering applications) [3]. MOKA provides a framework for storage and representation of knowledge in such a way that will be useful in a KBE system. MOKA is expressed in accompanying two models: one is the informal model which collects the knowledge from experts, documents and computer files in types of illustrations, constraints, activities, rules, and entities. The other one is a formal model to prepare the knowledge in a form that is suitable for computer systems and programming.

MOKA mainly focuses on "capture", i.e. collecting and structuring the knowledge, and "formalize" i.e. translating the informal model into a formal model. Structuring results in a user-friendly informal model that is understandable by both expert persons (they are usually engineers involved in design) and knowledge engineers (they structure and formalize the expert's knowledge). The intention of the formal model is to provide a more precise representation of knowledge. The knowledge is split into smaller objects and the objects are classified within categories. The formal model is to enable the last transition between the raw knowledge and KBE platforms, therefore, there is still no need for formal programming skills. The formal model is understandable by trained people with the opportunity to be interpreted by computer. [3]

The output of the structuring step is a set of linked ICARE forms. ICARE forms are the knowledge units and are used to store the knowledge in standardized forms. ICARE forms include illustration form, constraint form, activity form, rule form and entity form. Entity and activity are the two most important forms in ICARE. Entity describes the product with three key areas as function, structure and behavior. Activity describes the design process with rules attached to it. According to MOKA, the activity forms capture the "what" of the process while the rule forms intend to capture the "how" of the process. More detail about the activity explaining the "why" can be captured by textual design description in the activity form.

For each form a distinct template with separate fields is provided to record the specific parts of the knowledge unit. For example, the template for the activity form contains information such as author of the form, date, version number, status (in progress, complete, verified), a design description about the activity (explaining "what"), list of rules involved during the activity execution or link to the rule form (explaining "how"), objective of the activity, illustration, and input and output information.

2. Proposed framework

In this paper a MOKA-based framework is proposed for creating knowledge objects and design descriptions. According to MOKA, to automate the design process, the following steps shall be followed:

- First, the top layer of the process and its strategy and objective should be determined.
- Next, the process should be split into sub activities.

- Then, a top down approach is applied to decompose the activities to design tasks.
- Finally, one or a set of knowledge objects are developed for each design task by identifying their objectives and their input and output parameters.

Step 1, determining top layer of the process and its strategy, indicates the domain of the knowledge in the automation system. A knowledge domain is a set of knowledge objects that can target different domain such as design, simulation, etc. This arrangement makes the implemented system flexible to changes, enabling the plug and play of knowledge objects. The execution of the knowledge objects within the knowledge domain is scheduled by an inference engine. The inference engine arranges the automated knowledge in the knowledge objects in an executable order based on the input and output parameters of the automation part of the knowledge objects. This can be done prior to execution or dynamically in run time.

Once the domain is determined, the process is split into sub activities with defined inputs and outputs. For each sub activity, one knowledge object or a set of knowledge objects are to be defined.

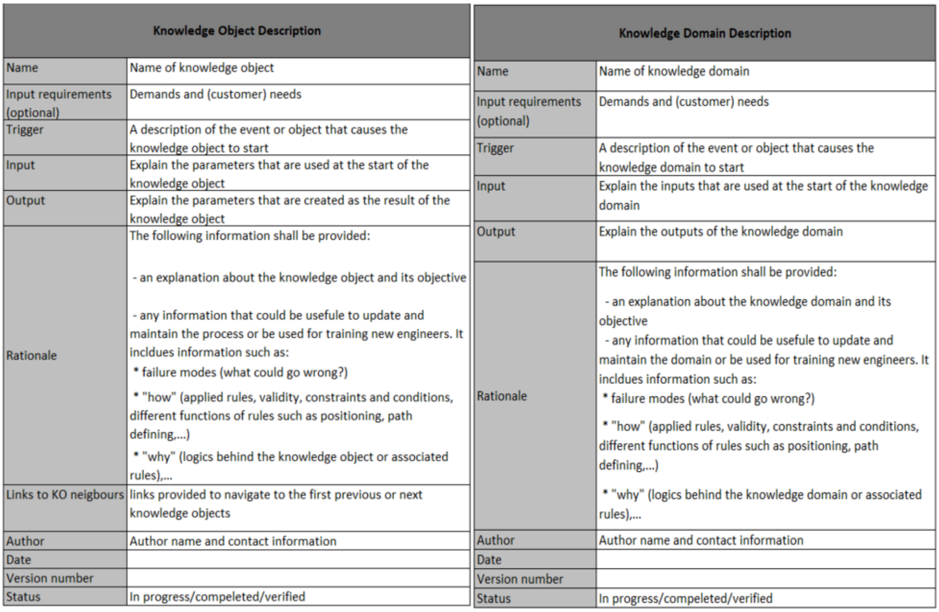
2.1. Design descriptions: knowledge domain and knowledge object descriptions

Depending on the level of the design description, if it is in the domain level or object level, design descriptions are categorized into knowledge domain descriptions and knowledge object descriptions. To support documentation and standardization of design descriptions, templates are provided for both knowledge domain and knowledge objects. Figure 2 shows the provided templates that are inspired from the activity form in MOKA. The fields in the templates are designed in a way to support know-what, know-how and know-why. The information in the design descriptions help the engineers when they need to perform changes in the process, help them when they search for errors or help training new engineers to understand the design process of the tooling.

The main difference between knowledge object descriptions and knowledge domain descriptions is the links to knowledge object neighbors in knowledge object descriptions. Links are to be provided to navigate to the first previous or first next knowledge objects to trace the executed objects.

3. Case study

Howtomation Suite is an automation tool that was developed by one of the authors of this paper [7]. Howtomation Suite is a constraint solver tool that is developed based on knowledge objects to automate design activities. The design process is broken down into a set of activities that need to be solved. In the system the activities are represented in form of knowledge objects. Based on the input information for a knowledge object, design rules are provided to execute the knowledge object. The output of the knowledge object will be input for the next knowledge object in the process until the problem is solved.



Knowledge Object Description	
Name	Name of knowledge object
Input requirements (optional)	Demands and (customer) needs
Trigger	A description of the event or object that causes the knowledge object to start
Input	Explain the parameters that are used at the start of the knowledge object
Output	Explain the parameters that are created as the result of the knowledge object
Rationale	The following information shall be provided: - an explanation about the knowledge object and its objective - any information that could be usefule to update and maintain the process or be used for training new engineers. It incldues information such as: * failure modes (what could go wrong?) * "how" (applied rules, validity, constraints and conditions, different functions of rules such as positioning, path defining,...) * "why" (logics behind the knowledge object or associated rules),...
Links to KO neigbours	links provided to navigate to the first previous or next knowledge objects
Author	Author name and contact information
Date	
Version number	
Status	In progress/compeleted/verified

Knowledge Domain Description	
Name	Name of knowledge domain
Input requirements (optional)	Demands and (customer) needs
Trigger	A description of the event or object that causes the knowledge domain to start
Input	Explain the inputs that are used at the start of the knowledge domain
Output	Explain the outputs of the knowledge domain
Rationale	The following information shall be provided: - an explanation about the knowledge domain and its objective - any information that could be usefule to update and maintain the domain or be used for training new engineers. It incldues information such as: * failure modes (what could go wrong?) * "how" (applied rules, validity, constraints and conditions, different functions of rules such as positioning, path defining,...) * "why" (logics behind the knowledge domain or associated rules),...
Author	Author name and contact information
Date	
Version number	
Status	In progress/compeleted/verified

Figure 2. Templates for knowledge object description and knowledge domain description.

An industrial example is used to explain how design descriptions in knowledge objects can be documented. The example targets the development process of a company that produces roof racks that are used for carrying skies and sport instruments on different car models. A component called bracket is used to connect the rack to the roof and hold it (see figure 3). A roof rack that is developed for a specific car can be adapted to a new car brand by redesigning the bracket (there are other parts that should be adjusted but for the sake of simplicity we don't consider them in this example). Slight changes in the design of an existing bracket allows reuse of a previously developed roof rack on a new car model.

All the previously designed brackets are stored in a database in the company. By entering a new car into the market, the engineers first search the database to find an applicable bracket for the new car. If they find a bracket that suits the new requirements the manufacturing of the bracket is started quickly by reusing the associated production tooling that were developed in previous projects. If none of the existing brackets are suitable for the new car, a new bracket is to be developed. Manufacturing of a bracket requires performing press mechanism on raw material. A tooling is required to be used in the press machine to form the required shape of the bracket. So, for developing a new bracket a new tooling should also be developed.

The example presented here targets the design process of a new tooling. The design process can be split into a number of tasks that are repeated every time a new tooling is to be developed. The objective is to define knowledge objects for the tasks and use Howtomation Suite to execute the knowledge objects automatically based on some input information that are set by the engineers. The output would be a complete CAD model for the tooling.

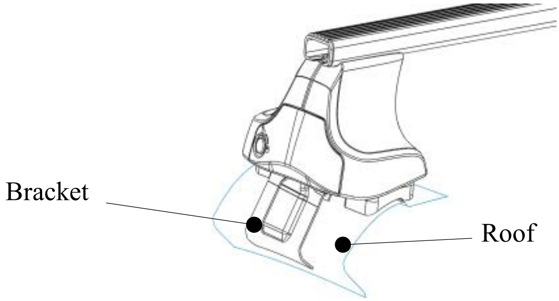


Figure 3. Use of bracket in installing a rack on the roof of a car.

3.1. Knowledge objects and design descriptions in the design automation system

A screenshot of Howtomation Suite and knowledge objects and parameters created for automatically developing the tooling is presented in figure 4. Due to the large size of the process, only half of the process is shown. The rectangular shapes represent knowledge objects and the oval shapes represent parameters. Executed knowledge objects are green, triggered knowledge objects are yellow and unreachable knowledge objects are red. Also, known parameters are green while unknown parameters are red.

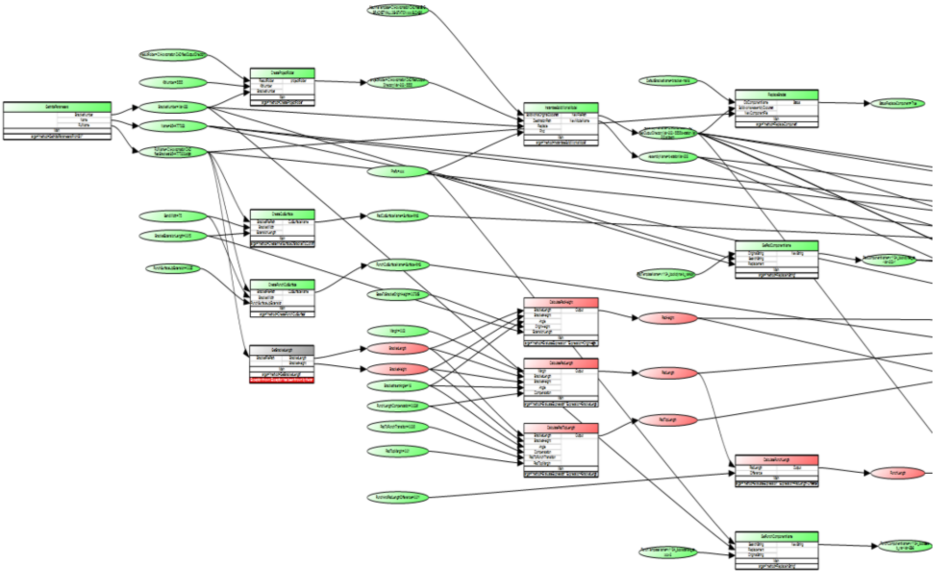


Figure 4. An overview on Howtomation Suite (print in color).

The process starts from the left side where the first knowledge object gets the initial parameters such as “bracket name” parameter from bracket’s CAD model. The output from the first knowledge object is then used by the next knowledge objects and so on.

For the design of the tooling, there are in total 22 knowledge objects such as GetInitialParameters, CreateProjectFolder, InstantiateSolidWorksModel, GetPad, GetPadComponentName, CalculatePadLength and AddFilletToPad, and 51 parameters such as AssemblyName, BracketLength, PunchHeight and PadLength.

Figure 5 shows a knowledge object called "AddExtrudeInPad" and its design description in Howtomation Suite (on the right side) which was created based on the template shown in figure 2. To integrate the design descriptions into Howtomation Suite, e-book technique was used. EPUB (electronic publication) which is a file format for electronic books was used to enable flexible and accessible documentation of knowledge objects. Each design description can contain texts, pictures and movies. The design description is stored as an xhtml webpage and is displayed in a window on the right-hand side of Howtomation Suite when the corresponding knowledge object is selected.

4. Discussion and conclusion

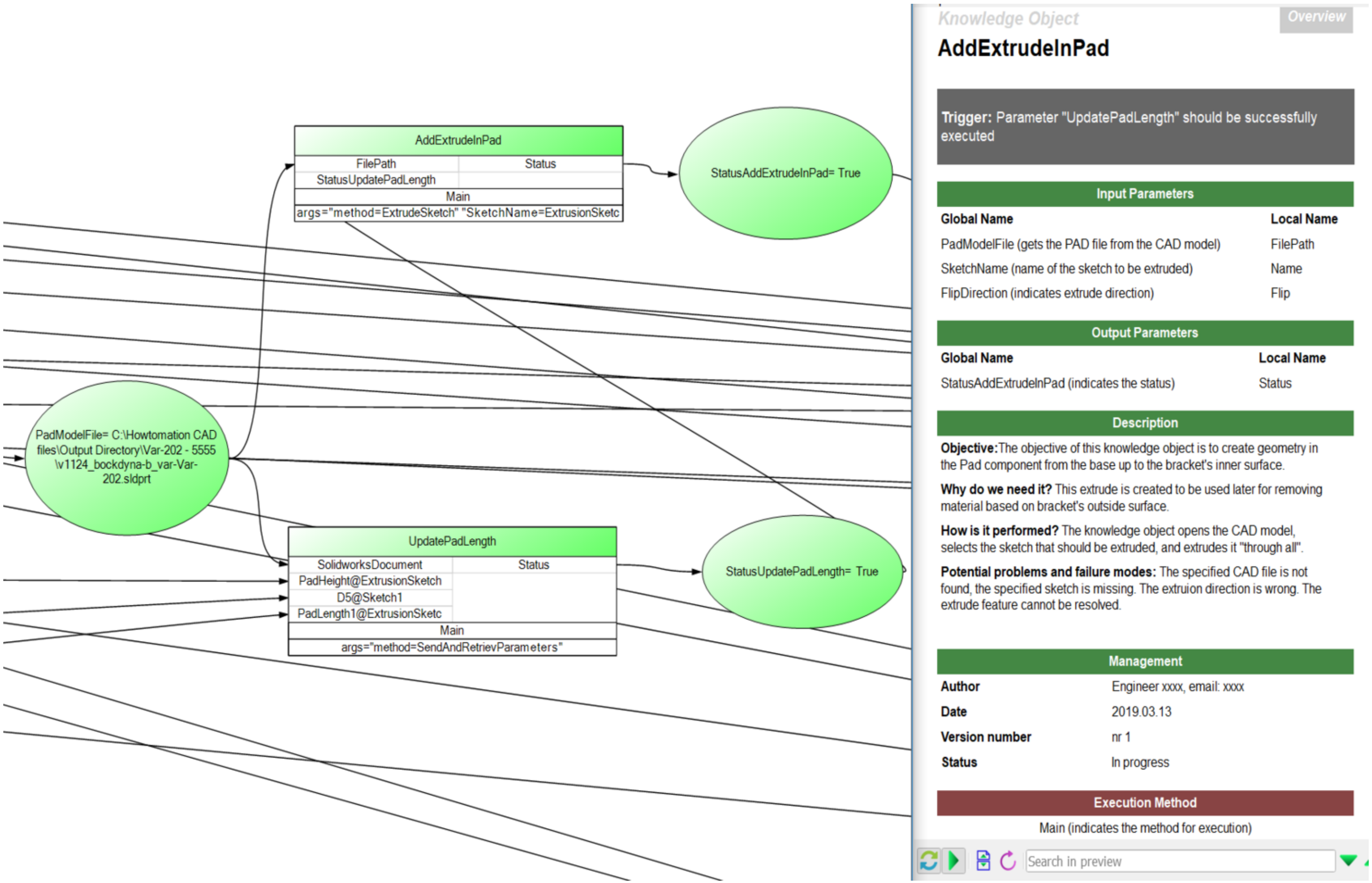
Providing methods and approaches for capturing design information including design rationale and sharing it for the design members supports collaboration and increases accuracy and efficiency in the development process. Usually, when it comes to documentation of design knowledge, specially design rationale, a challenge is to know what should be captured that will most probably be used in the future to support system maintenance. The situation becomes more critical when the knowledge expert who was involved in the develoment of the system is no longer available and a new engineer has to take over the responsibility.

The importance of design rationale in system maintenance has been addressed by research community, however, the major issue is to keep the documentation simple, thorough and understandable by others. The provided templates for knowledge object and knowledge domain descriptions is a way to standardize documentation of the descriptions that include both design definition and design rartionale. The templates were inspired from MOKA A-forms that target design activities. According to MOKA, the activity form captures the "what" of the process, however, more detail about the activity explaining the "why" and "how" can be captured by textutal description in the activity form. This can be seen in the proposed templates where the intention is to capture all the "what", "why", and "how" of the process.

The determined fields in the templates are the results of several discussions with author of this paper, system developer and engineers in the company as well as inspiration from previous research in the design rationale field. The content of the fields were tailored according to the company's needs and they can be adjusted if the template is to be used in another context or organization.

Electronic publication is a file format for electronic books that was developed by International digital Publishing forum. There exist free and commercial EPUB readers for all platforms. Several web browsers are capable of reading EPUB as well. The presented automation system in this paper allows text, pictures and movies in a design description. The design description is stored as an xhtml webpage. It is possible to generate an e-book including all design descriptions in a domain.

A design automation system that automates a set of design tasks in the design process was discussed in this paper to explore the use of knowledge objects for system maintenance. The system is developed based on the concept of "Knowledge Object".


AddExtrudeInPad
FilePath
Status
StatusUpdatePadLength
Main
args="method=ExtrudeSketch" "SketchName=ExtrusionSketc
StatusAddExtrudeInPad= True
PadModelFile= C:\Howtomation CAD files\Output Directory\Var-202 - 5555 \v1124_bockdyna-b_var-Var-202.sldprt
UpdatePadLength
SolidworksDocument
Status
PadHeight@ExtrusionSketch
D5@Sketch1
PadLength1@ExtrusionSketc
Main
args="method=SendAndRetrievParameters"
StatusUpdatePadLength= True
Overview
Search in preview

Knowledge Object

AddExtrudeInPad

Trigger: Parameter "UpdatePadLength" should be successfully executed

Input Parameters

Global Name	Local Name
PadModelFile (gets the PAD file from the CAD model)	FilePath
SketchName (name of the sketch to be extruded)	Name
FlipDirection (indicates extrude direction)	Flip

Output Parameters

Global Name	Local Name
StatusAddExtrudeInPad (indicates the status)	Status

Description

Objective: The objective of this knowledge object is to create geometry in the Pad component from the base up to the bracket's inner surface.

Why do we need it? This extrude is created to be used later for removing material based on bracket's outside surface.

How is it performed? The knowledge object opens the CAD model, selects the sketch that should be extruded, and extrudes it "through all".

Potential problems and failure modes: The specified CAD file is not found, the specified sketch is missing. The extruion direction is wrong. The extrude feature cannot be resolved.

Management

Author	Engineer xxxx, email: xxxx
Date	2019.03.13
Version number	nr 1
Status	In progress

Execution Method

Main (indicates the method for execution)

Figure 5. A knowledge object and its design description in Howtomation Suite.

Knowledge object contains both design definition and design rationale for computer routines and can be executed by computer. The advantage of providing design descriptions for knowledge objects that describe the objective, input and output parameteres, rationale, as well as corresponding author, the correct version of the knowledge object and the responsible person that can be contacted in case of need was emphasized by the company representative. More study is required to determine granularity on which knowledge objects are created. A challenge is decomposing the design activities into design tasks and providing knowledge objects for each task.

Acknowledgement

This paper was a part of a research carried out within the PROACT project, funded by the Jönköping County.

The authors are grateful for the provided research possibility in the industrial partner.

References

[1] M. Poorkiany, *Managing design rationale in the development of product families and related design automation systems*, PhD thesis, Jönköping University, School of Engineering, 2017.

[2] F. Elgh, *Modeling and management of product knowledge in an engineer-to-order business model.* in *DS 68-6: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 6: Design Information and Knowledge, Lyngby/Copenhagen, Denmark, 15.-19.08. 2011.*

[3] M. Stokes, *Managing engineering knowledge: MOKA: methodology for knowledge based engineering applications*. Vol. 3, Professional Engineering Publishing, London, 2001.

[4] J. Johansson and F. Elgh, *Knowledge Objects Enable Mass-Individualization*, in E. Andrés-Pérez, L.M. González, J. Periaux, N.R. Gauger, D. Quagliarella and K.C. Giannakoglou (eds.) *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*, 2019, Springer International Publishing, pp. 371-386.

[5] W.J. Verhagen, et al., *A critical review of Knowledge-Based Engineering: An identification of research challenges.* Advanced Engineering Informatics, 2012. 26(1): pp. 5-15.

[6] C.B. Chapman and M. Pinfold, Design engineering—a need to rethink the solution using knowledge based engineering. *Knowledge-based systems*, 1999. 12(5): pp. 257-267.

[7] J. Johansson, Howtomation© Suite: A Novel Tool for Flexible Design Automation. *Advances in Transdisciplinary Engineering*, Vol. 2, IOS Press, Amsterdam, 2015, pp.327-336.