INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

A WORKING KNOWLEDGE MODEL FOR SUPPORTING EARLY DESIGN THROUGH VISUAL TOOLS

Srikanth Devanathan[1] Christian Sauter[2], Albert Albers[2] and Karthik Ramani[1]
(1) School of Mechanical Engineering, Purdue University, USA
(2) Institute for Product Development (IPEK), University of Karlsruhe (TH), Germany

ABSTRACT

In this article, we present a working knowledge model (WKM) that collates information from commonly used visual tools to support iterations in conceptual design of mechanical artifacts. The key aspects of the WKM are the notion of mappings and the ability to handle multiple descriptions and alternatives that are encountered during the design process. We show how common visual-models and tools such as House of Quality, function-structure, requirement and parametric diagrams can be described using the elements of the WKM. This connection serves as the basis for development of computational system for (1) acquiring, (2) displaying, and (3) manipulating the working knowledge with minimal disruption to the designer. A case study of the design of a mechanical neck for a humanoid robot is presented. This work demonstrates the ability of the model to accommodate information associated with these tools.

Keywords: Knowledge model, Product model, Visual tool, Robot design, Working Knowledge

1 INTRODUCTION

Capturing the knowledge in design is usually identified as one of the important areas in design research [1]. Even though the majority of design knowledge is tacit, capturing the explicit knowledge has been shown to improve the quality of designs by providing a common understanding of the design problem among stakeholders in design. Computer based representation of design knowledge allows (1) proper documentation, (2) coordination of design activities, (3) reuse of design knowledge, and in the context of this paper, (4) provides a clear interface for several design support tools in design.

Traditional knowledge acquisition techniques use specialized forms where the users manually enter the relevant information in the appropriate slot. Such attempts to capture the knowledge within an organization in one database with a highly powerful and specialized tool have often failed so far [2]. One reason for this was that users had the feeling that using specialized knowledge management systems caused too much effort in addition to existing tasks. So, the benefits of these systems did not outweigh the disadvantage of additional effort for the individual user. It can be inferred that, to be successful, any knowledge based system should (1) allow them to focus on their core task, (2) not demand additional effort from the users, and (3) enable them to work more efficiently.

In this work we only consider knowledge that is or can be made explicit during a design process. Additionally, we restrict ourselves to just the knowledge *about* the current design problem. Since the knowledge about the design constantly evolved during the course of the design process, we call this the *working knowledge*.

Therefore, to acquire the working knowledge we observe that, as a part of structured design processes, designers express a portion of this knowledge using well-known representation schemes. Horvath identifies five distinct classes of knowledge representations, namely [3]:

(1) Pictorial (sketches, charts, diagrams, etc.),
(2) Symbolic (decision tables, argumentation diagrams, scripts and scenarios, etc.),
(3) Linguistic (verbal and textual communication, etc.),
(4) Virtual (computer geometric models, animations, simulations, etc.), and,
(5) Algorithmic (mathematical expressions, computational procedures, etc.).

Examples of such well-known "visual" representation schemes include Quality Function Deployment (QFD)[4], Function-Component-Matrix, Morphological Matrix[5], Design Structure Matrix, Pugh Concept selection table and even 3D geometry models. We note that computer based virtual representations are commonly used in later stages of design (where preciseness is important), whereas pictorial and symbolic representations dominate early design (where vagueness is important) [6]. We collectively designate pictorial and symbolic models as 'visual' models and the corresponding tools (or methods) that support decision making as visual tools (or methods).

Since most design decisions are made based on the analysis of these models, it is hypothesized that working knowledge can be acquired through a proper study of these visual models with minimal intrusion (see Figure 1).

In this paper, we explore the feasibility of such an approach using three example visual tools using the design of a robot-neck as a case study. In order to describe the content of these visual tools, we identify a minimal set of elements that constitute the working knowledge. These elements are derived by extending existing state-of-the-art product models such as the Core Product Model [7] to address additional aspects of the working knowledge that are not found in product models.

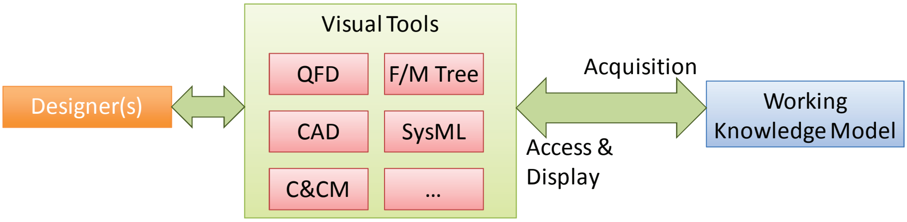


Figure 1. The knowledge (information & decisions) contained within the WKM is acquired through commonly used decision-support tools; same visual models can also be used to present relevant information to designer(s)

2 ELEMENTS OF THE WORKING KNOWLEDGE MODEL

There are two common views about the nature of knowledge in general: structural view- knowledge as the content of a representation, and functional-knowledge as the capability to solve problems [8]. A significant difference between a design database and a knowledge model is [9] the commitment to ontology, surrogate structure and also other content that supports synthesis and problem solving activities. In this section we describe the meta-model (structure) for representing the knowledge (content) that acts as a surrogate structure and supports synthesis and problem understanding & solving during design.

Restricting to design, Klein [10] identifies three main categories of design knowledge: (1) general domain knowledge, (2) case-specific object level knowledge, and (3) problem solving & control knowledge. In this view, general domain knowledge includes knowledge describing the relations between function, behavior, structure; case-specific knowledge consists of requirements and (possibly partial) design descriptions. During the design process, the domain knowledge of the designer engineers can be assumed to change little in comparison to case-specific knowledge [11]. In this work, we further restrict ourselves the "working" aspect of the design knowledge, i.e., only that which is known about the current design and can be expressed explicitly.

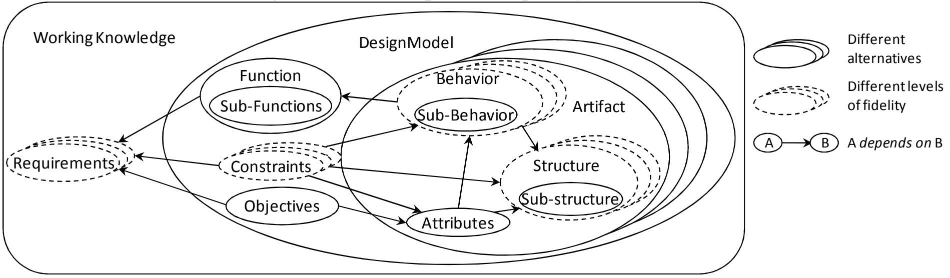


Figure 2. Entities considered within the working knowledge model

Figure 2 above shows the elements of the working knowledge that are considered in this implementation. The *DesignModel* corresponds to the generic artifact that is being designed to meet *Requirements*. The *DesignModel* consists of *Function*, *Behavior* and *Structure* information as well as *Constraints* and *Objectives*. The function, structure and behavior of a design can be described in a hierarchy. The *DesignModel* may give rise to several alternatives, all of which provide the same functionality and satisfy the constraints. The notions of *DesignModel* and *Artifact* instance are particularly important during configuration design: the design model contains the information about all the alternatives for sub-systems and the topology of the system. By using an appropriate configuration method, different alternatives can be generated for the overall system.

We also allow requirements, constraints, behavior and structures to be represented in multiple levels of fidelity. A simple line sketch, 2D drawing and a 3D geometric model can all represent the geometry of design but do so at different levels of fidelity. Similarly, a qualitative relationship between attributes, an algebraic relationship between the attributes and a Simulink model all represent the behavior but at different fidelity.

2.1 Requirements

Requirements represent the explicit conditions that final design should satisfy and are stated by the stakeholders in the design. These stakeholders include customers (e.g. 'should provide a particular functionality'), end users, manufacturers (e.g. 'should be manufacturable with the available resources') and design engineers themselves. Although requirements are described textually, we assume that the designer translates them into functional requirements, constraints and objectives during the course of the design process.

2.2 DesignModel and Artifact instances

In product knowledge models (e.g. [7, 12]), an Artifact represents the actual artifact being designed. An artifact is composed of other artifacts leading to a natural extension of an assembly model. Although this modeling approach is sufficient for describing the final outcome of the design process, or, a description of a single design that is created as a succession of design operations that only add more information to existing model. However, several representations are used during design to describe the same artifact [6]. Moreover, several descriptions of the same implementation can be generated during product development depending upon the intended use; manufacturing drawings, for example, are generated based on the designed geometry and provide additional process related information.

There is no clear definition of an artifact in literature when early stages of design are considered. That is, is solution principle ([5]) (or scheme ([13]), or concept ([14]), or configuration ([15]), etc.) an artifact? They all describe the artifact at different levels of abstraction, but identifying a given artifact description as an idea or concept or embodiment is subjective in nature. To overcome this ambiguity, we consider artifact instance to be a notional idea that is assembled from information available in the highest level of fidelity.

2.3 Attributes

A design model consists of several attributes that describe the model. *Attribute* describes the characteristic property of an artifact [16].'Manufacturability', 'cost' and 'color' are examples of attributes. *Parameters* form an important specialization of an attribute. Parameters are quantities that numerically describe the artifact, and in this case the design model. Certain parameters are explicitly identified by the designer as 'performance parameters' or 'design parameters'.

2.4 Constraints

Constraints represent the computer-verifiable expressions associated with the design. Weilinga [17] differentiates constraints from requirements based on the "tone", i.e., positive or negative. Requirements are associated with a positive tone that represents what is needed from a design. Constraints, on the other hand, are associated with a negative tone and represent what is possible with the design. Constraints, in general, restrict the choice available to the designer [18]. However, in this work we do not use "tone" to differentiate between constraint and requirement because of its subjective nature. Instead, we use 'the ability of a computer system to interpret' as the criteria; this is because any constraint (in the sense of ref. [17]) that is not traced to the customer can be ascribed to requirements from any of the stakeholders including designers and manufacturers. For example, even constraints arising out of physical phenomenon, such as '*stress<yield_strength*' can be traced to a requirement '*should not fail*'. In other words, constraints can also be described as the mathematical translation of stakeholder requirements.

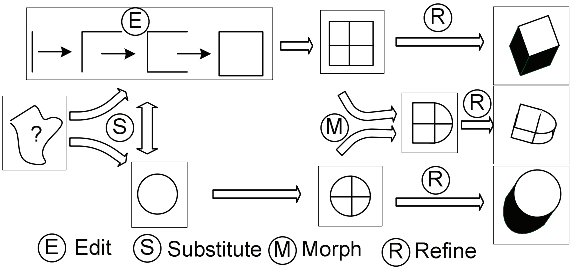


Figure 3. Relationships among DesignModels

2.5 Relationships among entities

Mappings are used to capture the relationships between objects belonging to the same entity class. These mappings indicate that the two model objects (*Attributes* or *BehaviorModels* or *Constraints*) correspond to each other in a way described within that mapping. Mappings due to substitution, refinement and composition are considered in this implementation (see Figure 3).

Substitution indicates that the involved entities can be substituted for each other; at the end of design, one of them would be *selected* for actual use. For example, the two motors shown in Figure 4 can be substituted for each other depending upon the requirements. The objects that can be substituted can come either from a catalog of finished products (or a design repository) or from explicit enumeration by the engineer such as those with different solution principles. Substitution mappings that arise from a catalog or database can have a hierarchical structure, and can have intermediate representations that capture the common elements.

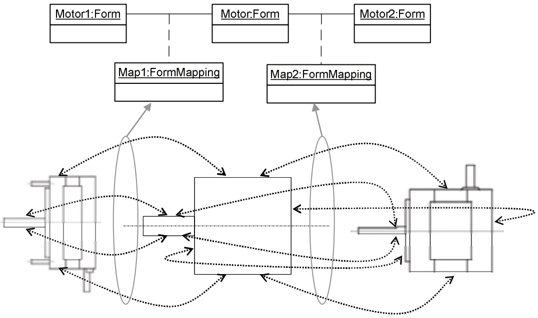


Figure 4. Example of mappings for substitution; the motor in the middle can be substituted by either Motor1 or Motor2

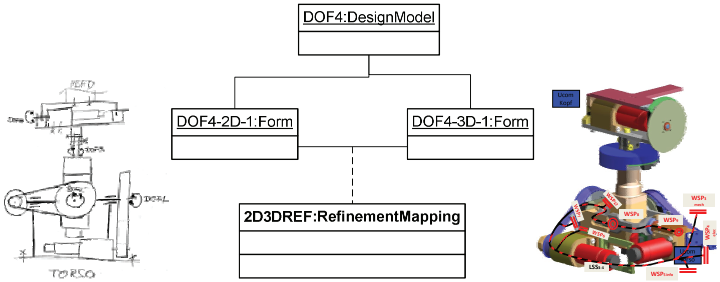


Figure 5. Example of refinement mapping applied to structure description

Refinement indicates that a design model is refined by another typically with more detail. For example, when a 2D sketch is refined by a simple 3D drawing (see Figure 5), the refinement mapping is used to capture corresponding entities between these geometries. Refinement mapping allows the design engineer to document requirements and constraints (e.g. parametric relationships) at different levels of detail: For a beam-like structure, the algebraic relationship $\delta = \frac{Wl^2}{8Eh}$ can refine a previously defined qualitative relationships $l \xrightarrow{+} \delta$ & $h \xrightarrow{-} \delta$ (read as δ decreases monotonically with h), and, can itself be refined by a structural finite element analysis model of the final design.

3 DESCRIBING VISUAL TOOLS IN DESIGN USING WKM ELEMENTS

Designers use various types of representations in their daily work [19]. A purely verbal description is often simply not suitable or less efficient for describing complex systems or problems than a visual model or, in general, a combination of verbal and visual representations [20]. Visual models support cognition, creativity, hypothesis building and reasoning [21, 22]. A large variety of different visual models exists which vary significantly in expressiveness and formality [5]. Some are easy to understand and universally applicable like bar charts or mind-maps. On the other end of the spectrum, there are highly formalized visual models which require background knowledge for effective use. These are often used by experts within a particular field; the graphical modelling languages SysML or Modelica are examples for this type of visual models.

The interaction with the knowledge model should be integrated closely into the work of designers (and in general, design teams). We achieve this integration by the use of visual models which are already being used in engineering design. The transition to a software tool or a suite of tools based on the idea of a working knowledge model would be much easier for designers if they could still use existing visual models as well as known interfaces instead of a totally new human-machine interface.

3.1 House of Quality

Quality Function Deployment (QFD) [23] has become a widely accepted method for engineering design in industry and software programs for computer support of this method are available. The main visual model of the QFD method, the House of Quality (HoQ) can be used very flexibly and it can capture various aspects of the working knowledge. The versions of the HoQ which are used most often relate customer requirements ("Voice of the customer") in the interrelations matrix to the engineering characteristics, or in other words, technical *Requirements* are related to *Attributes* of the design. The House of Quality can also be used to model objectives, constraints, alternatives, a relative ranking of requirements, and qualitative relationships among attributes.

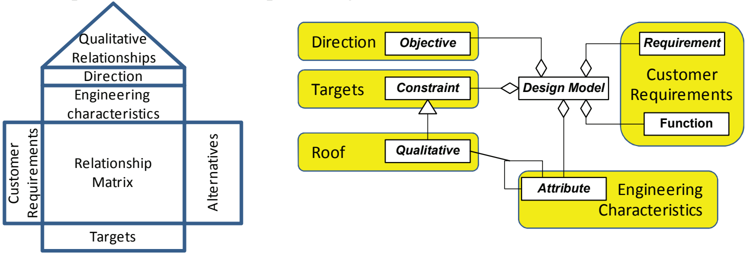
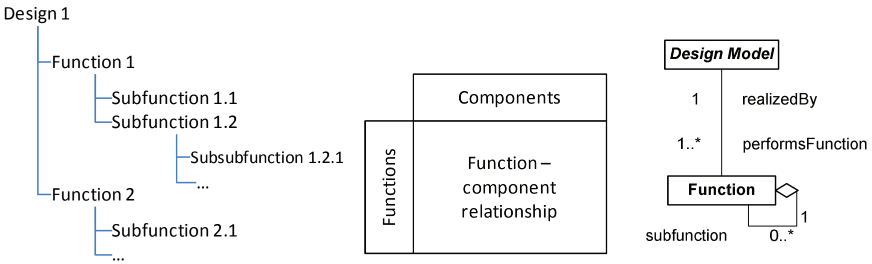


Figure 6. Description of the House of Quality using WKM elements

3.2 Function diagrams

Knowledge about *Function* is often visualized in hierarchical or procedural function structure diagrams [5]. Describing function without its respective implementation does not support the design process sufficiently as designers always have at least a vague idea of an implementation in mind when they think and argue about function. Especially function decomposition cannot be done meaningfully without relating function to form. Function-Means (F/M) trees [24] are a visual models that, in the context of WKM, assign a *DesignModel* to *Function* before a function is decomposed.

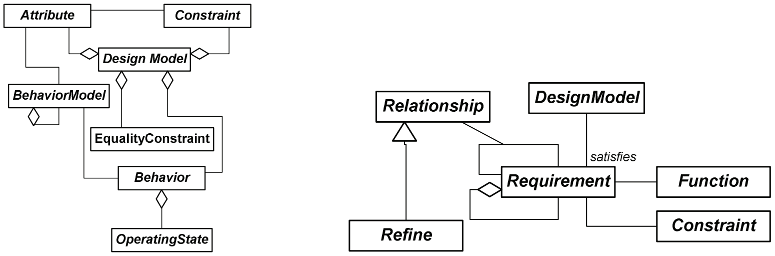


(a) Hierarchical function structure (b) Function-component matrix (c) WK Model description

Figure 7. Model of hierarchical function structure and function component matrix using WKM elements

3.3 SysML Requirement and Parametric Diagrams

The Systems Modelling Language SysML [25] is a graphical modelling language that was developed to support systems engineering processes. The SysML Requirement Diagram is used for representing *Requirements* in the WKM. Parametric Diagram can be used to describe *Attributes* and mathematical relations between these attributes that form the (equality) *Constraints* within WKM. Internal Block Diagrams are similar to class diagrams in UML2 and are used to represent the structure of *DesignModel*.



(a) (b)

Figure 8. Partial descriptions of SysML (a) Parametric diagram and (b) Requirements diagram usign WKM elements

4 DEMONSTRATION PLATFORM

The working knowledge model was implemented using JAVA and custom House of Quality, Function Component Matrix, Function and Product structure editors were also developed. The sysML authoring plugin for Eclipse [26], TOPCASED [27], was used as a viewer and editor for SysML diagrams. A dashboard (see Figure 9) showing the dependency information among these visual tools was used as the main interface for interacting with the WKM. Any visual tool can be re-opened by double clicking the corresponding icon.

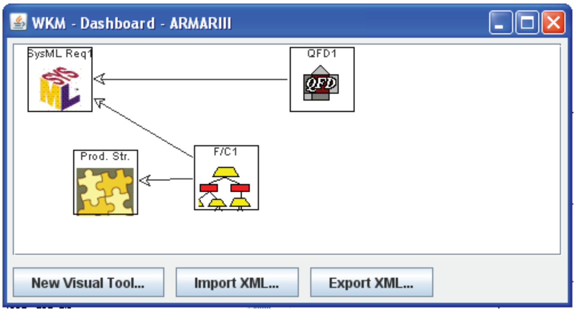


Figure 9. Main Dashboard for interacting with the working knowledge model

5 CASE STUDY

In this section we demonstrate the suitability of the visual models from the previous section to operate on the contents of WKM with three examples from the development of a robot neck. After a brief introduction to the neck of the humanoid robot ARMAR III, we show how a Function-Component Matrix, SysML Diagrams and the House of Quality are tied to WKM.

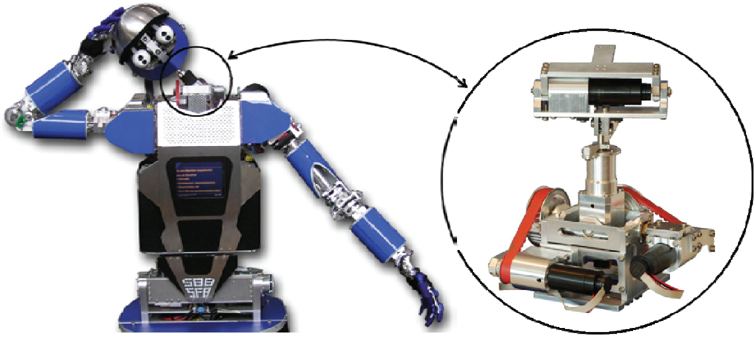
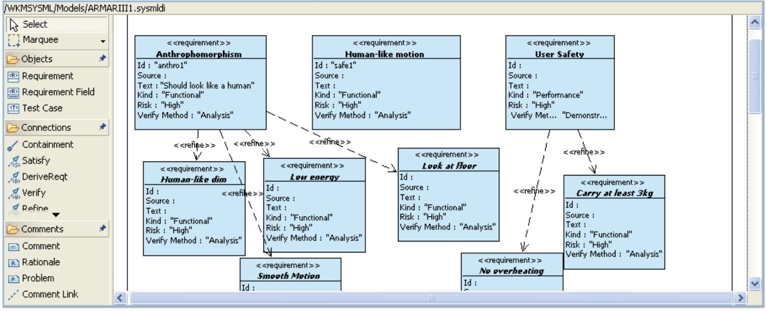


Figure 10. Upper body and neck of the humanoid robot ARMAR III

The collaborative research centre (*Sonderforschungsbereich*) 588 "Humanoid Robots – Learning and Cooperating Multimodal Robots" (SFB588) is a inter-disciplinary research project in Karlsruhe, Germany which has the objective of creating a machine that closely cooperates with humans [28]. ARMAR III as shown in Figure 10 is one of the full size humanoid robots of the collaborative research centre 588. Neck, torso, arms and the holonomic locomotion platform of ARMAR III were designed and built at the IPEK Institute of Product Development at the University of Karlsruhe (TH) in cooperation with the research partners from the SFB588.The initial set of requirements are shown in Figure 11.

For ARMAR III, the complex kinematics of the human neck was reduced to a serial kinematics with four rotational degrees of freedom (Figure 12). Three degrees of freedom were realized in the lower part of the neck. Two degrees of freedom allow the neck to move back and forth as well as to both sides. One degree of freedom allows rotation around the longitudinal axis of the neck. The upper portion of the neck (fourth degree of freedom) enables the robot to nod and to look at the ground right in front of it (an important requirement). All degrees of freedom are driven by small DC motors. For the conversion of torque and rotational speed, the drive train of each degree of freedom consists of Harmonic Drive transmissions either as only a transmission element or, depending on the needed overall gear ratio, a combination of Harmonic Drive transmission and a toothed gear belt. Angular sensors are placed at the motors and the axis of rotation for precise motion control. The motor controller boards for the neck had to be integrated into the head and torso of ARMAR III.



<<requirement>>
Anthrophomorphism
Id : "anthro1"
Source :
Text : "Should look like a human"
Kind : "Functional"
Risk : "High"
Verify Method : "Analysis"

<<requirement>>
Human-like motion
Id : "safe1"
Source :
Text :
Kind : "Functional"
Risk : "High"
Verify Method : "Analysis"

<<requirement>>
User Safety
Id :
Source :
Text :
Kind : "Performance"
Risk : "High"
Verify Met... "Demonstr...

<<requirement>>
Human-like dim
Id :
Source :
Text :
Kind : "Functional"
Risk : "High"
Verify Method : "Analysis"

<<requirement>>
Low energy
Id :
Source :
Text :
Kind : "Functional"
Risk : "High"
Verify Method : "Analysis"

<<requirement>>
Look at floor
Id :
Source :
Text :
Kind : "Functional"
Risk : "High"
Verify Method : "Analysis"

<<requirement>>
Carry at least 3kg
Id :
Source :
Text :
Kind : "Functional"
Risk : "High"
Verify Method : "Analysis"

<<requirement>>
Smooth Motion
Id :

<<requirement>>
No overheating
Id :

Figure 11. SysML requirements diagram showing the initial set of requirements for ARMARIII

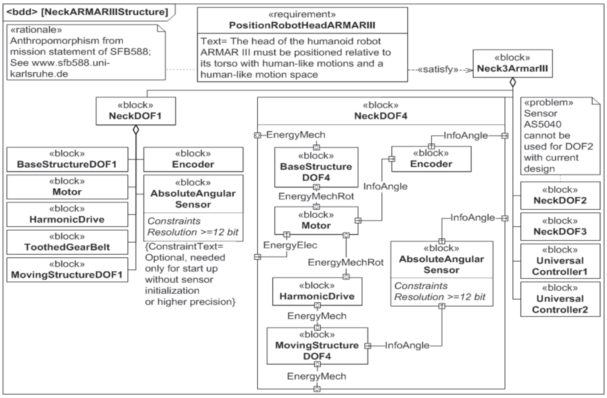


Figure 12. SysML block diagram showing the structure of ARMARIII

5.1 Function-Component-Matrix and WKM

In the function-component matrix shown in Figure 13, the functions of the robot neck are listed as rows and the components are listed as columns. Functions and components are grouped hierarchically. Square markers in the matrix fields indicate which components perform a certain function. The functions and components were extracted from the SysML requirements diagram (Figure 11) and the product structure information contained within the WKM. Some of decisions (assignment of function to components) that were made using the Function-Component Matrix and taken back in to the working model are shown in Figure 14.

	Enable "Upper Nodding" (DOF 4)								
		Tranform Energy		Define degree of freedom		Measure angular position			
	Connect mechanically to DOF3	Transform electrical energy into mechanical energy	Transform torque and angular velocity	Reduce to 1 rotational DOF	Limit range of motion	Measure motor angle	Measure joint angle	Control DOF4	Connect mechanically to Head
BaseStructureDOF4	■		■	■	■				■
Motor		■						■	
Harmonic Drive			■						
Moving StructureDOF4			■	■	■		■		
Encoder						■		■	
AbsoluteAngularSensor							■	■	
UniversalController2						■	■	■	

Figure 13. Function-component matrix for the neck (re-drawn for better clarity)

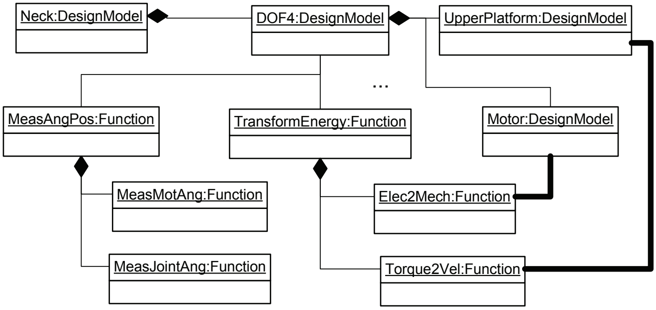
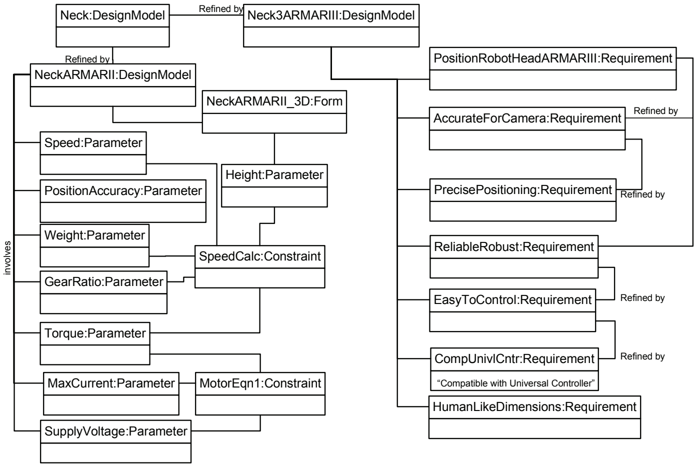


Figure 14. The instance of the WKM showing the association of the functions to the components of DOF4 (Bold line indicates the decisions that were brought back into the instance)



Partial instance of ARMARII neck

Partial listing of ARMAR III requirements

Figure 15. A portion of the working knowledge including information about neck of ARMAR II as well as requirements of ARMAR III neck

5.2 House of Quality and WKM

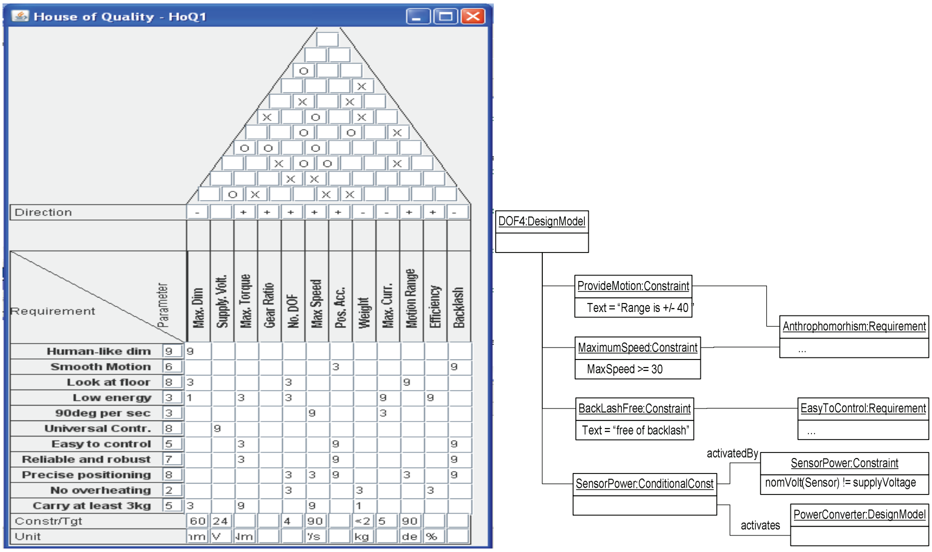
The instance of the WKM can be transformed as inputs to decision support tools and the subsequent decisions can be used to augment the working knowledge. In this section, we illustrate how the content of the working knowledge is used to setup a simple QFD and the targets set using QFD are then taken back as (soft) constraints and objectives.

Apart from the requirement information, knowledge about a previous version of the robot, ARMAR II, with an early implementation of a neck was included in the WKM. A small snapshot of the instance of NeckARMARII within the working knowledge is shown in Figure 15. This instance was described using the SysML Parametric diagram.

The House of Quality for ARMAR III was then populated using the constraints from ARMAR II such as `SpeedCalc` and `MotorEqn1` and the requirements of ARMAR III neck (see Figure 16(a)). A translator was developed to convert algebraic relationships among parameters into qualitative relationships: the gradient of the equations were obtained by algebraically differentiating the appropriate equations; the sign (positive or negative) is used to identify the correlation (positive or negative respectively) between the corresponding parameters. These correlations were then used to populate the 'roof' of the house of quality; other correlations were manually supplied. The relationship matrix was then used to set targets for the engineering characteristics (parameters). These targets were then taken back into the working knowledge of ARMAR III neck as constraints. The orientation directions that were set for some of the parameters were also incorporated as objectives within the working knowledge (see Figure 16(a)).

6 RELATED WORK

Knowledge captured through computational means during design has typically been tied very closely to the model of the underlying design problem that is being addressed. Several researchers have used selective approaches such as function-means (F/M) trees, simulation-based design, constraint-based formulations for modeling and supporting early phases of the design. For example, Schemebuilder [29] uses F/M trees for generation of schemes (concepts). O'Sullivan [30] combines F/M trees with numerical constraints to support early conceptual design. In SyDer [15], the structure of a product has been modeled and used for supporting configuration design; both the hierarchical breakdown of systems as well as interconnections within sub-systems is included within the structure. Similarly, there are several research works to support specific design activities such as constraint-based parametric design [31]; it is not intended, in this paper, to provide an exhaustive review of such approaches. Since these approaches focus on supporting certain design tasks, no formal description of the underlying model is presented.



Requirement / Parameter		Max. Dim	Supply. Volt.	Max. Torque	Gear Ratio	No. DOF	Max Speed	Pos. Acc.	Weight	Max. Curr.	Motion Range	Efficiency	Backlash
Direction		-		+	+	+	+	+	-	-	+	+	-
Human-like dim	9	9											
Smooth Motion	6							3					9
Look at floor	8	3				3					9		
Low energy	3	1		3		3				9		9	
90deg per sec	3						9			3			
Universal Contr.	8		9										
Easy to control	5			3				9					9
Reliable and robust	7			3				9					9
Precise positioning	8					3	3	9			3		9
No overheating	2					3			3			3	
Carry at least 3kg	5	3		9			9		1				
Constr/Tgt		60	24			4	90		<2	5	90		
Unit		m	V	m			/s		kg		de	%	

(a) (b)

Figure 16. House of Quality for ARMAR III neck, and the modified working knowledge model

Research in analogy based design synthesis support, such as those based on case-based reasoning (see for example, [32]) have helped develop comprehensive product models. Notably, the model presented in KRITIK [33] and KRITIK2 [21], use Structure-Behavior-Function (SBF) formalism to represent the structure of the product, its behavior based on the structure and the function achieved through the behaviors.

In the context of information modeling in design, researchers at NIST provide the most comprehensive modeling framework up to date for representing 'Artifacts'. Sudarsan et al. [34] present the Core Product Model (CPM) [35] and its extensions such as the Open Assembly Model (OAM) and Design-Analysis Integration Model (DAIM). CPM represents the function, form, behavior, architecture and other relationships associated with a product. OAM defines the hierarchical assembly relationships within the system. Based on the concepts of CPM, Xue et al. [12] use the notion of 'worlds' in capturing the operations performed on the product knowledge model during the design process as a part of an evolutionary design database.

7 DISCUSSION AND CONCLUSION

In this article, we present a working knowledge model that collates information from several (knowledge) sources including commonly used visual tools. The case presented in the paper is a theoretical study in assembling this knowledge model from the visual tools that were used during the design of a robot neck. This study has shown the feasibility of using the WKM to accommodate information associated with these visual tools.

We note that not all visual representations can be used for acquisition; the visual representation has to be semantically rich, and should pertain to the design process. For example, bar-charts and mind-maps are often used in design but cannot be mapped to the WKM in a unique manner, unlike HoQ and SysML structure diagrams. These visual models provide different mechanisms to access the content of the WKM in familiar ways depending on the context in design and personal preferences of the designers.

Additionally, there are several aspects that have to be considered within the WKM for it to provide effective computational support during design. While we have presented the grammars for few of the tools used in design, developing such parsers for a generic model is an open issue. Further research is needed in developing a system to maintain the consistency of WKM during design. We envision a Truth Maintenance System (TMS)-like system for such a purpose.

REFERENCES

[1] Szykman S., Sriram R. D., and Regli W. C. The Role of Knowledge in Next-Generation Product Development Systems. *ASME Journal of Computing and Information Science in Engineering,* 2001, vol. 1, pp. 3-11.

[2] Schütt P. The post-Nonaka Knowledge Management. *Journal of Universal Computer Science,* 2003, vol. 9, pp. 451-462.

[3] Owen R. and Horváth I. Towards product-related knowledge asset warehousing in enterprises in *International Symposium on Tools and Methods of Competitive Engineering* Wuhan, China, 2002, pp. 155-170.

[4] Dieter G. E. *Engineering Design: A Materials and Processing Approach,* 3 ed.1999 (McGraw Hill, New York, USA).

[5] Pahl G. and Beitz W. *Engineering Design: A systematic approach,* 2nd edtion ed.1999 (Springer.

[6] Goel V. *Sketches of thought: a study of the role of sketching in design problem-solving and its implications for the computational theory of the mind,* 1991 (University of California at Berkeley, Berkeley, CA, USA).

[7] Sudarsan R., Fenves S. J., Sriram R. D., and Wang F. A product information modeling framework for product lifecycle management. *Computer-Aided Design,* 2005, pp. 1-13.

[8] Reich Y. Synthesis and theory o fknowledge: general design theory as a theory of knowledge, and its implication to design. in *Engineering design synthesis*, A. Chakrabarti, Ed. 2002 (Springer-Verlag, London), 2002.

[9] Sowa J. F. *Knowledge Representation: Logical, Philosophical, and Computational Foundations,* 2000 (Brooks Cole Publishing Co., Pacific Grove, CA).

[10] Klein R. A Knowledge Level Theory of Design and Engineering. in *Globalization of manufacturing in the digital communications era of the 21st century: innovation, agility, and the virtual enterprise*, G. Jacucci, G. J. Olling, K. Preiss, and M. Wozny, Eds., 1998 (Kluwer Academic Publishers, Dordrecht), 1998, pp. 271-286.

[11] Klein R. Knowledge modelling in design–the MOKA framework. in *Artificial Intelligence in Design'00*, J. S. Gero, Ed. 2000 (Kluwer Academic Publishers, Dordrecht), 2000, pp. 77-102.

[12] Xue D., Yang H., and Tu Y. L. Modeling of Evolutionary Design Database. *Journal of computing and information science in engineering,* 2006, vol. 6, pp. 22-32.

[13] French M. J. *Conceptual Design for Engineers,* 1985 (Butterworth, London).

[14] Wilhelms S. Function- and constraint-based conceptual design support using easily exchangeable, reusable principle solution elements. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 2005, vol. 19, pp. 201-219.

[15] Feldkamp F., Heinrich M., and Meyer-Gramann K. D. SyDeR—System design for reusability. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 1998, vol. 12, pp. 373-382.

[16] Messac A. and chen W. The Engineering Design Discipline: Is its Confounding Lexicon Hindering its Evolution? *Journal of Engineering Evaluation and Cost Analysis, Decision-Based Design: Status & Promise,* 2000, vol. 3, pp. 67-83.

[17] Wielinga B. J., Akkermans J. M., and Schreiber A. T. A Formal Analysis of Parametric Design Problem Solving. in *In Proceedings of the 9th Banff Knowledge Acquisition Workshop (KAW-95)*1995, 1995.

[18] Roy U., Pramanik N., Sudarsan R., Sriram R. D., and Lyons K. W. Function-to-form mapping: model, representation and applications in design synthesis. *Computer-Aided Design,* 2001, vol. 33, pp. 699-719.

[19] Stacey M. and Lauche K. Thinking and representing in design. in *Design Process Improvement - a Review of current practice*, J. Clarkson and C. Eckert, Eds., 2004 (Springer, Cambridge), 2004.

[20] Eng N., Bracewell R. H., and Clarkson P. J. Studying the Role of Narrative Across Aerospace Knowledge Management Systems in *2008 International Astronautical Congress* Glasgow, Scotland, September 28 - October 3, 2008.

[21] Goel A. K. and Bhatta S. R. Use of design patterns in analogy-based design,. *Advanced Engineering Informatics,* 2004, vol. 18, pp. 85-94.

[22] Franke H.-J. Bilder und Begriffe beim konstruktiven Denken – Diskussionen und Ergebnisse eines Workshops in *Effizienter Entwickeln und Konstruieren* Munich: VDI Verlag, 1995.

[23] Dieter G. *Engineering Design: A Materials and Processing Approach,* Third Edition ed.1999

(McGraw-Hill.
[24] Malmqvist J. Improved Function-means Trees by Inclusion of Design History Information, . *Journal of Engineering Design,* 1997, vol. 8, p. 107 —117.
[25] http://www.omg.org/cgi-bin/apps/doc?formal/07-09-01.pdf, accessed June 2008.
[26] www.eclipse.org, accessed 10 January, 2009.
[27] gforge.enseeiht.fr/projects/topcased-sysml/, accessed 10 January, 2009.
[28] http://www.sfb588.uni-karlsruhe.de/, accessed 10 January, 2009.
[29] Bracewell R. H. and Sharpe J. E. E. Functional Description Used in Computer Support for Qualitative Scheme Generation - Schemebuilder. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 1996, vol. 10, pp. 333-346.
[30] O'Sullivan B. Interactive constraint-aided conceptual design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 2002, vol. 16, pp. 303-328.
[31] Yan X.-T. and Sawada H. A framework for supporting multidisciplinary engineering design exploration and life-cycle design using underconstrained problem solving. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 2006, vol. 20, pp. 329-350.
[32] Maher M. L., Balachandran M. B., and Zhang D. M. *Case-Based Reasoning in Design,* 1995 (LAwrence Erlbaum Associates Inc., Mahwah, New Jersey).
[33] Goel A. K., Bhatta S. R., and Stroulia E. Kritik: An Early Case-Based Design System. in *Issues and Applications of Case-Based Reasoning in Design*, M. L. Maher and P. Pu, Eds., 1997 (Erlbaum, Mahwah, NJ), 1997, pp. 87-132.
[34] Sudarsan R., Fenves S. J., Sriram R. D., and Wang F. A product information modeling framework for product lifecycle management. *Computer-Aided Design,* 2005, vol. 37, pp. 1399-1411.
[35] Fenves S. J., Foufou S., Bock C., and Sriram R. D., *CPM: A Core Product Model for product data*, Manufacturing Systems Integration Division, National Institute of Standards and Technology, Gaithersburg 2002.

Contact: Prof. Karthik Ramani
School of Mechanical Engineering, Purdue University
West Lafayette, IN 47907, USA
Postal Code, City
Phone + (765) 494 5659
URL: www.purdue.edu/precise

Karthik Ramani is a Professor in the School of Mechanical Engineering at Purdue University. He earned his B.Tech from the Indian Institute of Technology, Madras, in 1985, an MS from The Ohio State University, in 1987, and a Ph.D. from Stanford University in 1991, all in Mechanical Engineering. e serves in the editorial board of Elsevier Journal of Computer-Aided Design and ASME Journal of Mechanical Design. His interests are in digital and computational geometry, shape search, shape design and analysis, configuration, constraint representation and solving, and early design.

Srikanth Devanathan graduated in 2009 from Purdue University with a Ph.D. in mechanical engineering and currently works for Dessault Systemes Simulia Corp. in Cary, NC. Prior to his doctorate, Srikanth obtained his bachelors from Indian Institute of Technology, Madras in 2003. His interests include visual analytics in design, constraint processing and optimization.

Christian Sauter is Scientific Assistant at the Institute of Product Development at the University of Karlsruhe (TH) where he leads the robotics research team. He earned his Dipl -Ing from University of Karlsruhe (TH). His interests include design methodologies, design documentation and product realization.

Prof. Dr.-Ing. Dr. h.c. Albert Albers is Professor for Product Development and Director of the Institute of Product Development at the University of Karlsruhe (TH). His research is in the fields of development methodology and development management, CAE/optimization, drive technology and mechatronics.