Proceedings of ICAD2002
Second International Conference on Axiomatic Design
Cambridge, MA – June 10&11, 2002
ICAD 038

COMPUTER AIDED GEOMETRIC TOPOLOGY AND SHAPE DESIGN WITHIN AXIOMATIC DESIGN FRAMEWORK

Jinpyung Chung
Research Assistant
cjp@mit.edu

Nam P. Suh
Ralph E. & Eloise Cross Professor
npsuh@mit.edu

Department of Mechanical Engineering
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA. 02421, USA

ABSTRACT

This paper presents a computer-aided method for designing topologies and shapes of geometric artifacts from the highest-level functional requirements in conceptual design phase. The method combines a thinking process of engineering design and knowledge base within axiomatic design framework. The proposed thinking process is called a V-model in this paper. The V-model consists of three main sub-processes; top-down decomposition process of functional requirements(FRs) and design parameters(DPs), mapping process of DPs into geometric entities, and bottom-up integration process of the geometric entities. Knowledge base stores information on FRs, DPs, and corresponding geometric entities generated during the V-model design process in a unique structure designed to combine advantages of both top-down and bottom-up approaches. The design matrix is used to relate visually the effect of geometric entities to corresponding FRs. The method presented in this paper can be the basis for creating a new intelligent CAD system that incorporates the FRs of a design task. This method enhances the designers' creative thinking for geometric shape design and facilitates the reuse of CAD models by relating functions to geometric topology and shape design.

Keywords: CAD(computer aided design), axiomatic design, thinking process, knowledge base

1 INTRODUCTION

Mechanical design is an iterative, multi-level decision-making engineering process. It begins with an identification of customer needs, proceeds through a sequence of activities to seek a solution to the defined task through specification, synthesis, analysis, and evaluation, and ends with the detailed description and/or geometric models of the product. Design process may be divided into three stages. Stage 1 is a mapping from customer needs to functional requirements(FRs) where information on the product is collected and transformed into engineering specifications and goals. Stage 2 is the conceptual design phase, in which the primary concern is the generation of physical design embodiment to meet the FRs through the creation of design alternatives and selection of best design from the set. The designers at this stage must create rough sketches or solid models and pass them with description of the concepts to other designers to detail the concepts. The final stage is the detailed design phase. During this last phase, final decisions on dimensions, more detailed physical shapes of individual components and materials are made. Among these three stages, the conceptual design phase plays a very important role in the overall success of the design. Good design decisions made during the conceptual design phase reduce iterations between design stages and processes, and eventually lead to a satisfactory final product.

Current CAD technology successfully supports generation of geometric shapes and assemblies of the designed artifacts in detailed design stage using solid modeling techniques and specific data structures. However, current CAD programs are drawing packages, which cannot be used to support the conceptual design process. The main challenges in creating an intelligent CAD system that can deal with conceptual design are two folds. First, since commercially available CAD systems cannot reason and create geometric shapes based on the FRs of the design task, all the basic design decisions on geometric details must be made by human designers. Solid modeling techniques are useful only after detailed shapes of the designed artifacts have been determined. Most CAD programs do not use the information on FRs of the designed task. The lack of the functional representation of the CAD models slows down the design process in developing a new product, because designers do not understand why certain physical shapes are created and they must update the CAD models based only on the geometric shapes and their assemblies. Therefore, it is difficult to reuse existing design knowledge or concept using existing CAD models in developing new physical artifacts with computer support. Second, current CAD packages cannot automatically create physical artifacts from the leaf-level DPs. In the current CAD packages, the designed artifacts are decomposed mainly in the physical domain and only the information and knowledge of the physical decomposition are stored and dealt with. However, the designed artifacts must be functionally decomposed in the conceptual design stage. Thus, the information and knowledge of the functional decomposition should be related to the information and knowledge produced by the physical decomposition by using knowledge base. In addition, a way to deal with physical interfaces between physical entities in the hierarchies produced by the decompositions is needed.

In this paper, axiomatic design theory is used to propose a framework in implementing an intelligent CAD program to support conceptual design. Axiomatic design framework [1-3] has been created to incorporate functional aspects of the designed artifacts based on a natural thinking process for conceptual design. We formalize the axiomatic design theory to create a method for design of geometric topologies and shapes using computer. The following sections present the axiomatic design framework, the details of the proposed method, and an example to illustrate the method.

2. AXIOMATIC DESIGN FRAMEWORK

In axiomatic design, synthesized solutions that satisfy the highest-level FRs are created through a decomposition process that requires zigzagging between the functional domain and the physical domain as shown in figure 1. It decomposes a top-level FR into leaf level FRs, which are not decomposable any further. Designer creates leaf level DPs in his brain or extracts those from his knowledge base to satisfy the corresponding FRs. Once leaf level DPs are found, they must be integrated to create the whole design artifact, which is then checked to determine if they work well and satisfy FRs based on two design axioms. The first axiom is the independence axiom, which states that the independence of FRs must be maintained in design. Second axiom is information axiom, which states that the information content must be minimized.

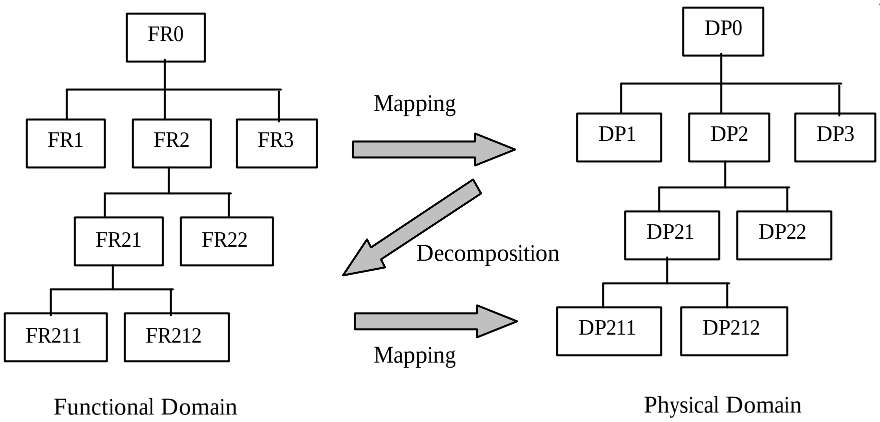


Figure 1. Zigzagging decomposition / mapping /composition process of axiomatic design

Mathematical representation of axiomatic design is

$$\begin{Bmatrix} FR1 \\ FR2 \\ FR3 \end{Bmatrix} = \begin{bmatrix} A11 & A12 & A13 \\ A21 & A22 & A23 \\ A31 & A32 & A33 \end{bmatrix} \begin{Bmatrix} DP1 \\ DP2 \\ DP3 \end{Bmatrix} \text{ where, } Aij = \frac{\partial FRi}{\partial DPj}$$

.Aij represents sensitivity, either qualitative or quantitative, of FRi with respect to DPj and it can be interpreted as a causality, which represents cause-effect between DPs and FRs. The elements Aij of the design matrix(DM) are determined in mapping process. Figure 2 shows coupling conditions represented by the design matrix.

$$\begin{bmatrix} X & O & O \\ O & X & O \\ O & O & X \end{bmatrix} \qquad \begin{bmatrix} X & O & O \\ X & X & O \\ X & X & X \end{bmatrix} \qquad \begin{bmatrix} X & X & X \\ X & X & X \\ X & X & X \end{bmatrix}$$

Uncoupled Decoupled Coupled

Figure 2. Coupling conditions represented by the design matrix

3 COMPUTER-AIDED METHOD FOR GEOMETRIC TOPOLOGY AND SHAPE DESIGN

3.1 DESIGN PROCESS

In artificial intelligence, top-down or bottom-up approach has been used to represent the human brain model. Top-down approach is represented as a process to decompose a big problem into small pieces from top to bottom and then to find out solutions of the problem. The decomposed concept or knowledge to solve a problem is pre-programmed and stored into a large knowledge base. Then, inference techniques based on deduction, induction and abduction are used to infer relevant concept or knowledge from given queries. Natural language processing[4] or CYC project[5] are the representative examples of the top-down approach. In contrast to the top-down approach, bottom-up approach begins with a relatively small number of physical building blocks. It interfaces the physical building blocks in the physical space, and finds out solutions by simulating their various combinations as alternatives to satisfy a certain goal. This approach has been used for robotics or artificial life[6, 7] using neural networks or genetic algorithms.

As a design process, the top-down approach requires a large knowledge base and extracts corresponding knowledge if the required design knowledge exists inside the knowledge base, but it has a small degree of freedom for creative design, because the traditional top-down approach lacks the re-composition process of combining extracted pieces of various design concept or knowledge. The bottom-up approach has more degrees of freedom to generate physical shapes or assemblies in the physical space that human designer have not considered. However, the combination of the building blocks is sometimes too large to determine the best solution. Thus, the bottom-up process has been applied to small ad hoc problems or confined to a limited number of physical building blocks.

A V-model, adapted from Do and Suh[8], is proposed as a thinking process for computer-aided design of geometric topologies and shapes. It combines the advantages of the top-down and the bottom-up approaches and consists of three sub-processes as shown in Figure 3: top-down decomposition process, mapping process, and bottom-up integration process. Top-down decomposition process is the zigzagging decomposition process used in axiomatic design process.[1-3] The decomposition process conceptually divides a big, complex problem into solvable small pieces and finds design solutions for the divided small problems. It produces language descriptions of decomposed FRs and DPs. A DP is a description of a proposed solution to satisfy the corresponding FR, and plays a role as a key design variable as a part of the whole design solution.

Mapping process is a process to create geometric entities based on the leaf level DPs produced by the top-down decomposition process. The geometric entities mapped to the leaf level DPs are key geometric objects that satisfy the corresponding leaf level FRs. They must be interfaced with each other and be integrated into a complete product in the physical domain to satisfy higher-level FRs through the bottom-up integration process. There are three steps in the bottom-up integration process. First step is to establish interfaces between the geometric entities created during the mapping process. The types of interfaces are classified into rigid attachment, mate, fit, roll, slide and so on. Second step is to construct topologies by integrating related geometric entities into complete solids. The construction of the topologies is determined by the locations of the geometric entities in the physical space. Third step is the determination of the final shapes.

Determination of the final shapes is done by controlling DPs to satisfy corresponding FRs and constraints within the constructed topologies. The design matrix, shown in Figure 2, is used to trace how each geometric entity affects FRs. An illustrative example about the use of the V-model for geometry design will follow in section 4.

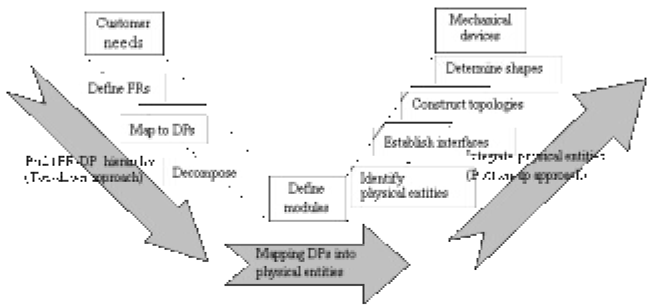


Figure 3. Axiomatic thinking process for designing physical products

3.2 KNOWLEDGE BASE

The V-model design process generates two types of design knowledge; one is knowledge about how FRs, DPs, and physical entities are related with each other, and the other is knowledge about how FRs and DPs are decomposed in depth. Figure 4 shows the knowledge base, which stores both types of design knowledge. It consists of databases expressed as:

fr-dp(“FR1”, [“DP1_A”, “DP1_B”, …..]) // fr-dp database

fr-decomposition(“A”, [FR0(FR1(FR11, FR12), //fr-decomposition database
FR2
FR3(FR31(FR311, FR312), FR32)])

dp-decomposition(“A”, [DP0(DP1(DP11, DP12), //dp-decomposition database
DP2
DP3(DP 31(DP 311, DP 312), DP 32)])

dp-geometry(“DP1_A”, “DP11_A”, “DP12_A”, slide) //dp-geometry database
dp-geometry*(“DP11_A”, x_translational_motion, [x], null, shape_representation)
dp-geometry*(“DP12_A”, z _rotational_motion, [theta], [x0, y0], shape_representation)

The fr-dp database stores information of mapping one FR into many DPs. It is used to search proper candidate DPs for a given FR. The fr-decomposition database stores information of hierarchies of decomposed FRs and the dp-decomposition database stores information of hierarchies of decomposed DPs. A character, “A” or “B”, in “DP1_A” and “DP1_B” is an index to the corresponding hierarchical tree. The dp-geometry database stores information of geometric entities mapped to corresponding DPs. Each leaf level geometric entity, denoted by *, is defined by motion type, location, reference of the motion, and representation of the shape. The motion types of a geometric entity in the physical space are classified into translational motion and rotational motion in x, y, and z direction. The location of a geometric entity is defined by generalized coordinates defined by x, y, z, and/or theta. The representation of a certain shape feature in 3D solids is defined by B-rep(boundary representation) and/or CSG(Constructive Solid Geometry).[9] Some DPs are defined only by interfaces of lower-level geometric entities. For example, dp-geometry for “DP1_A” is defined by the interface, “slide”, between two leaf-level DPs(DP11_A and DP12_A), instead of including all the elements used to define the two leaf level geometric entities.

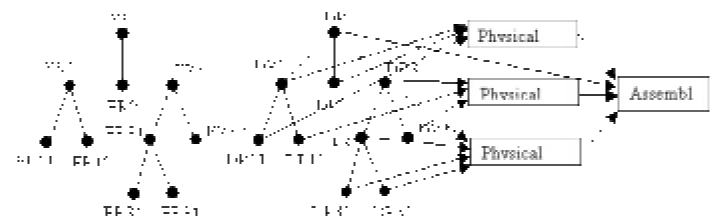


Figure 4. Fundamental structure of knowledge base

The language description of decomposed FRs and DPs is parsed as a set of grouped English words and then stored in the knowledge base. If a certain FR of hierarchy “B” has a similar meaning to an FR1 of hierarchy “A” already stored in the knowledge base during the language parsing process, the corresponding DP is stored in the fr-dp database as “DP1_B” in addition to “DP1_A”. This process can map one FR to many DPs in the database as suggested in the Thinking Design Machine[3]. A difference of the knowledge base proposed in this paper from the database of the Thinking Design Machine is a tree structure of the knowledge base and the hierarchical information flow. This hierarchical information flow may be efficient to infer design knowledge, although the inference was not dealt with in this paper.

Search of the proper DPs from given FRs is performed by a statistical approach. The statistical approach measures the distance between an FR already stored in the knowledge base and a query FR. The distance is calculated by the frequency of appearing words and correlations between words. Synonyms of a certain English word can be checked by WordNet[10]. Storing process of a certain FR will be performed by the same approach as that in the search process.

3.3 CAD SYSTEM ARCHITECTURE

Figure 5 shows the proposed CAD system architecture for implementing the method presented in this paper. The CAD system consists of several computer programs and databases to support the V-model design process. Axiomatic design agent is a main program to control all the information flows between a tool, agents, knowledge base and databases. Knowledge base stores information on the FRs, DPs, and geometric entities as explained in section 3.2. Language processing agent parses language descriptions of FRs and DPs into English words, place them in proper position and searches necessary information from given inputs. Geometric sketch tool receives designers’ input related to geometric entities and visualizes them on computer screen. Geometric integration agent helps designers to integrate geometric entities into complete shapes and assemblies automatically or interactively with designers. Mathematical modeling agent deals with mathematical equations, which represent FRs used in

engineering design problems. Solid model construction agent supports conversion process from the sketches to solid models.

3.3 CAD SYSTEM ARCHITECTURE

Figure 5 shows the proposed CAD system architecture for implementing the method presented in this paper. The CAD system consists of several computer programs and databases to support the V-model design process. Axiomatic design agent is a main program to control all the information flows between a tool, agents, knowledge base and databases. Knowledge base stores information on the FRs, DPs, and geometric entities as explained in section 3.2. Language processing agent parses language descriptions of FRs and DPs into English words, place them in proper position and searches necessary information from given inputs. Geometric sketch tool receives designers' input related to geometric entities and visualizes them on computer screen. Geometric integration agent helps designers to integrate geometric entities into complete shapes and assemblies automatically or interactively with designers. Mathematical modeling agent deals with mathematical equations, which represent FRs used in engineering design problems. Solid model construction agent supports conversion process from the sketches to solid models.

4. AN ILLUSTRATIVE EXAMPLE

This example shows how V-model is applied to a simple cam-pawl mechanism used for a parking mode in automatic transmission of automobiles. The created design knowledge is stored in the knowledge base to be reused later.

4.1 V-MODEL

Step 1: FR-DP decomposition

In this step, FR-DP hierarchies are produced by the zigzagging decomposition process. FRs are defined as "what we want to achieve" in functional domain and DPs are defined as "how we want to achieve it" in physical domain. Language description is used to explicitly represent the meaning of FRs and DPs. The real physical shapes of solid parts and their assemblies might be in the designer's mind, but have not been explicitly represented in this step. The followings are the decomposed FRs and DPs from top to bottom.

FR0 = Prevent an attended parked vehicle from moving on its own
- FR1 = Engage the pawl to the engaged position
  - FR11 = Push the cam
  - FR12 = Push up the pawl to the engaged position
- FR2 = Keep the pawl in the engaged position
- FR3 = Disengage the pawl from the engaged position
  - FR31 = Pull out the cam
  - FR32 = Pull down the pawl from its engaged position
- FR4 = Carry the weight of the vehicle in the engaged position
  - FR41 = Carry the force to the pin
  - FR42 = Endure the force at the pin

DP0 = Assembly of cam and pawl mechanism
- DP1 = Engagement mechanism
  - DP11 = Pushing force
  - DP12 = Tapered section of cam
- DP2 = Flat surface of cam
- DP3 = Disengagement mechanism
  - DP31= Pulling force
  - DP32 = Tension spring
- DP4 = Force carrying mechanism
  - DP41 = Vertical surface of the pawl
  - DP42 = Pin

<u>Step 2: Mapping DPs into geometric entities at the leaf-level</u>

One of the most important features for axiomatic design theory is the design matrix. The design matrix is a relational basis for controlling DPs to satisfy FRs based on the relationship between FRs, DPs, and geometric entities. The couplings between FRs of each design alternative are checked by the design matrix. In this step, only diagonal elements are checked as shown in the Figure 6.

A designer creates key geometric entities using the geometry sketch tool and links them to corresponding language descriptions of the DPs at the leaf level. Each geometric entity is defined by its location, orientation, and shape parameters in the physical space. Dp-geometry database stores information on the definitions of geometric entities as briefly explained in section 3.2. Table 1 shows the created geometric entities at the leaf level.

<u>Step 3: Establish interfaces</u>

All interfaces between geometric entities must be explicitly defined and represented during this step. A process of defining the interfaces using the geometry sketch tool makes the designer to represent all the geometric entities and their interfaces explicitly in a computational form. In this example, information on the pawl, the designer considered in the top-down decomposition process, but not shown as any DP, is explicitly represented and visualized during this step. Table 2 shows all the interfaces between geometric objects that include all the leaf level DPs. As shown in the table, one level higher DPs are constructed in this step.

<u>Step 4: Construct topologies</u>

The created and interfaced geometric entities through the previous steps are integrated into certain topologies in this "construct topologies" step. Three procedures are needed to complete this step:

1. Collecting related geometric entities that consist of complete solid parts based on their names or language descriptions
2. Locating the geometric entities in the physical space
3. Skinning by making boundaries between geometric entities and then filling materials inside the closed boundaries

Topology construction is crucial for the performance of the final product and is closely related to creativity. Generally, designers use sketches to create various topologies. However, topology construction procedure in the physical space is a very tedious even with a small number of geometric entities. Therefore, a computer support is needed for constructing various topologies with only key geometric entities. Figure 7 shows possible topologies for DP0 integrated from leaf level DPs.

<u>Step 5: Determine shapes to satisfy given FRs</u>

The best shape for each alternative design generated through the step 4 is determined by controlling DPs to satisfy the corresponding FRs. The FRs, described by language in step 1, can be represented by using mathematical equations to calculate system ranges, which satisfy given design ranges as shown in several engineering design problems of [1, 2]. In this example, six mathematical equations have been made for alternative 1 to calculate system ranges. Figure 8 shows the links of FRs to the mathematical equations for each module("SR" stands for "System Range"), and the links of DPs to geometric entities created in the physical space on the computer screen. SR1 is the difference between the pushing force(DP11) and the friction force generated from an interface between the pawl and the tapered section of cam(DP12). Thus, SR1 should be greater than zero to engage the pawl to the engaged position. SR2 is a safety factor to keep the cam at the engaged position from any disturbances. SR31 is the difference between the pulling force(DP31) and the friction force generated from an interface between the pawl and the flat surface of the cam(DP2). SR32 is a sum of vertical forces in downward direction to disengage the pawl from the engaged position after pulling out the cam. SR41 is a force applied to the pin carried by the vertical surface of the pawl(DP41) from the given force generated by the weight of the vehicle. SR42 is a maximum stress produced in the pin(DP42). The system ranges are calculated based on dimensions shown in Figure 9 based on assuming quasi-static equilibriums. Once design ranges are given, each geometric entity can be controlled by the corresponding DP to move the system range inside the design range. As shown in Figure 8, SR32 and SR41 are coupled. Thus, change of either DP32 or DP41 will affect both SR32 and SR41, requiring iterations to satisfy given design ranges for FR32 and FR41. The procedure to satisfy all the given design ranges determines the best shape to satisfy the FRs of the design task. The determined shapes of the product on the sketch tool are generated in solid models for further analysis in detailed design stage. Figure 8 shows one possible shape for alternative 1 by given arbitrary design ranges.

4.2 A USAGE OF KNOWLEDGE BASE

Information on the hierarchies of FRs and DPs, and corresponding geometric entities of the created cam and pawl mechanism is stored in the knowledge base as described in section 3.2. Figure 8 shows that how DPs are linked to geometric entities at each level, and the links can be stored into dp-geometry database as a part of knowledge base. The links could help designers to search proper geometric entities from given language descriptions of query FRs. Once proper geometric entities are searched, the integration agent helps designers to integrate the geometric entities automatically or interactively with designers. Let's assume that a designer does not want FR1, "engage the pawl to the engaged position," for the cam and pawl mechanism. Thus, he/she inputs query FRs except FR1 such as

FR2 = Keep the pawl in the engaged position
FR3 = Disengage the pawl from the engaged position
FR4 = Carry the weight of the vehicle in the engaged position.

The integrated shape based on input FR2, FR3, and FR4 does not include geometric entities, which correspond to DP1. Thus, the corresponding DP11 and DP12 must be eliminated from the solid shape of the cam and the final shape for alternative 1 would be that shown in Figure 10.

5. CONCLUSIONS AND FUTURE WORKS

Generation of possible alternatives that have various geometric topologies and shapes, and selection of one alternative among them are not an easy task in conceptual design stage. This paper presented a systematic way to generate geometric topologies and shapes and to determine the best shape to satisfy functional requirements using computer aid within axiomatic design framework.

The V-model design process is an important basis for the systematic approach for designing geometry. It decomposes a design problem into small pieces, creates geometric entities at the leaf level, and integrates them in the physical space. The illustrative example shows that all the geometric entities can be created to satisfy the leaf level FRs and be integrated from the leaf level into the higher level DPs to satisfy the corresponding FRs. The integration process notably begins from defining interfaces between geometric entities, and then constructs solid shapes and assemblies simultaneously in the same step, which is different from solid modeling process used in the existing CAD programs. The existing CAD programs generate solid shapes first and then interface them into an assembly without explicitly representing functional aspects of the generated shapes. The difference of the two approaches comes from the different decomposition strategies: the most CAD programs decomposes a design problem only in the physical domain, but the V-model starts to decompose a design problem in the functional domain and finds solutions in the physical domain.

The V-model design process can produce hierarchies of geometric entities based on the hierarchies of FRs. All the information on FRs, DPs, and geometric entities can be extracted during the V-model design process and stored in the databases. Knowledge base properly relates the information with each other to construct design knowledge. This approach makes possible to manipulate and to use the geometric entities based on given FRs.

The design matrix is used to deal with all the relationships between FRs and geometric entities. It is a basis for reasoning on couplings between FRs and on the determination of the geometric topology and shape by informing how the change of geometric topology and shape affects the FRs.

The incorporation of the functional aspects of the CAD models in a systematic way based on a thinking process – the essence of the V-model – in conceptual design stage is an important step in computer aided design. A CAD system implemented by the method presented in this paper will enhance understandability, adaptability, and reusability of design concepts related to the CAD models, and increase degrees of freedom for creative design. In the future, a metric for language processing and a geometric feature modeling technique to support the V-model design process will be developed to implement the proposed CAD system described in section 3.3.

The V-model proposed in this paper can be used as the basis for creating intelligent CAD systems in the future.

6. REFERENCES

[1] N. P. Suh, "Principles of Design," Oxford University Press, 1990
[2] N. P. Suh, "Axiomatic Design: Advances and Applications," Oxford University Press, 2000
[3] N. P. Suh and S. Sekimoto, "Design of Thinking Design Machine," CIRP Annals, 1990
[4] E. Charniak and D. McDermott, "Introduction to Artificial Intelligence," Addison-Wesley Publishing Company, 1985
[5] D. B. Lenat, "The Dimension of Context Space," Cyc Corporation, 1998

[6] R. A. Brooks, C. Breazeal, R. Irie, C. C. Kemp, M. Marjanovi_c, B. Scassellati, M. M. Williamson, "The Cog Project: Building a Humanoid Robot," 1998

[7] Lipson, H., Pollack J. B., 2000, "Automatic Design and Manufacture of Artificial Lifeforms," Nature 406, pp. 974-978

[8] S. Do and N. P. Suh, "Object-Oriented Software Design with Axiomatic Design," Proceedings of International Conference on Axiomatic Design, 2000

[9] H. Toriya and H. Chiyokura, "3D CAD: Principles and Applications," Springer-Verlag, 1991

[10] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "Introduction to WordNet : An On-line Lexical Database," http://www.cogsci.princeton.edu/~wn, August, 1993

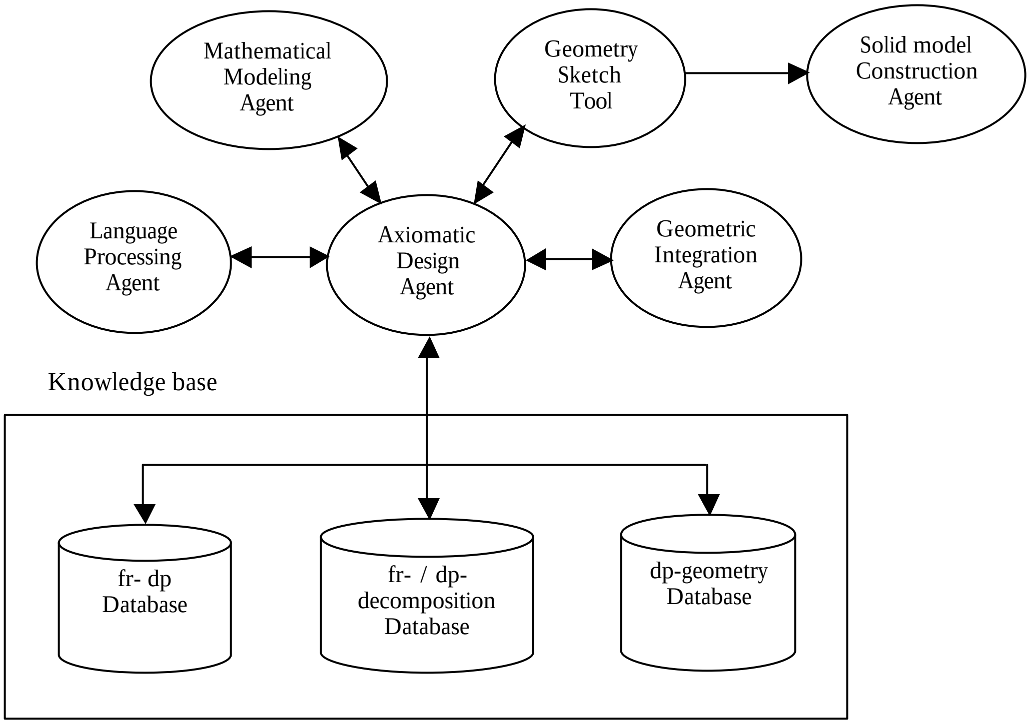


Figure 5. Proposed CAD system architecture

		DP0						
		DP1		DP2	DP3		DP4	
		DP1 1	DP1 2		DP3 1	DP3 2	DP4 1	DP4 2
FR1	FR11	X						
	FR12		X					
FR2				X				
FR3	FR31				X			
	FR32					X		
FR4	FR41						X	
	FR42							X

Figure 6. Full design matrix

DP11	F_{push} →	DP12	Cam1 - face1, $\theta 1$
DP2	Cam1 – face2		
DP31	F_{pull} ←	DP32	Spring1
DP41	$\theta 2$, Pawl1 - face1	DP42	Pawl1 - pin1

Table 1. Geometric entities mapped from leaf level DPs

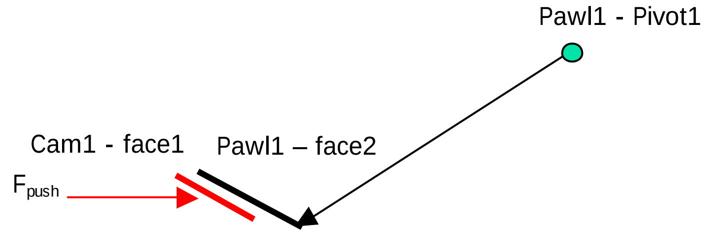
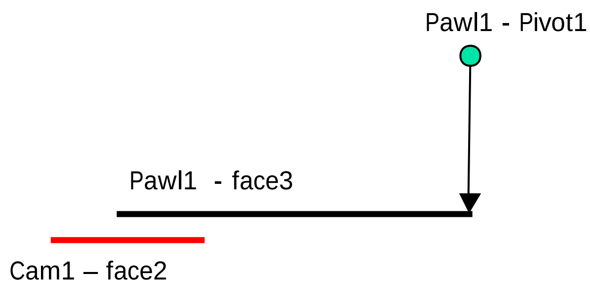
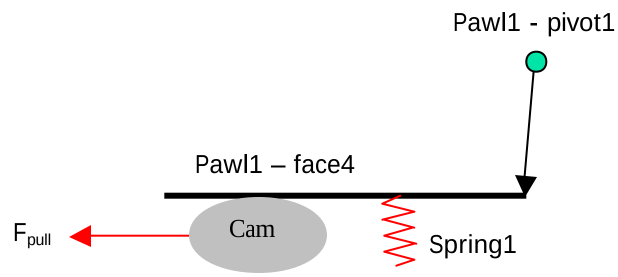
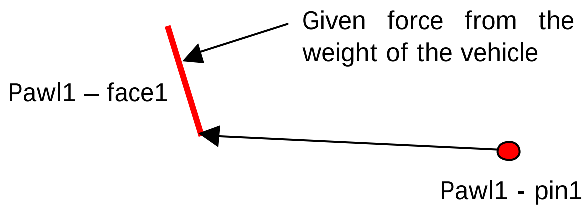
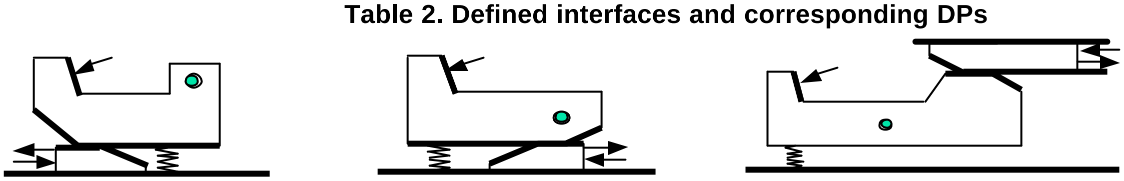
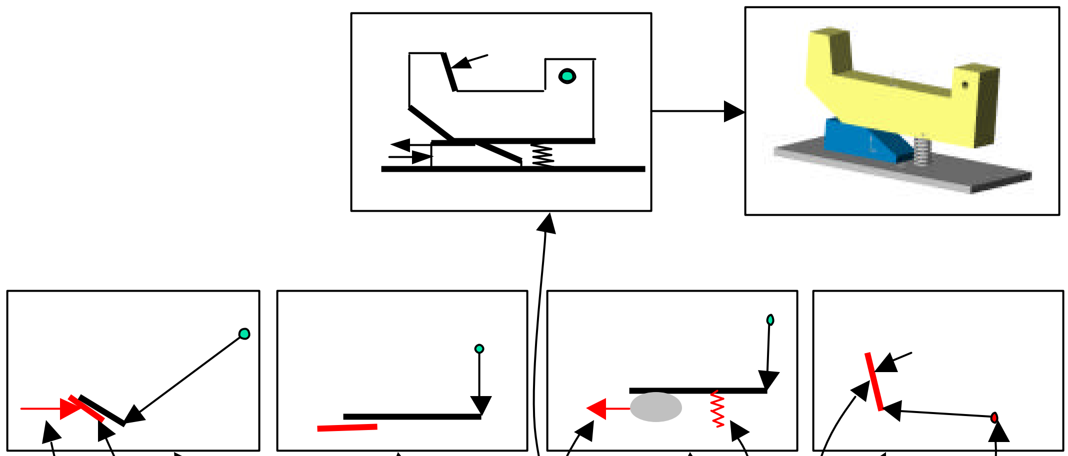
<table>
<tr><td>DP1</td><td>
</td></tr>
<tr><td>DP2</td><td>
</td></tr>
<tr><td>DP3</td><td>
</td></tr>
<tr><td>DP4</td><td>
</td></tr>
</table>

Table 2. Defined interfaces and corresponding DPs



Alternative 1 Alternative 2 Alternative 3

Figure 7. Possible alternatives based on topology construction



		DP0						
		DP1		DP2	DP3		DP4	
		DP11	DP12		DP31	DP32	DP41	DP42
FR1	FR11	X						
	FR12	X	X			X		
FR2				X				
FR3	FR31			X	X	X	X	
	FR32					X	X	
FR4	FR41			X		X	X	
	FR42			X		X	X	X

$$\text{SR1} = F_{push} - \frac{[k_2 x2_1 (L_3 \cos q' - h_3 \sin q')][(m_1 + m_2)\cos q_{cam} + (1 - m_1 m_2)\sin q_{cam}]}{h_1(\sin(q_{cam} + q') + m_1 \cos(q_{cam} + q')) - L_1(1 - m_1)\cos(q_{cam} + q')}$$

$$\text{SR2} = \; = d_3 - (x_{\text{disturbance}} + x_{\text{pawl}})$$

$$\text{SR31} = F_{pull} - \frac{m_1 + m_2}{L_5 + m_1 h_3}[L_3 k_2 x2 + L_4 F_{pawl} \sin q + (h_4 - h_3) F_{pawl} \cos q]$$

$$\text{SR32} = k_2 x2 + F_{pawl} \cos q \left(\frac{h_4 - h_3 - m_3 L_4}{L_3}\right) + F_{pawl} \sin q \left(\frac{m_3 (h_4 - h_3) + L_4}{L_3}\right)$$

$$\text{SR41} = F_{pin} = \sqrt{(F_{pawl} \cos q)^2 + [k_2 x2 (1 - \frac{L_3}{L_5}) - F_{pawl} \cos q (\frac{h_4 - h_3}{L_5}) - F_{pawl} \sin q (\frac{L_4}{L_5} - 1)]^2}$$

$$\text{SR42} = \frac{8 w_{pawl}}{p d_{pin}^{\;3}} F_{pin}$$

Figure 8. Links of FRs to system ranges and links of DPs to geometric entities on the design matrix

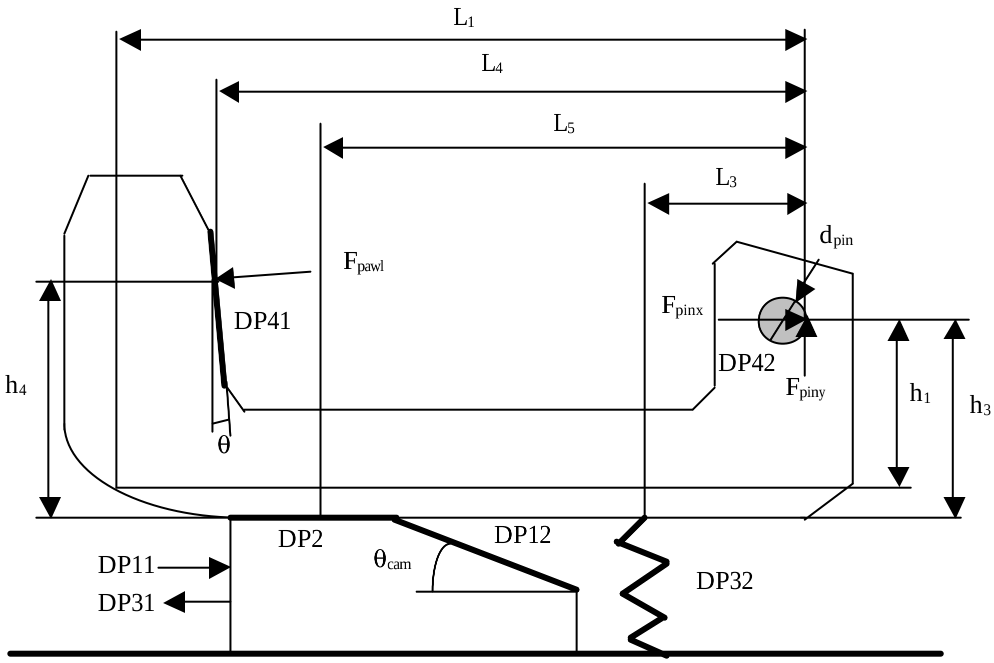


Figure 9. DPs and dimensions of alternative 1

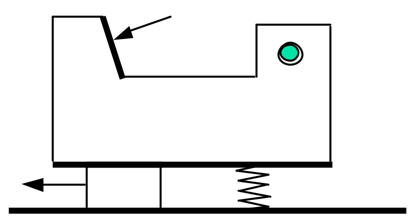


Figure 10. Generated shape without FR1