

# INVESTIGATING THE USE OF CONTROLLED NATURAL LANGUAGE AS PROBLEM DEFINITION INPUT FOR COMPUTER-AIDED DESIGN

**Hyunmin Cheong**

Dept. of Mech. & Ind. Eng.  
University of Toronto  
cheong@mie.utoronto.ca

**Wei Li**

Autodesk Research  
wei.li@autodesk.com

**L.H. Shu**

Dept. of Mech. & Ind. Eng.  
University of Toronto  
shu@mie.utoronto.ca

**Erin Bradner**

Autodesk Research  
erin.bradner@autodesk.com

**Francesco Iorio**

Autodesk Research  
francesco.iorio@autodesk.com

## ABSTRACT

We intend to develop a computer-aided design (CAD) system that takes design problem definitions as input and presents a set of geometries that solves the problem. For such system, we perceive using a controlled natural language (CNL) as one method within a multimodal interface to capture some of the required input. To evaluate the feasibility of using a CNL for problem definitions, we conducted a user study with 18 participants. We found that using a CNL increases the quality of problem definition statements for functional requirements compared to statements written in natural language. While a CNL limits the breadth of problem definitions, it can achieve a balance between natural expression and formal specification of problem definitions.

## KEYWORDS

Computer-aided Design, Natural Language Input, Controlled Natural Language, Problem Definition

## 1. INTRODUCTION

Commercial computer-aided design (CAD) systems mainly support modelling, simulation, and analysis of design solutions. However, they lack capabilities to understand design problems and to generate corresponding solutions. To address this challenge, we envision a new CAD system that can create multiple design spaces based on problem descriptions and explore sets of optimal solutions within those spaces. The system would solve constrained optimization problems formulated based on the design goals specified by a designer, e.g., optimize a certain property of the design while preserving design constraints and parameters.

To enable the envisioned system, a designer's definition of design problems must be taken as input. The key part of problem definitions involves identifying the desired functions of a design (Pahl and Beitz, 1996). In addition, because our problem solving approach is rooted on optimization, we must

capture constraints, which define feasible solutions, and objectives, which are used to judge how well solutions solve the problem (McCahan, 2013). The definition can also include descriptions of the design environment (Zeng, 2011), e.g., mechanical interactions and spatial configuration.

We are considering multiple user interfaces for capturing problem definitions. Grossman et al. (2012) found that traditional menu-toolbar-icon centric graphical user interfaces (GUI) often require designers to spend much of their time learning and managing the interface. Therefore, we are also investigating more natural interfaces such as gesture (Wang et al, 2011), sketch (Valentine et al, 2013), speech (Kou et al, 2010), or a combination of interfaces (Sharma et al, 2011; Nanjundaswamy et al, 2013). The main challenge is identifying which interface is appropriate for capturing different types of problem definition information.

For capturing high-level semantic information such as the functions, objectives, and constraints of a design, using a natural language interface could be ideal. Ullman (1992) observed that the initial goals of most mechanical design problems are expressed in a written or verbal language. However, using a natural language interface is challenging because of ambiguities and complexities involved in understanding a human-oriented language.

One approach in enabling natural language input is to use a controlled natural language (CNL). A CNL reduces ambiguities in a language by restricting its syntax and lexicons. Our eventual goal is to develop a CNL that can specify some of the problem definition information required by constrained optimization procedures. We aim to achieve a balance between the expressive power of the language and its computational complexity.

The main goal of the current paper is to examine the feasibility of using a CNL to express problem definition statements for design. We conducted a user study with engineering graduate students and asked them to write down problem definition statements in either natural language or a CNL. We also asked the participants who wrote statements in natural language to translate the statements into a CNL. We then evaluated the quality and quantity of problem definition statements written in natural language compared to those written in a CNL (either written originally or translated from natural language). Although the current paper focuses only on the CNL input, in the future we plan to integrate a CNL within a multimodal interface for our envisioned CAD system.

## 2. RELATED WORK

Few studies have investigated using natural language as problem definition input for CAD. Most research on natural language input for CAD has focused on enabling users to use a predefined set of voice commands to execute existing CAD operations (Kou et al, 2010; Sharma et al, 2011; Nanjundaswamy et al, 2013).

Peterson et al. (1994) created a system called KA that takes natural language descriptions of design problems as input and finds analogous solutions based on the *model-based design* problem-solving approach. They resolved the challenges of natural language understanding through interaction between its natural language interface and problem-solving module. For example, KA leaves some ambiguities of natural language input to be resolved by the problem-solving module.

Chen et al. (2007) also tackled the problem of understanding natural language input, particularly for analysing product requirements. The authors used lexical, syntactic, and structural analysis to

translate natural language descriptions in complete sentences into formal structure diagrams. Their formalized output could potentially support subsequent computational design procedures.

To reduce the computational complexity and ambiguities in processing natural language input, a controlled natural language (CNL) could be used. A CNL is a precisely defined subset of a natural language that restricts the syntax and lexicons. A CNL can be translated automatically into a formal target language that can be used for automated reasoning. In addition, users only need to learn the subset of syntax and lexicons, instead of an entirely new interface or input language for a CAD system.

Early examples of CNL include Cleopatra, a CNL interface introduced by Samad and Director (1985) for CAD commands input. CNL has also gained much attention as a high-level interface method to knowledge-based systems; see Schwitter (2010) and references therein. For automatic requirements analysis for software development, Requirements Analysis Tool by Sarkar et al. (2012) also uses a CNL to input software requirements.

## 3. METHODS

We conducted a user study to evaluate the feasibility of using a CNL as problem definitions for design.

### 3.1. PARTICIPANTS

Participants consisted of 18 MSc/MEng/PhD graduate students from the Mechanical and Industrial Engineering Department at the University of Toronto. All participants had experience working with multiple design projects. Eight participants had industry experience related to engineering design.

### 3.2. DESIGN PROBLEM

Participants were asked to write problem definition statements for the problem of designing a bike rack. The problem was chosen for its relative simplicity. Table-1 shows the instruction and design problem presented to participants.

**Table 1** – Instruction and design problem given to participants

<p>Please list problem definition statements for designing a lightweight bike rack for a car. Images of a typical car and a bicycle are shown below. List all relevant functional requirements: e.g., functions, objectives, and constraints of the bike rack, as well as some properties of interacting (environment) objects. Use a single, complete sentence to write each problem definition statement. Generate at least one statement for each category. Use the same object names [roof, trunk, bumper, bicycle] as labelled on the image.</p>	
---	---

### 3.3. CONTROLLED NATURAL LANGUAGE

We created a CNL that could be used to describe the design problem chosen for the study. We assumed

that a solution for the problem must support loads from other objects while anchoring itself on another. Thus, the CNL was designed to describe problem definition statements for a simple statics problem.

Another important criterion in creating the CNL was the expressive power necessary to formulate a constrained optimization problem, the requirement for our envisioned CAD system. This criterion framed the semantic categories of the CNL, and subsequently its syntax and lexicons.

### 3.3.1. Semantic Categories

By semantic categories, we mean the types of problem definition statements that describe different aspects of the design problem. The categories include functions, objectives, and constraints of the design, as well as properties of environment objects (Table-2). The definitions have been mostly adapted from a reference text (McCahan, 2013) for a first-year general engineering design course.

**Table 2** – Definitions and examples of each semantic category

Functions	<b>Definition:</b> Functions describe what the design must do. Functions should not describe how well the design should perform. <b>Example:</b> “The design must support the weight of the shelf.”
Objectives	<b>Definition:</b> Objectives are used to judge how well the design solves the problem. They should include evaluation criteria. <b>Example:</b> “The weight of the design must be minimized.”
Constraints	<b>Definition:</b> Constraints set absolute limits that the design should not violate. Limits should be defined in quantifiable measures. <b>Example:</b> “The width of the design cannot be greater than 3cm.”
Properties of environ. objects	<b>Definition:</b> Objects that will interact with the design may have properties that must be considered when designing the solution. <b>Example:</b> “The weight of the shelf is 15kg.”

### 3.3.2. Syntax

Table-3 shows the basic syntax created for the user study. We recognize that the current syntax is not expressive enough to describe all relevant information for statics problems. We focused on limiting the allowed syntax to remove any ambiguity and reduce the time required by participants to learn the syntax. This consideration was also taken in creating the lexicons.

### 3.3.3. Lexicons

Table-4 lists lexicons used for our user study. The Object set contains objects specific to the bike rack design problem. The Function set is selected from the primary, secondary, and tertiary function terms from the functional basis (Hirtz et al, 2002). The lead author selected only the terms that are deemed relevant to statics problems. The *ObjectiveFunc* set is created based on the assumption that optimization

involves maximizing or minimizing a certain property. Lexicons for *MechQuant*, *GeomQuant*, and *MathOperator* were created based on the authors’ knowledge of statics.

**Table 3** – Syntax used for the user study

Functions	- The design should <i>Function</i> ( <i>MechQuant</i> ) ( <i>preposition</i>  of) <i>Object</i> - The design should <i>Function</i> ( <i>MechQuant</i> ) ( <i>preposition</i>  of) <i>Object</i> ( <i>preposition</i> ) <i>Object</i> - The design should <i>Function</i> <i>Object</i> ( <i>NumValue</i> ) ( <i>preposition</i> ) <i>Object</i>
Objectives	- <i>MechQuant</i> of the design must be <i>ObjectiveFunc</i> - <i>GeomQuant</i> of the design must be <i>ObjectiveFunc</i>
Constraints	- <i>MechQuant</i> of the design must be <i>MathOperator</i> <i>NumValue</i> - <i>GeomQuant</i> of the design must be <i>MathOperator</i> <i>NumValue</i> - <i>MechQuant</i> of the design must be <i>MathOperator</i> <i>MechQuant</i> of <i>Object</i> - <i>GeomQuant</i> of the design must be <i>MathOperator</i> <i>GeomQuant</i> of <i>Design</i>   <i>Object</i>
Environ. objects:	- <i>MechQuant</i> of <i>Object</i> is <i>NumValue</i> - <i>GeomQuant</i> of <i>Object</i> is <i>NumValue</i>

**Table 4** – Lexicons used for the user study

<i>Object</i>	bicycle, trunk, roof, bumper
<i>Function</i>	separate, divide, distribute, transfer, transmit, guide, translate, connect, couple, join, link, stop, prevent, inhibit, store, contain, collect, supply, support, stabilize, secure, position
<i>ObjectiveFunc</i>	minimize, maximize
<i>MechQuant</i>	force, load, pressure, torque, weight, strength
<i>GeomQuant</i>	length, width, height, thickness, depth, area, volume, clearance, interference, distance
<i>MathOperator</i>	greater than, smaller than, equal to, different from
<i>NumValue</i>	any numerical number + unit

## 3.4. EXPERIMENTAL CONDITIONS

Each participant was randomly assigned to one of the two conditions described below. A between-subjects design was used (Table-5).

**Table 5** – Experimental design. The colour codes/shades are used in our figures for convenient reference.

CNL-only condition	Task: Write statements in CNL (20 min)	
	↑ Compare ↓	
NL-CNL condition:	Task 1: Write statements in NL (20 min)	Task 2: Translate NL statements to CNL (20 min)
	← Compare →	

**CNL-only condition:** Participants in this condition were asked to use the syntax and lexicons (Tables-3 and -4), given at the outset of the study session, to generate problem definition statements. Twenty minutes were given to complete the task.

**NL-CNL condition:** Participants in this condition performed two tasks. They were first asked to generate problem definition statements on their own, without any syntax or lexicons provided. Twenty minutes were given for the first task. In the second task, participants were asked to translate, if they

could, their statements into the CNL based on the syntax and lexicons provided (Tables-3 and -4). If they wanted, they could also generate additional statements using the syntax and lexicons. Twenty minutes were also given for the second task.

Both conditions were asked to write at least one statement in each semantic category shown in Table-2. For clarification, all participants received the definitions of semantic categories.

Our design of the experiment allows two different comparisons of data. First, we can compare between conditions; statements generated in the first task of the NL-CNL condition vs. the CNL-only condition. This comparison evaluates the effect of using a CNL in writing original problem definition statements. We can also compare statements generated between the first task and the second task in the NL-CNL condition. This comparison evaluates the feasibility of translating natural language problem definition statements into a CNL.

### 3.5. EVALUATION

Number of statements: The number of statements generated was tallied for each task to measure the fluency of the problem definition process.

Quality of statements: The criteria used for quality ratings were the definitions of semantic categories in Table-2, based on our need to formulate constrained optimization problems. We recruited two independent raters, teaching assistants of a first-year general engineering design course at the University of Toronto, to evaluate the quality of statements based on the definitions. The raters had been trained to evaluate design reports containing problem statements based on the same reference text (McCahan, 2013) used to create the definitions in Table-2. The raters did not know any information about the syntax or lexicons that participants used. The ratings were given in a Likert scale from 1 (low quality) to 7 (high quality). We used the average of the two raters' ratings for the analysis ( $ICC(3,2)=.709$ ,  $F=3.44$ ,  $p<.001$ ).

Easiness/usefulness of the CNL: These measures examine how participants' perceived easiness or usefulness of using the syntax and lexicons differs between: 1) writing original problem definition statements in the CNL vs. 2) translating problem definition statements first written in natural language into the CNL. The following questions were asked after the experimental tasks:

- 1) On a scale of 1 to 7 (1=difficult, 7=easy), how would you rate your experience of using the CNL to write problem definition statements?
- 2) On a scale of 1 to 7 (1=difficult, 7=easy), how would you rate the usefulness of using the CNL to write problem definition statements?

Percentage of NL statements translated to CNL: For Task 2 of the NL-CNL condition, participants

could not translate some of their natural language statements into the CNL. The percentage of those instances likely indicates the limitations of the CNL.

Statements generated for the "properties of environment objects" category were excluded from our analysis. We observed that most participants from the NL-CNL condition misunderstood the category as describing the environment in which the design will operate, e.g., road conditions, weather. Because the statements generated between the two conditions varied significantly, we focused our analysis on other semantic categories.

### 3.6. EXPERIMENTAL HYPOTHESES

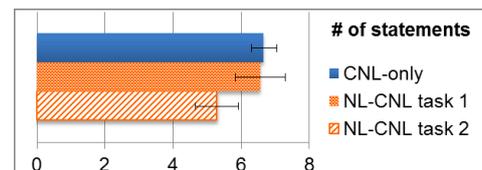
We hypothesized that using a CNL increases the quality of problem definition statements while decreasing the number of statements because of its limited expressiveness.

## 4. RESULTS

### 4.1. NUMBER OF STATEMENTS

Figure 1 shows that participants generated a similar number of statements in the CNL vs. natural language,  $t(16)=.145$ ,  $p=.886$ .<sup>1</sup> The results suggest that using a CNL and natural language resulted in similar fluency in writing problem definition statements.

However, the number of statements tends to decrease when translating from natural language into a CNL,  $t(16)=1.52$ ,  $p=.149$ .<sup>1</sup> This indicates that participants often could not express their original statements in the CNL.



**Figure 1** – Comparison of the number of problem definition statements generated. Error bars indicate standard error.

### 4.2. QUALITY

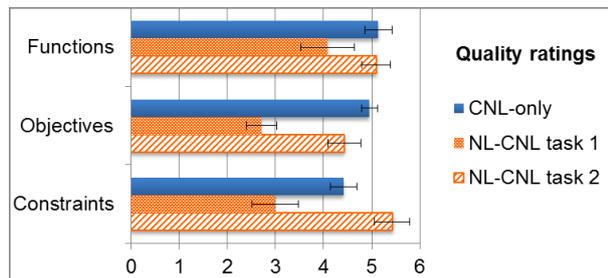
Figure-2 compares the quality of statements for each semantic category. For functions, the differences in quality were not statistically significant: CNL-only vs. NL-CNL task 1,  $t(16)=2.04$ ,  $p=.058$  and NL-CNL task 1 vs. task 2,  $t(16)=-2.00$ ,  $p=.133$ .<sup>2</sup> These results indicate that participants, all trained in engineering, could have already learned consistent ways similar to our CNL to define design functions.

We found that the quality of objective statements was significantly higher if they are written in the CNL compared to natural language: CNL-only vs.

<sup>1</sup> CNL-only:  $M=6.67$ ,  $SD=1.00$ ; NL-CNL task 1:  $M=6.56$ ,  $SD=2.07$ ; NL-CNL task 2:  $M=5.22$ ,  $SD=1.64$

<sup>2</sup> CNL-only:  $M=5.14$ ,  $SD=.72$ ; NL-CNL task 1:  $M=4.08$ ,  $SD=1.37$ ; NL-CNL task 2:  $M=5.20$ ,  $SD=.95$

NL-CNL task 1,  $t(16)=6.76$ ,  $p=.000$ .<sup>3</sup> Translating objective statements written in natural language into the CNL also significantly increases the quality: NL-CNL task 1 vs. task 2,  $t(16)=-4.34$ ,  $p=.001$ .<sup>3</sup>



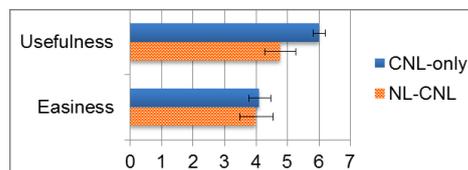
**Figure 2** – Comparison of quality ratings (rated out of 7) on the statements. Error bars indicate standard error.

Similarly, statistically significant differences are observed in the quality of constraint statements: CNL-only vs. NL-CNL task 1,  $t(16)=2.47$ ,  $p=.025$  and NL-CNL task 1 vs. task 2,  $t(16)=-3.02$ ,  $p=.008$ .<sup>4</sup>

Overall, the results confirm that using a CNL increases the quality of problem definitions.

#### 4.3. EASINESS AND USEFULNESS

Figure-3 shows that participants tend to assign high and medium scores for the usefulness and easiness of using the CNL, respectively. It is particularly encouraging to see high usefulness scores for the CNL-only condition.



**Figure 3** – Comparison of perceived usefulness and easiness (rated out of 7) of the CNL. Error bars indicate standard error.

The perceived usefulness was higher if the CNL was used right from the beginning, rather than used to translate statements already written in natural language,  $t(16)=2.48$ ,  $p=.025$ .<sup>5</sup> Participants of the NL-CNL condition could not express some of their original statements written in natural language. In addition, the NL-CNL condition participants likely evaluated the usefulness of the CNL relative to writing statements in natural language.

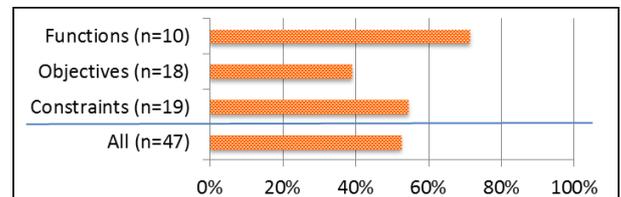
The perceived easiness did not vary much between the two cases,  $t(16)=.189$ ,  $p=.852$ .<sup>6</sup>

#### 4.4. TRANSLATION FROM NL TO CNL

Figure-4 shows the percentage of statements written in natural language that the NL-CNL condition participants could translate into CNL statements.

Overall, 52.5% of statements could be translated, indicating the limited breadth on the types of problem definitions that the CNL can express.

Participants were able to translate 71.4% of function statements, 39.1% of objective statements, and 54.5% of constraint statements. Again, function statements written in natural language did not seem to differ much from statements written in the CNL.



**Figure 4** – Percentage of NL statements for each semantic category translated into CNL statements (NL-CNL condition)

## 5. DISCUSSION

Our study showed that a simplified CNL guided participants to specify problem definitions in appropriate semantic categories created for our envisioned CAD system. Without the CNL, many participants in the NL-CNL condition seem to have mistaken the purpose of defining objectives and constraints. Instead, the participants described non-functional requirements such as affordances, usability, safety, etc. as objectives or constraints. By using the CNL, participants were able to frame the problem as a constrained optimization problem focusing on functional requirements.

The CNL also enabled participants to identify specific quantities that could be used to judge or limit solutions. For example, participants without the CNL tend to write statements such as “The design needs to be not too bulky” or “The design must not block the rear view.” While these are valid design considerations, more precise and quantifiable measures are preferred in problem definitions (Ullman, 1992; Pahl and Beitz, 1996; McCahan 2013). Using the CNL, participants could write more specific definitions such as “Volume of the design must be less than [...]”

In addition, some participants expressed that the lexicons helped them identify new functional requirements. For instance, the terms listed under “MechQuant,” e.g., “torque,” “pressure,” and “weight,” stimulated participants to consider different mechanical loads in problem definitions. This observation may explain why using the CNL did not compromise fluency.

However, a CNL can limit the breadth of information considered in problem definitions. We observed that about 50% of the statements written in natural language could not be translated into the CNL. Most of them described the non-functional design features as mentioned earlier. Subsequently, participants of the NL-CNL condition perceived the

<sup>3</sup> CNL-only:  $M=4.95$ ,  $SD=.44$ ; NL-CNL task 1:  $M=2.71$ ,  $SD=.89$ ; NL-CNL task 2:  $M=4.56$ ,  $SD=.91$

<sup>4</sup> CNL-only:  $M=4.42$ ,  $SD=.75$ ; NL-CNL task 1:  $M=2.99$ ,  $SD=1.56$ ; NL-CNL task 2:  $M=4.97$ ,  $SD=1.21$

<sup>5</sup> CNL-only:  $M=6.00$ ,  $SD=.50$ ; NL-CNL:  $M=4.78$ ,  $SD=1.39$

<sup>6</sup> CNL-only:  $M=4.11$ ,  $SD=.93$ ; NL-CNL:  $M=4.00$ ,  $SD=1.50$

syntax and lexicons as less useful because they could not express some of their original natural language statements. While we could add more syntax and lexicons to increase the expressiveness, this approach can quickly become impractical. Attempting to capture all types of problem definitions for different problems, contexts, and perspectives would be a significant challenge.

We emphasize that the purpose of our CNL is not to capture the entire knowledge of designers during the design process. Instead, a CNL would be one of different input methods used to funnel some of the knowledge into formal data that can leverage the enormous potential benefits of computational design. We envision that our system would have capabilities to formulate multiple design spaces based on the input, which is an essential process of creative design (Gero, 1994). To enable such capabilities, we must take high-level problem definitions as the input, not solution-oriented information used by existing CAD systems. The current study demonstrated the feasibility of a CNL in capturing high-level problem definitions.

## 6. CONCLUSIONS AND FUTURE WORK

We demonstrated that using a CNL, compared to using natural language, can aid designers to produce problem definition statements that entail more useful information for formulating constrained optimization problems. However, using a CNL can limit the breadth of information considered in problem definitions. Hence, a CNL should be devised carefully so that it does not restrict much of the freedom and creativity that designers want in their design activities and solutions.

Our future work will focus on understanding the natural language input and translating the input into formal data. We will create a formalized knowledge base of design lexicons, such as functions, shapes, materials, etc. We will also explore application of natural language understanding techniques, such as syntactic analysis and reference resolution, to handle more variety and natural expression of problem definition statements. This approach is likely more scalable than manually expanding the CNL.

In terms of future user studies, we could examine any longitudinal effect of receiving training to use a CNL. We also plan to evaluate the effectiveness of the CNL, either in comparison to or in combination with, input methods other than natural language on expressing different types of problem definitions. We must validate that the overall user experience is in fact enhanced through the use of a CNL.

## 7. ACKNOWLEDGMENTS

We thank the Natural Sciences and Engineering Research Council of Canada for financial support.

## REFERENCES

- Chen ZY, Yao S, Lin JQ, Zeng Y, and Eberlein A, "Formalisation of Product Requirements: from Natural Language Descriptions to Formal Specifications", *Int. J. Manuf. Res.*, Vol. 2, No. 3, 2007, pp 362-387
- Gero JS, "Towards a Model of Exploration in Computer-Aided Design", In: JS Gero and E Tyugu (eds), *Formal Design Methods for Computer-Aided Design*, North-Holland, Amsterdam, 1994, pp 271-291
- Grossman T, Fitzmaurice G, and Attar R, "A Survey of Software Learnability: Metrics, Methodologies and Guidelines", *CHI '12*, 2012, pp 649-658
- Hirtz J, Stone RB, Szykman S, McAdams DA and Wood KL, "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts", *NIST Technical Note 1447*, 2002.
- Kou XY, Xue SK and Tan ST, "Knowledge-Guided Inference for Voice-Enabled CAD", *Computer-Aided Design*, Vol. 42, 2010, pp 545-557
- McCahan S, "Introduction to Engineering Design: Preliminary Edition", Wiley, 2013
- Nanjundaswamy VG, Kulkarni A, Chen Z, Jaiswal P, Shanker SS, Verma A, and Rai R, "Intuitive 3D Computer Aided Design System with Multimodal Interfaces" *Proc. of ASME IDETC/CIE*, 2013
- Pahl G, Beitz W (1996) "Engineering Design: A Systematic Approach, 2nd Edition", Springer-Verlag: London, UK.
- Peterson J, Makesh K and Goel A, "Situating Natural Language Understanding in Experience-Based Design", *Human-Computer Studies*, Vol. 41, 1994, pp 881-913
- Samad T and Director SW, "Towards a Natural Language Interface for CAD", *22nd Design Automation Conference*, Paper 2.1, 1985, pp 2-8
- Sarkar S, Sharma VS and Agarwal R, "Creating Design from Requirements and Use Cases: Bridging the Gap between Requirement and Detailed Design", *Proc. of India Software Eng. Conf. '12*, 2012, pp 3-12
- Schwitter R, "Controlled Natural Language for Knowledge Representation", *Coling 2010*, Poster Volume, 2010, pp 1113-1121
- Sharma A, Madhvanath S, Shekhawat A, and Billingham M, "MozArt: A Multimodal Interface for Conceptual 3D Modeling", *ICIM '11*, 2011, pp 307-310
- Ullman DG, "The Mechanical Design Process", McGraw-Hill, New York, 1992
- Valentine S, Vides F, Lucchese G, Turner D, Kim H, Li W, Linsey J and Hammond T, "Mechanix: A Sketch-Based Tutoring & Grading System for Free-Body Diagrams", *AI Magazine*, Vol. 34, No. 1, 2012, 55-66
- Wang R, Paris S and Popovic J, "6D Hands: Markerless Hand-Tracking for Computer Aided Design", *Proc. of UIST '11*, 2011, pp 549-558
- Zeng Y, "Environment-based design", *Proc. of ASME IDETC/CIE*, 2011