INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

ENRICHING REQUIREMENT-ACTIVITIES IN DESIGN THROUGH FRENCH-US INSTRUCTION COMPARISON

Guy PRUDHOMME (1), Franck POURROY (1), Joshua David SUMMERS (2), Jean-François BOUJUT (1)
1: Grenoble-INP, France; 2: Clemson University, United States of America

ABSTRACT

Engineering requirements are taught through different approaches in US and French universities. In the globalization of engineering product development, these different approaches can introduce semantic challenges among the engineers. If the academic institutions are to educate future engineers for this global environment, then it is the responsibility of the institutions to begin to develop a shared, global understanding of central engineering issues, such as requirements and associated activities. This paper seeks to begin to reconcile and enrich these approaches by examining the activities that are supported through the steps and tools of each approach. Through this, it is found that there are opportunities for improvement in each, such by introducing (1) a requirement spreadsheet to the Grenoble approach for detailing meta-information and (2) the interactors graph to the Clemson approach to help guide engineers in discovering interaction centric requirements. Several other extensions and integrations are suggested to begin to develop a richer global approach, but a deeper study remains to justify and implement this integration.

Keywords: design education, requirements, design tool, AFNOR, function, co-evolution

Contact:
Prof. Joshua David Summers
Clemson University
Mechanical Engineering
Clemson
29634-0921
United States of America
jsummer@clemson.edu

1 TWO PERSPECTIVES OF REQUIREMENTS IN DESIGN

This paper explores the role that requirements elicitation, definition, documentation, and analysis play in engineering design as taught at two different universities, Clemson University in the USA and the University of Grenoble in France. These two different approaches are illustrated based on what is taught, rather than how requirements are actually implemented and used. The goal is to compare these approaches, focusing on the tools and design activities, to highlight similarities and to expose opportunities. This paper is ***not*** a comparison of requirements modeling tools, a review of engineering requirements, a case study of requirements in practice, or experimental study of influencing factors. A review of design tools found in engineering textbooks and the role that is defined in the books is found in (Joshi et al., 2012) and how design problem level of detail relates to solutions in capstone projects (Joshi et al., 2011). Rather, this paper focuses on comparing what is taught and how these taught tools and methods might interconnect or be augmented with approaches from opposite sides of the Atlantic. Our larger motivation, beyond the scope of this paper, is to develop an approach to systematically design product development processes through linking of design tools based on information exchange and support of design activities.

The roles of requirements explained here for both US and French engineering education are based on general practices detailed in standards in France enforcing a commonality in educational experiences (AFNOR, 1991, 1996) or in various design textbooks in the US (Ullman, 2010). However, implementation of each in the classrooms at Grenoble and Clemson each has evolved beyond the linear approaches detailed in the standards or the textbooks to more fully accommodate the basic assumption that engineering design is about the co-evolution of the design problem and the design product simultaneously. This co-evolution is generally studied through practice to understand *how* design happens (Dorst and Cross, 2001; Maher and Tang, 2003). This concept of developing the design problem, exploring new concepts, refining the problem based on the concepts, and developing or refining these concepts based on the evolved problem is embodied in different declarative (Dinar et al., 2011; Gero and Kannengiesser, 2004) and prescriptive (Pahl et al., 2007; Suh, 1990) models of engineering design processes. While others have studied this co-evolution, we are interested here in understanding how to teach requirements to students through the tools and methods employed.

We readily recognize that teaching engineering design is often best realized through practice, as evidenced through the preponderance of literature focused on capstone, cornerstone, and design through the curriculum efforts (Atman et al., 2005; Howe and Wilbarger, 2006; Hyman, 2001). However, even in these problem/project based courses, methods, tools, and processes are taught to the students through lectures and workshops. Often, the linking of these tools and methods are only made apparent to the students during the projects, rather than based on discussions of the operational activities associated with them, such as eliciting, documenting, and reasoning. Thus, the intent and purpose of the tools and methods are not articulated. This paper seeks to begin to provide this justification through a comparison between Grenoble and Clemson teaching approaches. To help illustrate this comparison, a design problem is roughly defined:

> *A homeowner, in the workshop, needs to hold two plates of wood together for different activities. As a first example, the plates of wood need to be held in place as they are glued together and the glue cures. As a second example, the plates are held in place while a hole is drilled through them at the same time.*

This brief will be expanded, detailed, and solutions offered through the two different approaches.

Firstly, in section 0, the main concepts and tools of the French approach are presented. Based on the previous brief, a co-evolution loop illustrates how the problem definition takes place, and how solutions can emerge and lead to refining the initial requirements. In the same way, section 0 examines the US approach and describes some of the associated tools and concepts. In section 0, the global design activity is broken down into more elementary activities, making it possible to identify different steps in the co-evolution process. Finally, a discussion takes place in section 0, in order to compare the two approaches and to match the different tools with the defined activities.

2 GRENOBLE'S APPROACH TO REQUIREMENTS

AFNOR (Association Francais de NORmalisation) is a French association which aims to support the economic development of industrial organizations. Standardization is one of its core activities and is a major challenge for companies. Among the fields covered, the design method and tools is the one that concerns us here, and more particularly the *Functional Analysis* (AFNOR, 1991). Such standard has been extended to the European level (AFNOR, 1996) in which part 1 is about *Value Management, Value Analysis and Functional Analysis vocabulary*. One objective of these standards is to support the product design activities and to ensure solutions more closely aligned to the needs of customers. To do that, a general method for designing is proposed, not so far methodologically speaking from the first step of 'clarifying the task' (Pahl et al., 2007). However, arguing that one has to completely define the design problem before considering solutions, the latter is more problem solving based than co-evolution oriented. It was shown that some methods and tools from this general approach enable designers to work cooperatively and in a co-evolution manner (Prudhomme et al., 2003), and this is what we teach to students.

In France, graduate students follow four semesters of courses to obtain a master's degree. In the University of Grenoble, in the mechanical and industrial engineering schools, methods and tools we present hereafter are taught in the first semester. During the second, students implement them in internal design projects. Then, they use them during their internships in industry at the end of this second and/or during the fourth semester. Figure 1 describes what is taught in the first semester.

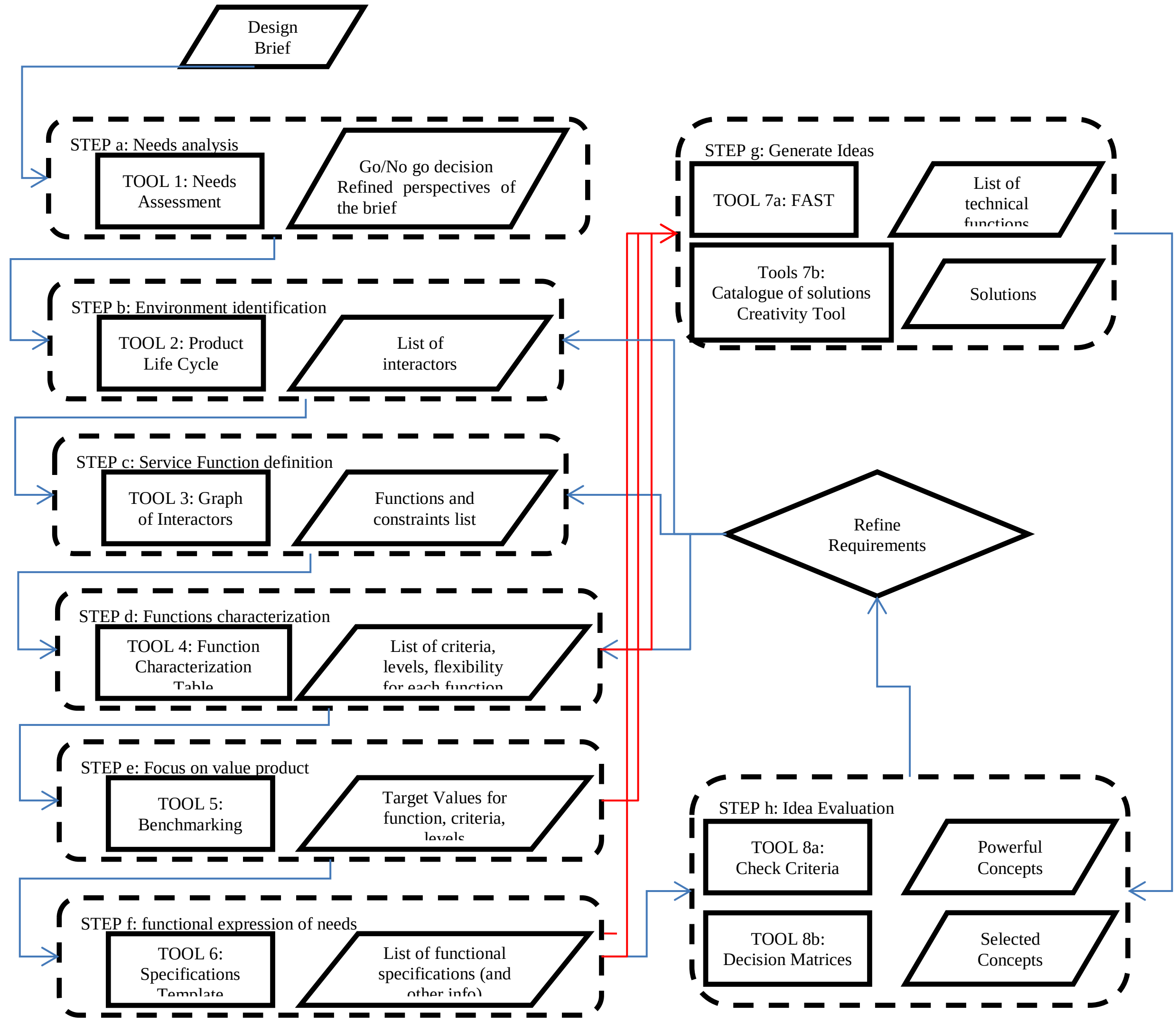

Figure 1: Flow Chart for AFNOR Based Requirements Method

Steps a to *h* do not represent the design process but tasks that have to be performed to design a product. Method and tools we teach are means to support activities related to each of these tasks. Arrows between these steps represent flows of information between the tasks.

The project starts with a design brief containing the wishes or demands of a client. However, while some needs are explicit, many are latent and implicit. The objective of *Step a* is to analyze the needs expressed and to explicit the implicit. The related tool helps the designers to clarify the needs and to understand the context. It is made of two questionnaires. The first one is called *Needs assessment* and includes three questions: Who will be the future product beneficiary (who will be the client)? Why to use it (what will it act on)? For what purpose it has to be used (to do what)? The second one is the *Product needs validation* and includes again three questions: Why will the product exist? What could make the needs evolve? What could make the needs disappear? If the benefit of this first step is clearly a more shared and detailed understanding of the needs and context, the tangible deliverable is mainly a go/no go decision, taking into account that there are stable needs that justify a design activity.

Step b is the first step of external functional analysis. External analysis has the objective to list and to characterize the services the product to design is required to offer irrespective of the means to provide them. To do that, the product to design (or the stuff to design) has to be considered as a black box inserted in an external environment and having to interact with this environment. The objective of *Step b* is to define: the boundary of the stuff, that is to say what is include is the stuff and what is part of the environment; what are outer environment (Simon, 1998) or interactors that have to interact with the stuff during its life cycle. An interactor could be physical or technical (material, energy, other products, …), human (an operator), representing an economical or legal point of view (standard), but it has to be tangible to really interact with the stuff.

Once interactors have been defined, it is possible to define service functions with the graph of interactors from *Step c*. A service function represents an interaction between the stuff and interactors which contribute to needs satisfaction.

It is either an interaction between interactors by the means of the stuff (interaction function) or a simple relation between a single interactor and the stuff (adaptation function). Constraint is a characteristic, effect or provision for design, which is made compulsory or forbidden for whatever reason. A constraint is a limitation of the designers' freedom for choices, something that has to be respected without being the purpose of the stuff. Figure 2 represents the graph of interactors obtained for the plate holder's problem. It highlights interaction and adaptation functions that the stuff has to provide, and constraints the stuff has to respect.

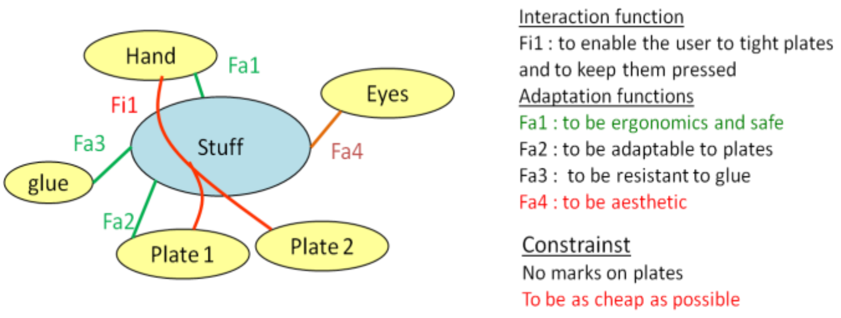

Figure 2: Plate Holder Graph of Interactors

Function characterization is the aim of *Step d*. The objective is to define characteristics (assessment criteria) and associated metrics (levels) to evaluate the expected performances of the stuff. Flexibility is an indication about the possibility of modulating or negotiating the level required for a given assessment criterion. Flexibility is expressed using categories: from F0 (no negotiation possible) to F3 (open discussion). Figure 3 is the resulting table of this *step d* for our plate holder example. Such a table is a dynamic reference of what performances the stuff to design has to perform.

Function	Criteria	Level	Flexibility
Fi1 : to enable the user to tight plates and to keep them pressed	Handling Force Clamping Force Trace on plates Easy to use	10 N ± 2N 50 N ± 5N No trace One hand	F1 F3 F0 F0
Fa1 : to be ergonomics and safe	injury Dimensions touch	No 15x7x3 (cm) Soft	F0 F2 F3
Fa2 : to be adaptable to plates	Thickness Shape default parallelism	From 5mm to 30mm each 10° maximum	F1 F1
Fa3 : to be aesthetic	Colors Shapes	Bright harmony resistant dynamic	F2 F2 F1 F3

Figure 3: Table of Characterization of Functions

The benchmark of *Step e* enables to analyze the existing products and to define target values that make the stuff effective and competitive.

All information built during the previous steps, but also information about the market, context and objectives of the project, report of meeting with customer or stakeholders, evolution of the external expression of the needs, are gathered in a specific document called ‘cahier des charges’. It is the reference for ideas evaluation and for legal issues. But as it is also (based on the co-evolution approach) a document that could evolve throughout the process, it is the base for negotiation that could lead to: modification of levels of assessment criteria; addition or removal of criteria; modification of functions in their definition or structure.

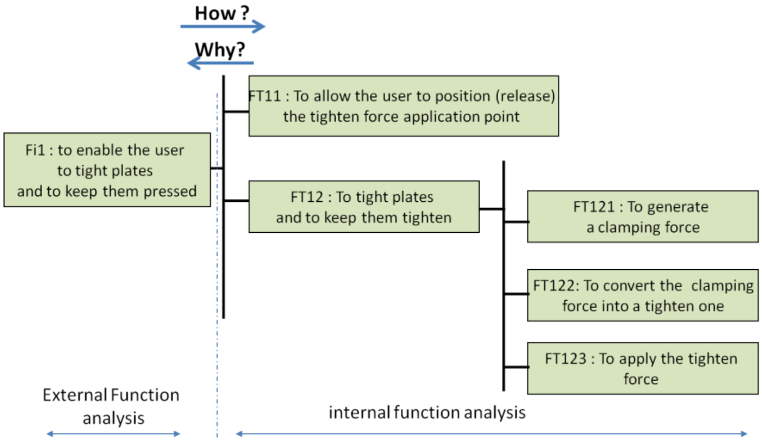

Figure 4: Results from FAST Tool

Internal functional analysis, called sometimes technical functional analysis, proposes means to conduct *Step g* and generate solution. It enables designers to define how the stuff will perform the external functions. A technical function is an action between constitutive parts of the stuff that will participate to provide the required service functions. It represents a technical choice made inside the boundary of the stuff. The FAST tool (Figure 4) helps to move from the external needs to the definition of technical functions to achieve them, and then progressively by deepening and dividing these technical functions into more elementary ones, to obtain technical elements that could realize these expected effects.

3 CLEMSON’S APPROACH TO REQUIREMENTS

The design brief provides a simple problem that has requirements that are both functional (transformational) and non-functional (descriptive of attributes and states); requirements that are both necessary (constraints) and measures of goodness (criteria). In terms of criticality, requirements might be defined as constraints or criteria. Constraints are necessary conditions that *must* be met while criteria are used to evaluate and compare multiple solutions. These are often called musts/shoulds or demands/wishes (K. Otto and K. Wood, 2001; Pahl et al., 2007; Ullman, 2010), but they are generally used to distinguish between requirements in how they are used in the process (validating vs. evaluating). The second dimension of classification relates to the level of detail through “functional” or “non-functional” definitions (Chung and Nixon, 1995; P Shankar et al., 2010). A functional requirement defines a behavior or action that needs to be supported with the system. This is typically defined early in the process and often serves as the main driver for the project. The second type, non-functional requirements, refers to descriptions of the properties of the system. These might include size, cost, mass, material, or others. Typically, the non-functional requirements are derived from the functional requirements. There has been much study of non-functional requirements in software engineering (Chung and Nixon, 1995; Chung et al., 2000), with limited study in mechanical systems (McLellan et al., 2010; P Shankar et al., 2010). It is this second dimension that is of interest in this research, where the functional requirements are defined as those that include an “action” desired.

The Department of Mechanical Engineering at Clemson University, much like other engineering departments in the US has a two semester sequences of senior design courses. In the first semester, students are introduced to the design process in a highly controlled manner; while the second semester is a capstone experience where students demonstrate their understanding of engineering design in application to an industrial sponsored project. The methods taught in the first semester are those that are illustrated here, not the methods or tools actually employed by the students in the second semester. The students are not required to use any specific tools in the second semester. The students are taught eight general steps and associated tools, as illustrated in Figure 5.

The start of the project begins with a design brief that is then explored further by asking a set of questions (K. Otto and K. Wood, 2001; Pahl et al., 2007; Ullman, 2010): Primary Questions: (1) What is the underlying problem really about? (2) What implicit wishes and expectations are involved? (3)

Do the specified constraints actually exist? (4) What paths are open for development? Secondary Questions: (1) What objectives are the intended solution expected to satisfy? (2) What properties must it have? (3) What properties must it not have? The project brief is then extended and detailed with answers to these high level questions. Students are told to revisit these questions throughout the project to ensure that they are continually evaluating and refining their understanding of the design problem. *Step 2* is to define the design objectives through the use of objective trees, or comparable tools (Ullman, 2010). These objectives are expanded by identifying requirements for different topical areas, including function, safety, geometry, material, manufacturing, or recycling (Pahl et al., 2007). The functions identified are linked and decomposed through tools such as functional modeling (K. Otto and K. Wood, 2001; Sen et al., 2010). Functions are defined as the transformations of material, energy, or signal. The functions are the sub-problems for which concept fragments are identified in morphological charts (Richardson et al., 2011; G. Smith et al., 2012). The concept fragments are then integrated into solutions through different idea generation techniques, such as Gallery Method or C-Sketch (Shah et al., 2001; Shah et al., 2003). The requirements from the checklist are detailed with target values on their measures of goodness through benchmarking (Hauser and Clausing, 1988; K.N. Otto and K.L. Wood, 1998). Every requirement, function, and attribute is documented in a formal Product Definition and Specification (PDS) spreadsheet (Pahl et al., 2007). The concepts generated are evaluated against the constraints to determine if they are feasible and then the passing concepts are compared based on the criteria through decision matrices (Ullman, 2010). The requirements are then refined based on a new understanding about the problem or new concepts are explored. New requirements are documented in the PDS and the justification included.

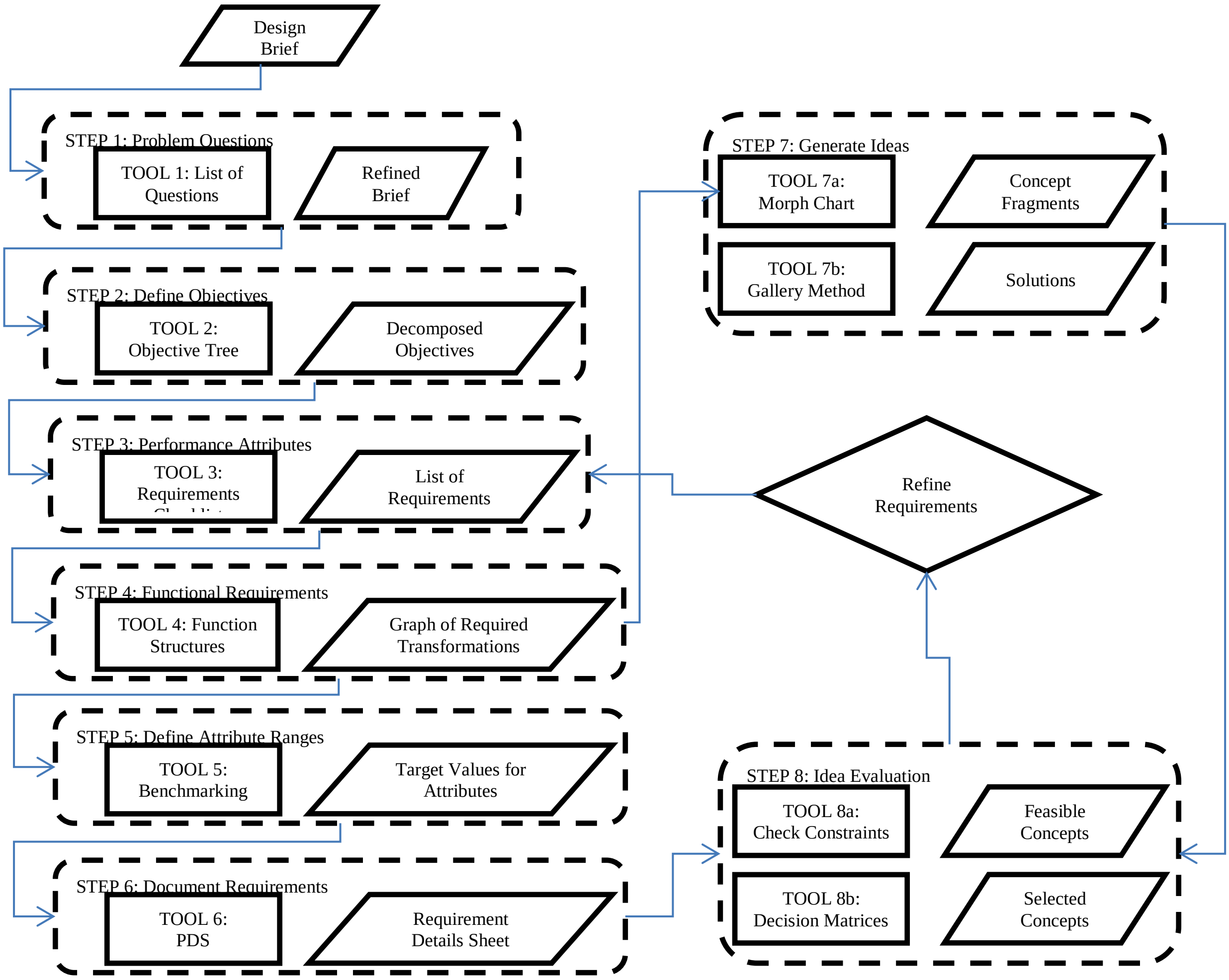

Figure 5. Flowchart for Clemson University Requirements

Figure 6 illustrates a resulting objective tree from *Step 2*. In this objective tree, three objectives are identified: easy to use, safe to use, and not move after being held. These are then decomposed further.

Some advocate applying weights for relative importance of these objectives (Ullman, 2010). The objectives are intended to help describe and understand the design problem.

As the design team progresses, examples of attributes defined through the use of a requirement checklist (*Step 3*) might include: new requirements for geometry (Not open wider than a hand) or loads (weigh less than 0.5 N and be operated by hand).

A functional model might be generated similar to Figure 7. Function structures and functional decomposition is more appropriate for highly transformative devices (Pahl et al., 2007; Schultz et al., 2010). It should be noted that there are many possible function structures that can be developed, each illustrating different perspectives of the problem and the solution architecture. The outcomes from each of the function structures can be used to populate the first column (function list) of morphological charts (G. Smith et al., 2012).These requirements are documented in a PDS (*Step 6*). Next, concept fragments are generated for each of the functions and ideas are integrated.

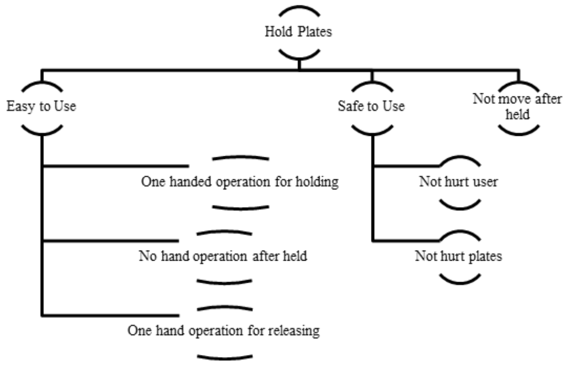

Figure 6. Objective Tree for Plate Holder

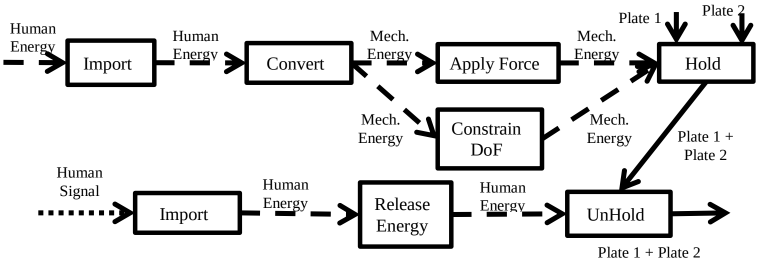

Figure 7. Function Structure for Plate Holding System

4 ACTIVITIES: HOW ARE REQUIREMENTS USED

In reflecting on the methods and tools proposed in the two different schools, we can compare the activities associated with the requirements. These activities are on the roles and use of requirements in engineering design; they do not include traditional “requirements management” activities such as tracing. Three general classes are defined: synthesizing, representing, and reasoning, similar to those found in (Salinas et al., 2008). *Synthesizing* is the activity set that relates to the birth or elicitation of the requirements. This activity might include *understanding* new requirements based on a new and deeper understanding of the problem, customer, and context. The “questions” of *Step a* or *Step 1* are used for this activity. The results from these steps are new requirements based on an interrogation of the general design brief. Engineers might follow this by *discovering* existing requirements with the assumption that requirements need to be uncovered. A customer may not be able to articulate what they want, thus, it is the engineer’s job to discover these requirements. This activity requires significant interaction with the customer, either directly or indirectly. Next, engineers *develop* the requirements from a notional concept to something that is articulable and understandable.

A second class of activity is *formalizing* requirements. This can be by *detailing* individual requirements. The details might include the elements of a requirement, such as target values, the subject (complete system, sub-system, component) of the requirement, or the priority, preference, or weighting of the requirement. Much of the work in defining quality metrics for assessing requirements has focused on the elements, clarity, and internal consistency (Lamar and G. Mocko, 2010). Other details might include conditionality of the requirement such as under what conditions is the requirement actively considered or under what states are the target values set. Beyond formalizing the individual details of a requirement, the context of the requirement might be represented. This context can include the owner of the requirement, the testing and validation approach, or the date and revision of the requirement. This meta-level information is important to support documentation, version control, and project management. Finally, the collection of the requirements can be *related* through a network, tree, outline, or graph structure. This might be hierarchical or peer relation. These formalizations can support trade-off considerations, requirement cascade, or and change analysis.

Thus, the third requirement activity class is the *reasoning* one. This might include *evaluating* the solutions against the requirements (*Step 8* and *Step h*), *making decisions* on how to proceed within the project (*Step a*), *mapping* the requirements to the solution space explicitly (*Step g*), *challenging* or *analyzing* the requirements (*Step 1*).

5 COMPARISON OF THE TWO APPROACHES

The first challenge in comparing these two approaches is through the terminology used in the classroom and the tools employed such as with defining requirements, constraints, criteria, needs, wishes, demands, desires, goals, or objectives. This is not necessarily a translation challenge, but a semantic challenge within each respective language. We do not seek to resolve these issues, but allow the terms and their meanings to co-exist, much as others have proposed this for function (Vermaas, 2013). The two methods and their associated steps are compared to these activities in Table 1. From the table, one can see that some activities are more well covered in the AFNOR approach, such as understanding requirements (*Step a* and *Step b*), detailing requirements (*Step d*, *Step e,* and *Step f*). While, the Clemson approach seems to emphasize the discovering (*Step 2* and *Step 4*), and relating requirements (*Step 2*). Further, there is noticeable gap in an activity that is not supported by either approach, analyzing the requirements, such as change propagation and prediction. The steps that relate to “external” environments as opposed to “internal” environments are bold. Moreover, these steps are not all resolved at the same level of detail.

Table 1: Taught Steps Compared to Requirements Activities

		Grenoble	**Clemson**
Synthesizing	Understanding	***Step a, Step b***	***Step 1***
	Discovering	***Step c***	*Step 2, Step 4*
	Developing	*Step g*	*Step 3*
Formalizing	Detailing	*Step d, Step e, Step f*	*Step 5, Step 6*
	Relating	*Step g*	*Step 2, Step 4*
Reasoning	Evaluating	*Step h*	*Step 8*
	Deciding	*Step a*	
	Mapping	*Step g*	*Step 7*
	Analyzing		

Interestingly, while the activity of understanding is supported in *Step a* and in *Step 1* through the question tools, the questions are not eliciting the same types of information. For instance, in Grenoble, the designers are asked about what could make the need disappear, allowing the engineer to develop a deeper understanding of the underlying problem and the market. This is not explicitly asked as part of the question set for the Clemson *Step 1*. In the Clemson set of questions, the engineers are asked about what properties should not be included or functions should be avoided (*Step 1*). There is no step within Grenoble that supports this type of interrogation.

The discovering activity supported by *Step c* is more guided and focused than the discovery supporting steps of Clemson, *Step 2* and *Step 4*. The graph of interactors seems to be more promising for eliciting non-technical requirements than the objective trees (*Step 2*) or function structures (*Step 4*).

The definitions of constraints and criteria are similar and compatible, yet the understanding of function is not consistent. This aligns with the challenge of function modeling and thinking identified by other researchers (Eckert et al., 2011; Vermaas, 2013). Further, the role of using functions to develop requirements through Clemson *Step 3*. The Grenoble *Step g* includes the definition of technical functions, which appears to be comparable to the definition of functions for Clemson, but it is not used in a requirement developing activity. The Grenoble view on functions also includes the human user, a concept that is not clearly captured in any of the Clemson steps. Therefore, the Clemson approach could be augmented with the inclusion of the graph of interactors (*Step c*) to capture this. Matching the user centric approach of Grenoble to the technical centric approach of Clemson is identified as a key for future study. One possible approach to this could be affordance, scenario, or function-interaction based (Carroll et al., 1998; Maier and Georges M. Fadel, 2008; Ramachandran et al., 2011).

While both approaches appear to cover the detailing activity well, the type of information detailed is not the same. For instance, Clemson *Step 6* (PDS tool), includes a worksheet form for the engineers to identify the source of the requirement, the justification for the need for this requirement, the responsible party for testing, and the type of verification that should be done. This meta-information for the requirements is not captured in any of the Grenoble detailing activities (*Step d*, *Step e*, *Step f*). Thus, integrating a formal requirement spreadsheet tool, such as the PDS of Clemson *Step 6*, could help improve the coverage of the Grenoble approach.

Another activity that appears supported, based on Table 1, is the relating activity. For both approaches, the tools relate requirements in a hierarchical manner. Clemson *Step 4,* includes a peer-level relationship between functions, but not for all requirements. Within relating, the FAST tool helps decompose the initial functions, but it is weakly guided and controlled from the engineer’s perspective.

Therefore, the FAST (*Step g*) could be augmented with labeling the types of flows in the graph, similar to the function structure (*Step 4*). A new tool and step could be integrated into both approaches that explicitly relates the requirements to each other, such as a Design Structure Matrix type of model (Maier et al., 2007; Beshoy Morkos et al., 2012).

There are numerous complementary aspects of the different tools and steps for the two approaches. Discovering these opportunities through the supported activities and identifying the contributing characteristics of the tools is identified as the next step in this research. While this is a first step in reconciling the two perspectives, a global approach could provide a richer design process.

ACKNOWLEDGMENTS

We gratefully acknowledge the support from G-SCOP and CNRS in the form of the Visiting Professorship for 2012-2013.

REFERENCES

AFNOR. (1991) *NF X 50-151 Analyse de la Valeur, Analyse Fonctionnelle, Analyse de la Valeur, Analyse Fonctionnelle-Expression fonctionnelle du besoin et cahier des charges fonctionnel*, Paris, France.

AFNOR. (1996) *NF EN 1325-1: Vocabulaire du management par la valeur, de l'analyse de la valeur et de l'analyse fonctionnelle–Partie 1: Analyse de la valeur et analyse fonctionnelle, Paris, AFNOR Editions*, Paris, France.

Atman, C., Cardella, M., Turns, J. and Adams, R. (2005) Comparing freshman and senior engineering design processes: an in-depth follow-up study, *Design Studies*, Vol. 26 No. 4, pp. 325–57.

Carroll, J., Rosson, M., Chin, G. and Koenemann, J. (1998) Requirements development in scenario-based design, *Software Engineering, IEEE Transactions on*, IEEE, Vol. 24 No. 12, pp. 1156–70.

Chung, L. and Nixon, B. (1995) Dealing with Non-Functional Requirements: Three Experimental Studies of a Process-Oriented Approach, *17th International Conference on Software Engineering*, Seattle, WA.

Chung, L., Nixon, B.A., Yu, E. and Mylopoulos, J. (2000) *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers, Internatio., p. 476.

Dinar, M., Shah, J., Hunt, G., Campana, E. and Langley, P. (2011) Towards a formal representation model of problem formulation in design, *Proceedings of ASME DETC*.

Dorst, K. and Cross, N. (2001) Creativity in the Design Process: Co-Evolution of Problem Solution, *Design Studies*, Vol. 22, pp. 425–437.

Eckert, C., Alink, T., Ruckpaul, A. and Albers, A. (2011) Different notions of function: results from an experiment on the analysis of an existing product, *Journal of Engineering Design*, Taylor & Francis, Vol. 22 No. 11-12, pp. 811–837. doi:10.1080/09544828.2011.603297

Gero, J.S. and Kannengiesser, U. (2004) The situated function–behaviour–structure framework, *Design Studies*, Vol. 25 No. 4, pp. 373–391. doi:10.1016/j.destud.2003.10.010

Hauser, J.R. and Clausing, D. (1988) The House of Quality, *Harvard Business Review*, Vol. 66 No. 3, pp. 63–73.

Howe, S. and Wilbarger, J. (2006) 2005 National Survey of Engineering Capstone Design Courses, *ASEE Annual Meeting, Chicago*.

Hyman, B.I. (2001) From capstone to cornerstone: A new paradigm for design education, *International Journal of Engineering Education*, TEMPUS PUBLICATIONS, Vol. 17 No. 4/5, pp. 416–420.

Joshi, S., Morkos, B and Summers, J D. (2011) A Coding Scheme for Analyzing Capstone Design Reports: Problem and Solution Descriptions, *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, ASME, Washington, DC, Vol. 2011, pp. 559–568.

Joshi, S., Morkos, Beshoy, Shankar, Prabhu and Summers, Joshua D. (2012) Requirements in Engineering Design: What are We Teaching?, in Horvath, I. (Ed.),*Tools and Methods for Competitive Engineering*, Karlsruhe, Germany, p. no. 38.

Lamar, C. and Mocko, G. (2010) Linguistic Analysis of Natural Language Engineering Requirement Statements, in Horvath, Imre (Ed.),*Tools and Methods for Competitive Engineering*, Ancona, Italy.

Maher, M. and Tang, H.H. (2003) Co-evolution as a computational and cognitive model of design, *Research in Engineering Design*, Springer, Vol. 14 No. 1, pp. 47–64.

Maier, J.R.A., Ezhilan, T., Fadel, G M, Summers, J D and Mocko, G. (2007) A hierarchical requirements modeling scheme to support engineering innovation, *International Conference for Engineering Design, Paris, France*.

Maier, J.R.A. and Fadel, Georges M. (2008) Affordance based design: a relational theory for design, *Research in Engineering Design*, Vol. 20 No. 1, pp. 13–27. doi:10.1007/s00163-008-0060-3

McLellan, J.M., Morkos, B, Mocko, G.G. and Summers, J D. (2010) Requirement Modeling Systems for Mechanical Design: A Systematic Method for Evaluating Requirement Management Tools and Languages, *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, ASME, Montreal, Canada, pp. DETC2010–.

Morkos, Beshoy, Shankar, Prabhu and Summers, Joshua D. (2012) Predicting requirement change propagation, using higher order design structure matrices: an industry case study, *Research in Engineering Design*, Taylor & Francis, Vol. in press, pp. 1–22. doi:10.1007/s00163-012-0132-2

Otto, K. and Wood, K. (2001) *Product Design Techniques in Reverse Engineering and New Product Development*, Prentice Hall, Upper Saddle River, NJ.

Otto, K.N. and Wood, K.L. (1998) Product evolution: a reverse engineering and redesign methodology, *Research in Engineering Design*, Springer, Vol. 10 No. 4, pp. 226–243.

Pahl, G., Beitz, W., Wallace, K. and Blessing, L. (2007) *Engineering Design: A Systematic Approach*, Springer-Verlag London Limited, London, 3rded.

Prudhomme, G., Zwolinski, P. and Brissaud, D. (2003) Integrating into the design process the needs of those involved in the product life-cycle, *Journal of Engineering Design*, Taylor & Francis, Vol. 14 No. 3, pp. 333–353.

Ramachandran, R., Caldwell, B.W. and Mocko, Gregory M. (2011) A User Study to Evaluate the Function Model and Function Interaction Model for Concept Generation, *Volume 9: 23rd International Conference on Design Theory and Methodology; 16th Design for Manufacturing and the Life Cycle Conference*, ASME, Vol. 2011, pp. 273–284.

Richardson III, J.L., Summers, J D and Mocko, G M. (2011) Function Representations in Morphological Charts: An Experimental Study on Variety and Novelty on Means Generated, *Interdisciplinary Design: Proceedings of the 21st CIRP Design Conference*, p. 76.

Salinas, M.P.C., Prudhomme, G. and Brissaud, D. (2008) Requirement-oriented activities in an engineering design process, *International Journal of Computer Integrated Manufacturing*, Taylor & Francis, Vol. 21 No. 2, pp. 127–138.

Schultz, J., Sen, C., Caldwell, B., Mathieson, J., Summers, J D and Mocko, G M. (2010) Limitations to Function Structures: A Case Study in Morphing Airfoil Design, *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, ASME, Montreal, Canada, pp. DETC2010–28559.

Sen, C., Summers, J D and Mocko, G M. (2010) A Formal and Extended Representation of the Functional Basis, *Tools and Methods for Competitive Engineering (TMCE 2010)*, p. No–127.

Shah, J J, Vargas-Hernández, N., Summers, J D and Kulkarni, S. (2001) Collaborative Sketching (C-Sketch)—An Idea Generation Technique for Engineering Design, *The Journal of Creative Behavior*, Blackwell Publishing Ltd, Vol. 35 No. 3, pp. 168–198.

Shah, Jami J., Smith, S.M. and Vargas-Hernández, N. (2003) Metrics for Measuring Ideation Effectiveness, *Design Studies*, Vol. 24, pp. 111–134.

Shankar, P, Morkos, B and Summers, J D. (2010) A Hierarchical Modeling Scheme With Non Functional Requirements, *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, ASME, Montreal, Canada, pp. DETC2010–28544.

Simon, H. (1998) *The Sciences of the Artificial*, MIT Press, Cambridge, MA.

Smith, G., Richardson, J., Summers, J D and Mocko, G M. (2012) Concept exploration through morphological charts: an experimental study, *Journal of mechanical design*, American Society of Mechanical Engineers, Vol. 134 No. 5, pp. 051004.1–10.

Suh, N.P. (1990) *The principles of design*, Oxford University Press, New York, NY, Vol. 990.

Ullman, D.G. (2010) *The Mechanical Design Process*, McGraw-Hill, New York, NY.

Vermaas, P.E. (2013) On the Co-Existence of Engineering Meanings of Function: Four Responses and Their Methodological Implications, *Artificial Intelligence in Engineering Design Analysis and Manufacturing*, Vol. 27 No. 3, pp. 1–19.