An Introduction to The Consensus Model of Engineering Design Decision Making

David G. Ullman
Mechanical Engineering
Oregon State University
Corvallis, OR 97331
ullman@me.orst.edu

Bruce D'Ambrosio
Computer Science
Oregon State University
Corvallis, OR 97331
dambrosi@cs.orst.edu

Abstract

This paper presents an introduction[1] to a model of a collaborative working environment which supports distributed team argumentation, negotiation, consensus building and rationalecapture. Based on a natural model of team deliberation, this model is the basis for the development of a system which enables team support and the capture of the design rationale in value added activities. The consensus model is the result of over fifteen years of studying and modeling design engineers by the authors and the integration of research results from the fields of negotiation and argumentation modeling, design rationale capture, decision theoretics, and engineering best practices.

1. Introduction

We have developed a relatively simple model of engineering decision making. It integrates concepts such as criteria development, value modeling, argumentation, negotiation and evaluation into the engineering design workflow. It is based on a legacy of models from engineering, psychology, and sociology. We call this model the Consensus Model.

The Consensus Model was developed to help clarify the information developed during the natural design decision making work flow of teams and individuals. The terminology used is carefully defined. These classes of information serve as a basis for unifying design best practices and for design rationale system development.

Engineering design process best practices such as those for capturing and managing requirements, and those for structuring evaluation, have matured greatly over the last fifteen years. Each best practice embodies a decision making strategy to help resolve a design issue and so supports team collaboration and rationale capture. A robust model of decision making such as the Consensus Model provides a unifying semantic basis for best practice strategies.

In order for design rationale systems to be developed and used, the decision making process needs to be modeled and understood. The Consensus Model defines the classes of information needed for the capture and organization of design rationale information.

Finally, the Consensus Model serves as a foundation for the development of analytical techniques that support the decision making process. These techniques go beyond the traditional analysis in support of evaluation and support argumentation, negotiation and process planning.

In this paper the Consensus Model is developed in Section 2. Background and support for the model is presented in Sections 3 and 4. Finally, in Sections 5 and 6 the Consensus Model is used to explain some example best practices and how it can serve as a basis for design rationale capture.

2. The Consensus Model

The authors have individually been studying the design process and decision making for over fifteen years. For the last four years we have been collaborating in an effort to develop a model that unifies our work. The model that has evolved, called the Consensus Model, is well supported by prior research[2]. Key features of this model are:

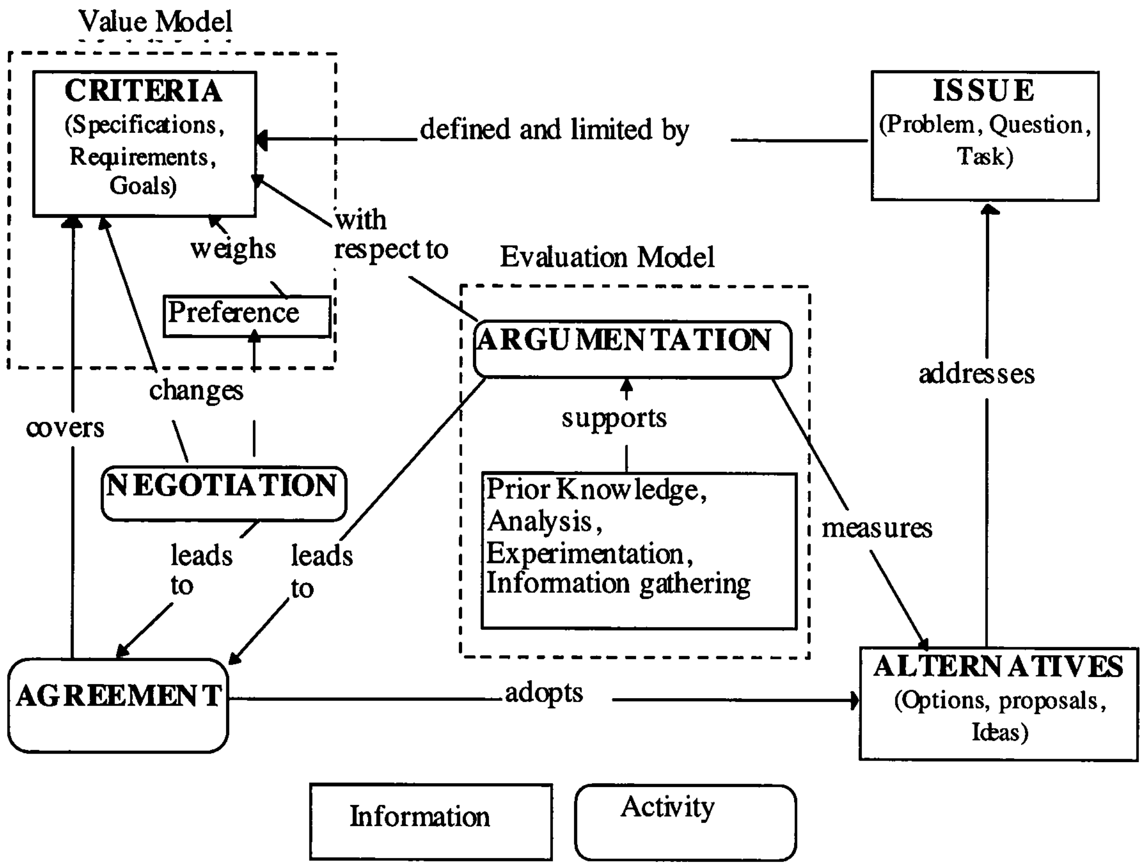
1. It combines the best of decision support research from many different, previously unconnected disciplines, such as decision theoretics, design problem solving, argumentation research, negotiation research,

[1] The full paper has been submitted for review to Design Studies.

[2] See Section 5. For Related work.

analytical support of engineering evaluation and requirements capture.
2. It integrates important classes of product and process information. Thus, it forms a foundation for design rationale capture and for enabling collaborative team work.
3. It can represent decision making by individuals and teams.
4. It can represent decision making at the finest, cognitive level (typically 1 minute in duration) to the corporate planning level (typically weeks or months in duration).
5. It can explain and unify strategies and semantic structures proposed by engineering design best practices.

The Consensus Model is shown in Figure 1. Each of the classes of information and their relationships shown in the figure are defined in the text below. The Consensus Model will be discussed in building block fashion, one block at a time. This is to enable the careful definition of terms. Careful definition is necessary as this model borrows from so many different disciplines and there is a natural problem with many nearly-synonymous terms. Each section adds a new term and describes its relationship with the previously defined terms. Supporting information is in the following sections.



2.1. Issue

An Issue is a statement of an area of concern. It is a call for action to resolve some question or a problem. An issue is defined and limited by the criteria used to measure its resolution. During design activities, a majority of the issues state the desire to generate an alternative to satisfy the criteria associated with the issue. Other types of issues observed in studies of designers in action are the need to gather information (assimilation issues) and for planning and ordering subsequent activities (planning issues).

Design issues are generally expressed as the desire to change, refine, design, redesign, create, fix, develop, or choose an object that meets a number of stated and unstated criteria. The term "object" can mean system, assembly, part, or feature. It can refer to hardware, electric device or software. It can refer to the form or function of the object.

2.2. Criteria

A Criterion limits solutions to the concern raised by an issue. The term "criterion" is here used synonymously with requirement, goal, constraint or specification, as all limit the space of acceptable solutions for the issue. In actuality, issues are usually defined by their criteria. In other words, issues are articulated by designers in terms of their criteria. In design problems criteria usually define functional performance, human factors, physical requirements, reliability, life-cycle concerns, resource concerns or manufacturing requirements.[3]

Criteria are developed by the issue stakeholders, those individuals responsible for or affected by the resolution of the issue. Each stakeholder has preference for how important the criterion is to the successful resolution of the issue.

The combination of the criteria and the preference for them is often called the *value model* because their combination is used to measure or place a value on the alternatives. The relationship between the value model and the alternatives will be refined in the following sections.

A well formulated criterion contains three pieces of information: an attribute, the target structure for this attribute and the conditions under which the criterion applies. In other words: *Each criterion limits the solution by defining a conditional target for one or more attributes of the alternatives proposed to resolve the issue.* Based on this definition, the value space is characterized by four measures: attributes, targets, conditions and preference.

Each criterion states that, in order for an alternative to be a satisfactory solution for an issue, a certain attribute of it must compare favorably to a target. For example, while selecting a material for a component, a criterion might be; "it must be hard" or "material has hardness greater than 50 on the Rockwell C scale." The first of these uses "it" to describe the alternative, "hard" for the attribute and measure, and has an implied target of the-harder-the-better. This is not a very clearly stated criteria.

[3] For a detailed list of criteria types see page 109 in *The Mechanical Design Process* [Ullman 97].

In the second example, the attribute is clearly "hardness", the measure "Rockwell C" and the target is "greater than 50."

2.3. Alternatives

An Alternative is an option generated by a designer or the design team to address or respond to a particular design issue(s). The goal is to find an alternative that the participants agree to adopt. Alternatives are often called "options", "ideas", "proposals", or "positions." Any number of alternatives may be developed to resolve a design issue. Some important points about alternatives are: (1) information known about an alternative is often incomplete; (2) alternatives are often dependent on the resolution of other issues; (3) alternatives will often introduce new sub-issues; (4) alternative solutions to one issue may be inconsistent with those to solve another; (5) one alternative may be proposed to resolve more than one issue; (6) information concerning an alternative may be abstract, concrete or some mix of the two.

Alternatives can be represented as physical artifacts, graphically (formal drawings or rough sketches), or the values of attributes that describe the attributes of the artifact.

2.4. Evaluation

Evaluation is the activity of determining how well the alternatives resolve the issue. This requires the measurement of an alternative's attributes for comparison with the criteria. Where the generation of alternatives and criteria result in the actual instantiations of objects, evaluation is an activity. The only resulting artifacts are the results of the evaluation, e.g. the notes from a meeting, the data from an experiment, or the data from a computer simulation.

Evaluation is the activity of argumentation supported by information developed through prior knowledge, analysis, experimentation or information gathering.

Argumentation is the act of drawing conclusions and applying them to the issue under discussion.[4] Thus, argumentation leads a designer or a member of the design team to a belief about an alternative's ability to meet a criterion. Belief

[4] This is a paraphrase of the Merriam-Webster Collegiate Dictionary definition of "argumentation." Note that argumentation does not imply an argument among people.

can be seen then as a representation of the likelihood of success for a particular solution to the problem.[5] Argumentation has two dimensions; knowledge about the alternative and confidence in alternative satisfying the criterion.

Knowledge is a measure of how much the evaluator knows about the attribute of the alternative being evaluated. During design activities knowledge is generally increased through experimentation, i.e. doing simulations (analytical and physical), or finding additional sources of information (e.g. books, experts, consultants). A reality of team design is that the members of the team each have a different level of knowledge about each attribute of every alternative.

Confidence is a measure of how well the evaluator believes the alternative's attribute actually meets the target set by the criterion. Confidence can be raised or lowered as knowledge is increased. Members of a design team may have different levels of confidence when presented with the same analysis, experimental results or other efforts to increase knowledge. In other words, belief in results of evaluation may not be the same for all team members.

It is useful to represent the knowledge and confidence on a Belief Map, shown in Figure 2. Belief maps offer a quick, easy to use tool for an individual or a team to ascertain the status of an alternative's ability to meet a criterion and to visualize the change resulting from analysis, experimentation or other knowledge increase.

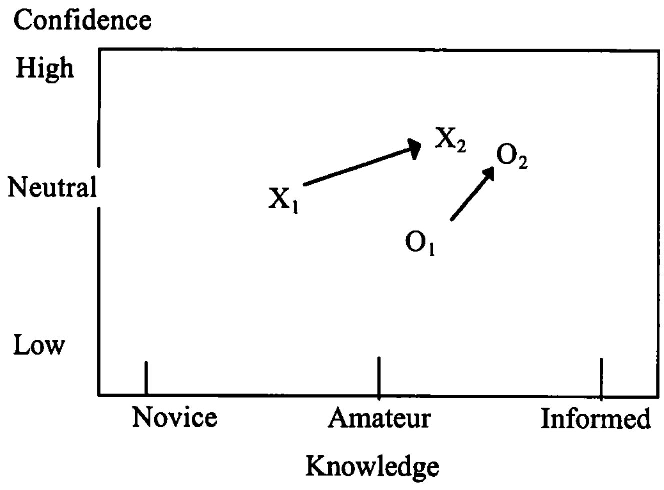


Figure 2: A Simplified Belief Map

[5] This is a decision theoretic view of argumentation and belief. For more on this see [Herling et.al. 95].

To show how a Belief Map works consider the example in Figure 2. Here two engineers are evaluating a specific attribute of an alternative relative to a criterion. On their initial attempt to evaluate the alternative, subscript 1, Engineer X has medium confidence in the alternative's ability to meet the target set by the criterion based on his prior knowledge. Meanwhile Engineer O has more initial knowledge and less confidence in the alternative. After some analysis, experimentation or other information gathering, the two engineer's belief about the alternative has changed, subscript 2. Both engineers now have more knowledge, yet they do not have the same confidence, as they interpret the results of their fact finding differently. However, through discussion (i.e. argumentation) their evaluation is converging, they both know more, their knowledge has been increased based on the same information and their confidence is higher. In this example, the engineers' confidence went up with their increased knowledge, it could have gone down. As knowledge increases, confidence either increases toward the upper right corner leading toward alternative acceptance or decreases toward the lower right corner leading to alternative abandonment. It is the history of this change in confidence that is key to the rationale behind design decisions.

Thus, *an argument is the rationale for either supporting or opposing a particular alternative.* Argumentation measures alternatives with respect to criteria, and these arguments lead to agreements. Arguments are attribute comparisons made either between alternatives with respect to a criteria, or between an alternative and a criterion. Arguments are made with varying levels of confidence, knowledge, and abstraction and are posted by various members of the design team. They may be inconsistent with each other and across team members. They may be incomplete (i.e. not all arguments are made about all the alternative/criterion pairs) and they may differ in both value and confidence. To refine the arguments, the team must allocate scarce resources. Part of a decision on an issue is how much time and expertise to devote to increasing the completeness, confidence and consistency of the arguments.

2.5. *Agreement*

Agreement is the activity of reaching a decision to adopt an alternative(s) to resolve the issue. Decisions are dynamic, they may later be changed as criteria and preferences change and new alternatives are generated. Early in the design process, decisions are often made to leave an issue open pending the resolution of other, interdependent issues.

Although the term "agreement" generally refers to a group of people, its use here for an individual designer is also consistent. An engineer has to reach some agreement with him/herself about which alternative to select. Agreement, either for a team or an individual is reached through argumentation (evaluation) and negotiation (value modeling).

2.6. *Negotiation*

Negotiation is the activity through which a consensus for the value model (i.e. criteria and preference) is reached. The value of an alternative is measured by its ability to meet criteria tempered by the preference shown for these criteria. In other words, criteria that are thought to be more important by the decision maker(s) weigh more heavily on the assessment of alternative value.

During the solution of a design problem by a team, the resulting individual assessments may conflict as team members disagree on the "best" alternative. This conflict can either be about the preference (i.e. the weight) of a criterion, the target value of the criterion itself, or the validity of criterion in the problem at all. For example, participants might disagree on how important it is to achieve a 2 second response time. They might disagree on whether the target should be 1.5 seconds or 2 seconds, or they might disagree on the relevance of response time to this problem at all (or any combination of these).

Although the term "negotiation" usually is used to refer to group activity, individual designers also negotiate about preference and criterion details in that they must manage these information.

2.7. *Consensus Model Summary*

The model presented in Figure 1 is the heart of this paper. Note, that even though this is a model of decision making, it does not include a class of information "decision". Earlier versions of this model did include this, however it proved awkward and did not capture any clear information or activity.

3. The Genesis of the Consensus Model

Rittel [Rittel and Webber, 73] describes IBIS, a model for organizing the deliberation process that occurs during complex decision making. The IBIS model organizes the deliberation process into a network of three data elements; *Issues, Positions,* and *Arguments.* An issue is an identified problem to be resolved by deliberation. Each issue can have many *Positions* that are proposed solutions developed to resolve the issue. Each position can have any number of *Arguments* that support or oppose that position.

Where the IBIS model focuses on alternative identification and selection, Rittel noted that there is also a process of *negotiation* in decision making. Barry Boehm and his colleagues have studied negotiation in the development of software engineering requirements. Theory-W [Boehm and Ross, 89, Boehm, et al., 95] and the Win-Win framework [Bose, 95] provide a domain independent framework in which stakeholders collaborate and negotiate during the requirements engineering phase of software design.

Theory W states that a project will succeed "if and only if you make winners of all the critical stakeholders." Agreement among the stakeholders is insured by finding the intersection of requirements that the stake holders consider important (i.e. a win-win situation). If there is no intersection then feasible options consisting of a relaxed set of requirements (i.e. criteria) that still address the issue is found through negotiation leading to a non-lose or win-satisfactory situation. In the Theory W model there are four "artifact" classes: Win Conditions, Issues, Options and Agreements. These map to the value model, issues, alternatives and agreements in the Consensus Model.

Although Boehm's work is focused on the development of software requirements (not alternatives to meet requirements), it does emphasize the negotiation involved in modifying the requirements until agreement is reached.

The Consensus model can also trace its formulation to descriptive studies of individual designers based on Herbert Simon's information processing system (IPS) model of human cognition [Stauffer and Ullman 91]. In this work a detailed model of individual mechanical designers was developed based on empirical studies of five engineers. This model showed the strategy followed by designers on a cognitive level, with decisions made at approximately 1 minute intervals. This model was extended to cover teams using concepts from IBIS model of deliberation [Nagy and Ullman 92]. It was further refined in the identification of the types of information designers need for a design rationale system [Kuffner and Ullman 91]. Since the early 90s, the Consensus Model has evolved with the addition of concern for computational support methods commonly used in engineering such as optimization, utility theory and Taguchi's method of robust design, and other best practices.

IBIS/DT is a decision-theoretic extension of the IBIS representation. This model has a rigorous decision-theoretic analytical capability for supporting argumentation and negotiation. In the Engineering Decision Support System, EDSS [Ullman *et al*, 1997, Herling 97, Herling et.al. 95], an instantiation of IBIS/DT, team members begin by individually identifying alternatives and criteria for the issue. For each alternative/criterion pair, each team member's argumentation is based on *knowledge/confidence* semantics, a two dimensional measure of his/her evaluation of this pair. Entry of an individual argument is a fairly straightforward process, one simply states the likelihood that an alternative will meet a criterion, qualified by a level of expertise of the evaluator. Negotiation is supported through the semantics of the criteria and the weight assignment on each. EDSS records the deliberation through a database entry of each addition or change made by the team members. Beyond supporting deliberation, sorting and searching this database makes EDSS a design rationale system.

In EDSS, several analyses are possible. First, in support of argumentation, the *expected value* for each decision can be computed. Decisions can be ranked by expected value, with the highest expected value indicating the most preferred decision. The intent of this computation is not that the system should dictate a decision, but rather that the analysis aids the team in understanding and integrating the information presented to date. The expected value analysis can also be used in support of negotiation [Ullman et.al. 97] Second, sensitivity measures, such as informativeness and *value of perfect information* can be computed for each alternative/criterion pair. Again, these can be ranked, with the highest value indicating the most useful pair on which to gather further information (i.e., expand/refine the argument model, perhaps by thought experiments, scenario creation and evaluation, running experiments or other research, etc.). This type of analysis supports process analysis [Ullman et.al. 97].

4. Support for the Consensus Model

In the 1980s MCC implemented the IBIS model on the computer for ease of posting issues, positions (here called proposals) and arguments. The resulting tool, called gIBIS [Conklin and Begeman 88] (graphical IBIS) was designed as both a record keeping and decision organizing tool. gIBIS is a Hyper-text tool where the information can be graphically presented as a network of nodes representing issues, positions and arguments. A software development project at NCR used IBIS (on paper) and gIBIS (on computers) to support the design of a software system (Yakemovic and Conklin 89,90). The entire project addressed 2260 issues. Some of the results from this study are:

- The IBIS and gIBIS tools provided a shared memory for the design team. The history of decisions made and recorded in the tools was easily reviewed and utilized by members of the design team and management.
- The use of IBIS and gIBIS helped the design team detect design issues that had ``fallen through the cracks." In fact, it was estimated the resulting savings were 3-6 times greater than the cost of using the tools.
- The use of these tools helped the team to more quickly understand the problem they were trying to solve.
- The tool helped structure information (issues, positions, and arguments) and helped establish and focus the agenda, keeping team meetings productive.
- Use of the tools supported team communication with other levels of the organization. People not at a design team meeting could easily discern what was discussed -- not just what the final decision was.

Olson [Olson et.al. 92, 96] and his colleagues studied 10 software design groups of 3-7 people to find the structure of the design arguments. They found they could model the team's argumentation using an extended set of IBIS classes.

Herling [Herling 97, Herling et.al. 95] based EDSS on a model that predates the Consensus Model. EDSS could perform reporting and supporting type activities across most of the Consensus Model classes of information. Herling built a computer based system and validated it with balanced design experiments across teams, some using EDSS and some using only paper. The results of these experiments showed that the reporting capabilities and some of the supporting capabilities of EDSS clearly added value to the team.

5. Engineering Best Practices and the Consensus Model

Engineering design best practices are blueprints for design information organization and decision making. Typical of these practices are company specific practices and general engineering practices such as: Quality Function Deployment, QFD, that helps the design team to develop, organize, manage customer requirements and engineering specifications; Robust Design methods that help organize analytical or experimental evaluation; and Task Scheduling that helps designers develop the optimal ordering for tasks. Best practices define the classes of information, their organization and strategies for structuring the process.

A complete discussion of mechanical engineering design best practices can be found in *The Mechanical Design Process* [Ullman 97]. Computer Science best practices are found in literature about the Unified Modeling Languages, UML, as well as in the SEI Capability Maturity Model. There is no known unified reference of electronic or electro-mechanical systems.[6]

To see the relationship of these practices to the Consensus Model consider the following discussion about QFD and Taguchi's method. QFD is a refinement of the Consensus Model for the specific issue of "develop the customer requirements and engineering specifications for the product or system." In its most basic structure it gives a form (the House of Quality) for developing the information and so records the results of the deliberation. For each room in the house (each room defines a sub-issue) there are criteria for the alternative information developed (i.e. engineering specifications must be measurable). Most instantiations of QFD[7] also have a weak process analysis capability. This attempts to give support for the process by identifying which engineering requirement is most important and must be worked on. Currently, this is not well developed or used by many practitioners as it has low value added.

Taguchi's method of robust design is a refinement of the Consensus Model for the issue "optimize the parameters to maximize product function or minimize manufacturing variation." It reports, supports, and guides. An early step in the process is to determine the control factors (their locus defines the set of alternatives) , the noise factors (they are part of the criteria definition), and the signal and quality measure (these refine the definition of the issue). There is a specific form for these pieces of information. Taguchi's method gives a well defined method for evaluation. The results of this evaluation are the sensitivity of the mean value and the signal/noise ratio to changes in the alternative parameters. Although the method gives some guidance for the argumentation, it is then left to the engineers to interpret the results. The method is quite structured and gives good guidance about what to do next.

6. The Consensus Model as a Basis For Design Rationale Capture and Query

Product development is a large and complex task, and early design decisions can have a crucial impact on the quality, cost, robustness, and reusability of the product. Developers are facing the task of maintaining large-scale, complex systems for decades, in an environment where contributors do not stay on the project for more than a few years. Organizations managing such projects have recognized the need to capture,

[6] In fact, the Electrical and Computer Engineering Department at OSU uses *the Mechanical Design Process* in their cap-stone design course.

[7] Instantiations of QFD: QFD DesignerQS, QS Software Corp. Rochester Hills, MI; and QFD/Capture, International TechneGroup Inc., Columbus, OH.

manage, and browse design rationale, but they cannot today find practical tools for the task.

Design rationale includes the design itself, why it is structured that way, and how the design evolved. Without knowledge of what exists and why it exists the way it does, a system upgrade team is left with little more than "system archeology" to gain an understanding of existing systems. To date, most design rationale systems force designers to use specialized notations and work outside of their normal workflow to enter design rationale [Ullman 94]. Further, such systems offer little immediate pay-back to the design team to compensate for this burden. As a result, such systems are rare.

Design rationale representation and capture has been an active research area for some time. The IBIS and gIBIS systems [Yakemovic and Conklin, 89, 90] were the first to apply Rittel's Issue Based Information System (IBIS) representation to software design rationale capture. The IBIS model of rationale has since become something of a default standard in design rationale representation despite its many limitations. QOC [McKerlie and MacLean, 93; Shum, 93] is the representation used in the Design Space Analysis (DSA) approach to design rationale. QOC stands for *Questions*, *Options*, and *Criteria*, the three fundamental elements of the representation. DSA and QOC differ from previous approaches in two fundamental ways. First, the intention is not design rationale *capture*, but rather post-hoc *construction*. Second, QOC introduces the notion of objectives or criteria into the representation, a notion lacking in earlier systems. PPIS [Monk, 96] views design rationale as a co-product of the design process. PPIS is perhaps closest in spirit to the Consensus Model: it supports qualitative and quantitative evaluation of alternatives. However, it is intended for a single designer (no effort is made to fuse arguments or represent multiple perspectives), and it does not provide analyses beyond alternative evaluation. The Sibyl system uses a semiformal representation for capturing and reusing decision rationales. In this system, some knowledge is represented formally, e.g. relationships between the *Goals*, *Alternatives*, and *Arguments* for an issue. The descriptions of each of these however, can be a mixture of free text and formal objects representing domain knowledge. The approach of providing the minimal formalism required to support automated processing goals, while permitting the intermixing of free text and other media, is very much in accord with the philosophy of Consensus Builder. A more formal approach has been taken in the development of DRL (Design Rationale Language). KBDS [Alcatara and Lababidi, 95] is a more ambitious effort to integrate *design history*, *design structure*, and *design rationale* into a single environment.

The Consensus Model and its earlier instantiation, IBIS/DT, attempt to model the elements necessary for the capture of design rationale information as part of the normal workflow. In the Consensus Model, the information and activities defined are the building blocks necessary for a design rationale system.

7. Conclusion

In this paper a unified decision model has been developed and supported. Part of its usefulness is in the very definition of the terms commonly associated with engineering design, decision making, and negotiation. Further it is useful in its ability to help build understanding of how best practices affect the design process. Finally, it serves as a basis for the development of design rationale systems.

8. Bibliography

Banares-Alcatara, R. and Lababidi, H., KBDS: an Experimental Prototype, Computers and Chemical Engineering, Vol 19, 1995, pp279-301.

Boehm, BW and Ross R. 1989. Theory W Software Project Management: Principles and Examples. IEEE Trans. Software Engr. (July).

Boehm, BW, Bose P, Horowitz E, Lee M. 1995. Software Requirements Negotiation and Renegotiation aids: a theory-W based spiral approach. Proceedings of the 17th International Conference on Software Engineering (ICSE-17), IEEE Computer Society Press, Seattle.

Bose, P. 1995. A model for decision maintenance in the WinWin collaboration framework. KBSE Proceedings.

Conklin, J and Begeman M. gIBIS: a hypertext tool for exploratory policy discussion. ACM Trans Office Info Systems (Oct), pp. 303-331, 1988.

Herling, D., "An Engineering Decision Support System (EDSS) with Alternative-Criterion Pair Evaluations", Ph.D. Thesis, Department of Mechanical Engineering, Oregon State University, 1997.

Herling, D., Ullman, D.G., and B. D'Ambrosio, "Engineering Decision Support System (EDSS)", ASME 1995 Design Engineering Technical Conferences, Vol 2, 1995, pp 619-626.

Kuffner, T. and D.G. Ullman, "The Information Requests of Mechanical Design Engineers", Design Studies, 12(1), pp 42-50, 1991.

McKerlie, D and MacLean A. 1993. QOC in action: Using design rationale to support design. In Proceedings of ACM INTERCHI"93 Conference on Human Factors in Computing Systems.

Monk S., 1996, Design Rationale and PPIS, Proceeding of the 5th Message Understanding Conference, DARPA SISTO 1993, and http://www.comp.lancs.ac.uk/computing/PROTEUS/PPIS.html

Nagy, R.L. and D.G. Ullman, "A Data Representation for Collaborative Mechanical Design," Research in Engineering Design, Vol. 3, No. 4, 1992, pp. 233-242.

Olson, G.M. et. al., "Small Group Design Meetings: An analysis of Collaboration", Human-Computer Interaction, 1992, Vol 7, pp 347-374.

Olson, G.M. et.al., "The Structure of Activity During Design Meetings", in Design Rationale: Concepts, Techniques, and Use, Edited by Thomas P. Moran and John M. Carroll, Lawrence Erlbaum Associates, Publishers, Mahwah New Jersey, USA, 1996, pp 217-244.

Rittel, HW and Webber MM, "Dilemmas In a General Theory of Planning", Policy Sciences, 4, pp 155-169, 1973

Stauffer, L.A. and D.G. Ullman, "Fundamental Processes of Mechanical Designers Based on Empirical Data," Journal of Engineering Design, Vol. 2, No. 2, 1991, pp. 113-126.

Ullman, D. Issues Critical to the Development of Design History, Design Rationale and Design Intent Systems. Design Theory and Methodology 94, DE, 68:249-258

Ullman, D and D'Ambrosio B., "A Taxonomy for Classifying Engineering Decision Problems and Support Systems", Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, AI EDAM #9, pp. 427-438, 1995.

Ullman, D., Herling, D. and D'Ambrosio, B., "What to do Next: Using Problem Status to Determine Course of Action", Research in Engineering Design, #9, 1997, pp 214-227.

Ullman, D. 1997. The Mechanical Design Process, 2nd edition, McGraw Hill, 1997.

Yakemovic, K.C.B. and Conklin J. 1989. The Capture of Design Rationale on an Industrial Development Project: Preliminary Report Technical Report. STP-279-89 MCC.

Yakemovic, K.C.B. and Conklin J. 1990. Report on a Development Project Use of an Issue-based Information System: Technical Report STP-279-90 MCC.