The Space Congress® Proceedings

1988 (25th) Heritage - Dedication - Vision

Apr 1st, 8:00 AM

Design Knowledge Capture: Preserving Engineering Knowledge For Future Applications

R. B. Purves
Boeing Aerospace Company PO Box 1470, MS JA-68

James R. Carnes
Boeing AI Center PO Box 1470, MS JA-65 Huntsville, AL 35807

Scholarly Commons Citation

Purves, R. B. and Carnes, James R., "Design Knowledge Capture: Preserving Engineering Knowledge For Future Applications" (1988). *The Space Congress® Proceedings*. 4.
https://commons.erau.edu/space-congress-proceedings/proceedings-1988-25th/session-6/4

DESIGN KNOWLEDGE CAPTURE:
PRESERVING ENGINEERING KNOWLEDGE FOR FUTURE APPLICATIONS

R. Byron Purves
Boeing Aerospace Company
PO Box 1470, MS JA-68
Huntsville, AL 35807

James R. Carnes
Boeing AI Center
PO Box 1470, MS JA-65
Huntsville, AL 35807

ABSTRACT

A new problem beginning to challenge the artificial intelligence community is "design knowledge capture." There has been an increased desire to construct systems with inherent built-in evolvability toward more advanced technologies and machine intelligence. Success in this evolution process depends on being able to capture "as-built" design knowledge from the outset. Requirements and objectives for such systems are reviewed and an application to capture design knowledge within a combined data and knowledge environment is presented.

Design knowledge and rationale are implemented as design objects. Knowledge represented within an object can be physical, conceptual, functional, or structural. Storage and retrieval of rationale is achieved through a network of design objects and a model of the design process. The context, as well as the content, of captured knowledge is described within the process network of requirements, trades, and analyses. Design knowledge applications include, for example, traceability to requirements, standards, and specifications. It can be used to describe attributes of a part, or for input to further analyses, and trade studies. Other uses include verification, simulation, and maintenance activities.

1. INTRODUCTION

Design Knowledge Capture is a process used to acquire and store engineering knowledge generated during the many phases of a major program. This initiative specifies the collection of knowledge to a level of detail and frequency of capture that benefits information management in a program. The most visible benefits to a program are effective management of engineering change and increased product quality through a common platform for engineering design, analysis, test, manufacturing, etc. Although there are significant benefits from this process during a program, even stronger benefits come from the accumulated body of program knowledge, whose sum can be used to fuel countless knowledge-based applications managing life-cycle functions beyond delivery.

Design Knowledge is defined to include both the design object and the designer's knowledge, the "what" information and "why" information of a design. In addition to a description of the physical parameters of an object, it includes the rationale behind design decisions and results of analyses performed on the object. The objective of this process is to capture, organize, and retain design knowledge for the Space Station.[1,4]

The new capabilities offered within a design knowledge capture system result from two major trades:

(1) how much of the spectrum of traditional data generated during the engineering design process is available, or can be collected, in machine intelligible form, and

(2) what subset of mature tools and technologies can be harnessed to serve such a large task in an economical fashion.

In the traditional engineering design and development environment, a tremendous amount of "interesting" data is generated. This data covers a wide spectrum, with some elements managed quite well, some managed adequately for the job at hand, and still others managed ineffectively, or not at all. For example, contract documents, design drawings, etc. have been well-managed and do not present a problem in the DKC environment. Internal reports, coordination sheets, trade summaries, meeting minutes are filled with valuable data, but usually have no formal collection mechanism (they are usually filed manually, in paper form). At the other end of the spectrum, design note books and pitch charts often contain the real thinking and decision making process, but are never captured. Capturing this type of data can yield the largest increase in engineering capture, but this can be the most difficult data to capture.

Object-oriented systems respond well to the design knowledge storage requirements and provide a good environment to constrain Design Knowledge acquisition within the following three parameters:

- o Perspective. Design Knowledge is only captured in domains for which uses have been identified.
- o Visibility. The amount of knowledge captured depends on its importance.
- o Version. The frequency of capture will be limited to controlled knowledge acquisition.

The actual bounds on the total amount of knowledge to be captured are flexible and reflect the need for more summary information early and more detailed knowledge later in the program.[2,3]

This process must become integrated into, but not overshadow a program. Care must be exercised to establish the proper levels and frequencies for collection -- neither to acquire too little nor too much design knowledge. The justification for Design Knowledge Capture must include analysis to demonstrate that development of this AI technology (and its applications) is both wise and profitable.

Many papers have been written addressing the availability and quality of engineering data, as well as the different technologies required for its acquisition and storage. However, as was described earlier, there is more to knowledge capture than networks of design objects, each bursting with engineering data. For these networks to be truly useful for future application, each significant parameter must be qualified not only with current values, but with the "rationale" for selection. The bulk of the paper will focus on this topic within engineering knowledge capture systems: design rationale and its relationship to design knowledge.

2. RATIONALE CAPTURE

The major objective of rationale capture is to store an "explanation" of "how" or "why" a particular value has been assign to a design attribute. In order to be of use to any application software, the explanation must be captured in machine intelligible form. This process is a complex transformation from a set of requirements, standards, and engineering practice into a design (see Figure 1). During the development of the design knowledge base, several important aspects of the requirements-to-design transformation appear, indicating the potential use of design rationale in management applications:

A. The influence of individual requirements can be traced.

B. The requirements source and reasoning process behind each design feature can be traced.

C. The influence of proposed requirements changes on the design can be assessed.

D. The effects of changes in assumptions can be assessed.

E. The overall reasoning process can be reviewed for possible future improvements.

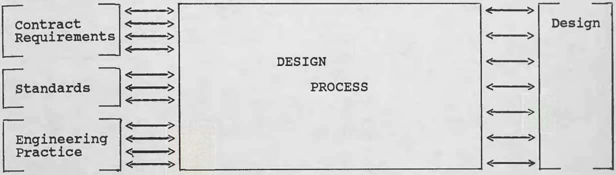


FIGURE 1: Simple Design Sequence.

3. BUILDING BLOCKS

The design process maps a sequence starting with a set of requirements and ending with a set of deliverables. There are, of course, several sets of internal milestones to synchronize the process. The following rationale elements form a core set which define the different building blocks used in the capture of engineering rationale. Figure 2 illustrates how these elements can be combined into a network documenting a subset of design. Others may be created to describe more specific aspects of the program or special design knowledge to be captured.

Formulate (F) a description of candidates (requirements, design concepts, operations concepts). This is not necessarily detailed engineering. Usually this activity is a prelude to an analysis, and as so must incorporate sufficient information to support the analysis. For example: describe a rack in sufficient detail that a weights analysis can be performed. Often this is a major component of the creative activity.

Analyze (A) a requirement, design concept, test concept, etc. Usually some tool or a standard analysis method is used. The output of analysis may be a value judgement (good/nogood) or a performance curve. For example: (1) analyze a pressure vessel for capability to withstand loads, or (2) analyze an operations concept to estimate the crew time required.

Trade (T) options against each other and make a decision (either elimination or selection). Usually a decision is based on formulations of multiple options which have been analyzed. For example: (1) decide between waffle-grid and skin-and-stringer construction, or (2) decide on power budgets for racks in a module.

Describe (D), or add detail to formulated requirement or design concept. This process uses little creativity, and is usually involves design-to-standard engineering practice. The culmination may be a drawing release. For example: (1) given a composite rack concept, produce/release the design drawings, or (2) given a waffle grid concept, dimensions, etc., produce and release the design drawings.

Evaluate (E), or test, a build object. For example: (1) test a computer program configuration item to determine compliance with allocated requirements, or (2) test a rack for satisfaction of EMI/EMC requirements.

Build (B) an article to design specification and manufacturing standards. For example: code software to detail design specification.

Simulate (S), or build, a mockup to assess a formulated concept or design. This role is similar to analysis. For example: build a workstation mockup to collect data for determination of best placement of video screens.

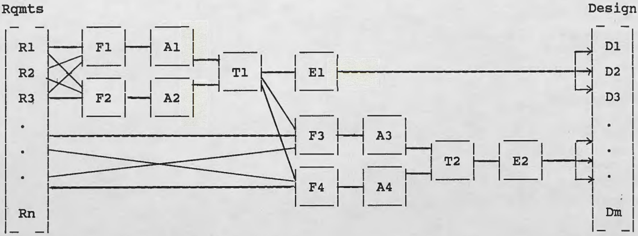


FIGURE 2: SIMPLE DESIGN SEQUENCE.

4. DESIGN KNOWLEDGE ENVIRONMENT

Inside each rationale element is a specification of the engineering design process. An expansion of the rationale node may yield more specific design knowledge in a capture network, or more specific design information in data flow diagrams that occur below the visible rationale capture level. Information within these data nodes is collected implicitly as part of the day-to-day design and analysis activity.

The example in Figure 3 shows an expansion of two processes from Figure 2. Coming into node F4 are data values [a b ... e] which are processed through a small decision network with subnodes f, g, and h, yielding data values [i j ... n]. The subnodes f, g, and h can be more concrete engineering processes or additional design knowledge capture points (which would contain further subnodes).

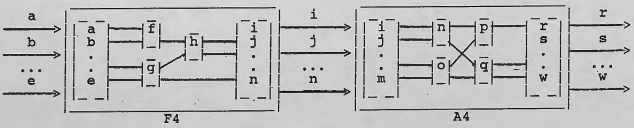


FIGURE 3: SAMPLE EXPANSION OF A KNOWLEDGE PROCESS.

A Design Knowledge Management Tool has been developed to support graphical query of design and rationale from multi-level networks with full perspective (see Figure 4). Forward and backward relationships are be elicited. All information is ultimately translated into fundamental data relationships and stored within the a relational database for ease of delivery. The relational database also provides the basic capabilities for version control and relation locking.

The engineering process is improved through the use of an intelligent interface to database and tools, driven by graphical displays and menus. The graphical presentation of the design process provides better access to data, so managers and engineers can get a deeper understanding of design problems in reduced time. Users always have complete access to rationale and information associated with design nodes.

The Management Tool initially supports the engineering process at higher levels. Design engineers must complete an additional task, manually entering rationale as design decisions are made at lower levels. Eventually the interface to the design knowledge data base will be driven through intelligent interface that queries the user based on existing information.

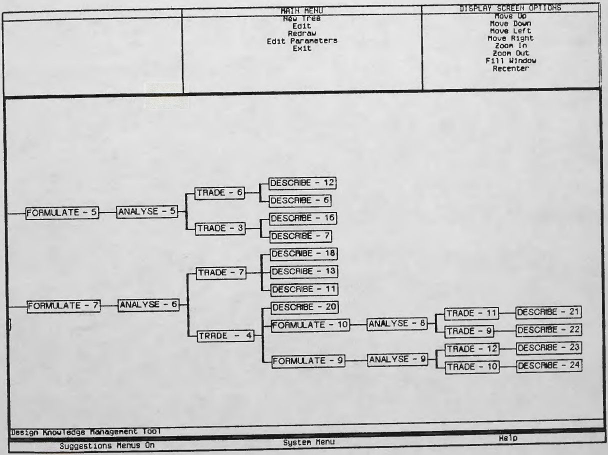


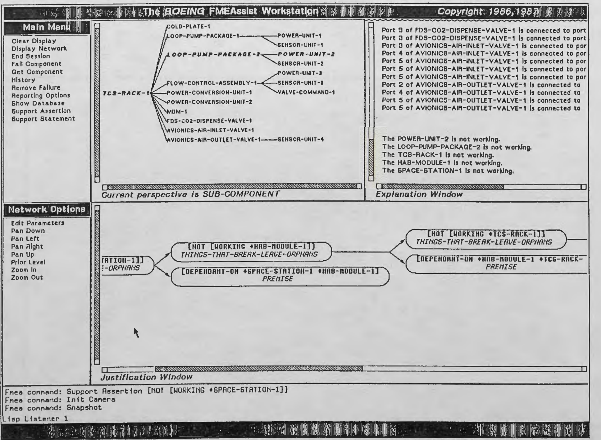
FIGURE 4: DESIGN KNOWLEDGE MANAGEMENT TOOL

5. FUTURE APPLICATIONS

Benefits that will be realized are both immediate and far-reaching. Near-term benefits will occur in the areas of design review, system integration, manufacturing, operations, maintenance, sustaining engineering, and logistics. Long-range benefits include enabling applications of artificial intelligence and robotics on the Space Station by providing knowledge needed for knowledge-based systems. Tools to be built in this environment have been divided into three model-based areas for study and development: process model only, design model only, and both process and design models.[2,3]

Process model applications build links, adding structure to the process network. The design knowledge management tool is primarily a process tool. It represents the design as a network of discrete design events. Other tools under consideration in this area include:

- Requirements Shredder
- Reports and Documents
- Engineering Change Tool



```
Port 3 of FDS-CO2-DISPENSE-VALVE-1 is connected to port
Port 3 of FDS-CO2-DISPENSE-VALVE-1 is connected to port
Port 3 of AVIONICS-AIR-INLET-VALVE-1 is connected to por
Port 4 of AVIONICS-AIR-INLET-VALVE-1 is connected to por
Port 5 of AVIONICS-AIR-INLET-VALVE-1 is connected to por
Port 5 of AVIONICS-AIR-INLET-VALVE-1 is connected to por
Port 5 of AVIONICS-AIR-INLET-VALVE-1 is connected to por
Port 2 of AVIONICS-AIR-OUTLET-VALVE-1 is connected to
Port 4 of AVIONICS-AIR-OUTLET-VALVE-1 is connected to
Port 5 of AVIONICS-AIR-OUTLET-VALVE-1 is connected to
Port 5 of AVIONICS-AIR-OUTLET-VALVE-1 is connected to


The POWER-UNIT-2 is not working.
The LOOP-PUMP-PACKAGE-2 is not working.
The TCS-RACK-1 is not working.
The HAB-MODULE-1 is not working.
The SPACE-STATION-1 is not working.
```

```
Fmea command: Support Assertion [NOT [WORKING +SPACE-STATION-1]]
Fmea command: Init Camera
Fmea command: Snapshot
Lisp Listener 1
```

FIGURE 5: FAILURE MODE ANALYSIS TOOL

Design model applications build subnodes, adding depth to the process network. One tool in this area that has been well-developed is a Failure Modes and Effects Analysis Assistant (FMEAssist). FMEAssist aids the engineer in populating design objects with engineering data about potential failures (see Figure 5). Design model-based tools will be developed for engineering analysis applications, both before and after delivery. There are many possible applications in this area, including:

- Functional Model Builder
- Application Generators
- Logistics Tool
- Reliability Tool

Most complex tools will be built on both models. Planning applications are perhaps the most common of these complex tools, presently under consideration. These tools traverse the networks of specified nodes and links, searching both the breadth and depth of a network. Many planning applications are under consideration, including.

- Test Plan Construction Tool
- Resource Allocation Tool
- Operations Planning
- Experiment Planning
- Mission Planning

REFERENCES

1 Conklin, J., Hypertext: An Introduction and Survey, IEEE Computer, pp. 17-41, September, 1987.

2. NASA JSC 30741, Process Requirements for Design Knowledge Capture, November 12, 1986.

3. NASA TM 89190, Advancing Automation and Robotics Technology for the Space Station, Advanced Technology Advisory Committee (ATAC), Progress Report Three, October 1, 1986, pp. 27-28.

4. Weinreb, D. Statice: An Object-Oriented Database System, Position Paper, OOPSLA-87, October 1987.