

ADDVAC: Applying Active Design Documents for the Capture, Retrieval and Use of Rationale During Offshore Platform VAC Design

Ana Cristina Bicharra Garcia, Joper Cezar de Andrade,
Rogério Ferreira Rodrigues and Ricardo Moura

ADDLabs - Departamento de Ciência da Computação
Universidade Federal Fluminense, Praça do Valonguinho, s/n
Niterói, RJ, 24210-130, Brazil
E-mail: bicharra@dcc.uff.br

Abstract

Engineering design is a plan to build an artifact that needs to be communicated and discussed among designers. The concept is an abstract idea that can only be transmitted through a document containing the description of the artifact's form, function and behavior as well as the designer's rationale for them. Most engineering design domains are multidisciplinary. Understanding design decisions and their rationale becomes crucial to allow collaborative work. The Active Design Documents (ADD) approach has been very effective in assisting the capture, retrieval and use of rationale during the design process in various engineering domains. This paper presents the development and results of ADD applied to the domain of Ventilation and Air Conditioning (VAC) design of offshore oil production platforms. We focus our discussion on the way rationale availability improved the design process itself and on the problems encountered during development.

Introduction

An engineering artifact represents an answer given by one or more designers to a set of needs identified in a specific environment, restricted by physical laws and evaluated by economical laws. This only happens when there is an owner who pays for the development, construction and maintenance of the designers' solutions. On the other hand, this owner demands designers to develop a high quality, low cost solution, in a short time schedule.

The artifact lifecycle can be roughly divided in three main phases: design, construction, operation/maintenance. Although our research focuses on the design phase, we considered the information needs during later phases.

The design process involves people from different domain areas working in portions of the overall design solution. In spite of having a common global goal, conflicts often arise from the optimized local solutions. Frequently, the conflicts remain due to the lack of knowledge on the other designer's decisions criteria. Conflicts are solved based on the understanding of each

others' rationale and of what leads to the global optimum solution [Peña-Mora, 95]. Design rationale is valuable information to have available not only during design, but also in the remaining phases to minimize rework. Although the artifact design is the main theme of messages exchanged by team participants, perceiving design intent plays a major role to ensure truly cooperative behavior among design participants.

An efficient and practical environment that allows design agents to encode their own ideas and decode those of the others has been the focus of much research in the area of design rationale. Powerful design documentation tools have been proposed [Kuokka, 1994] [Favela, 1993] [Klein, 1992] [Conklin, 1988] [Bradley, 1991] among which is the Active Design Document — ADD [Garcia'92]. ADD is a dynamic computational representation of design rationale based on an adjustable underlying model of the device conception process. The system captures and delivers design rationale thus providing the necessary design communication infrastructure. The system becomes the medium; the interface the language and code; and the computer, the channel.

In this paper, we present an implemented version of active design documents developed for the domain of Ventilation and Air Conditioning design of offshore oil production platforms (ADDVAC). The results are discussed considering the impacts on the design process and product. We used the number of explored alternative solutions, the development time and the final cost as the metrics for our evaluation.

The VAC Design Problem

Off-shore platform design involves a number of different knowledge areas such as Metallic Structures, Electricity, VAC (Ventilation and Air Conditioning), and Architecture

disciplines. The design team is in charge of developing the conceptual design for offshore oil production platforms. The conceptual projects are further used as the basis for the whole bidding process. Production field data are the input for initializing the conceptual design phase.

However, field data are not static. As the exploration process evolves, new data keeps coming into the design process, making it a totally dynamic activity. Moreover, the highly concurrent sub-tasks cannot be held isolated from each other. Decisions in one, typically affect the other(s). An example of this situation is a need to decrease platform space, experienced by architects, leading to different solutions for the VAC systems. Additionally, the whole conceptual design process is a time-critical one, which often leads to insufficient time to explore different design alternatives. In general it works as an one-shot decision.

The design of an VAC system for an offshore oil platform consists of defining:

- for each room, the type of inlet/outlet air (natural, mechanical ventilation, air conditioning, return, or exhaust), the cooling/heating loads, and the air flow;
- the group of rooms that can be served by the same equipment;
- the equipment specifications;
- the location of each equipment;
- the rooms air flow balance;
- the total cost, weight and power consumption.

VAC system designers generate design documents to communicate their concepts. Especially during preliminary design, documents are used for 2-way communication. There are many tradeoffs in design between cost, weight, function, etc.; also many assumptions and preferences still need to be articulated. Thus, a lot of negotiation goes on during the conceptual design.

ADDVAC: Using Active Documents for the VAC Design of Offshore Oil Platforms

ADD has proven to be very effective for domains represented by a parametric model considering rational decision-making agents [Mittal & Frayman, 1989]. We can make the following assumptions for these domains:

- decisions can be mapped to the assignment of parameters' values;
- a decision on a parameter value may impact other decisions;

- restrictions are boundaries on the behavior, form, and function of the designed artifact;
- criteria represent preferences ordering the alternative values.

In addition to the assumptions on the domain representation, the rational decision-making expected in engineering design consists of:

- generating alternative parameters' values;
- restricting the possible values;
- evaluating and ordering the alternatives according to a set of criteria
- selecting the best ranked parameter value.

A decision on a parameter value is either made by the user and verified by ADD or suggested by ADD. In both cases, ADDVAC generates its own expectation on the solution. In case the user suggests something different than the expected, ADD enters in a Knowledge Acquisition phase. It assumes this point as revealing conflicts between the computational and the human design agents. The user can accept ADDVAC's suggestion, force his or her value or change the knowledge base to lead to an agreement. This behavior characterizes an active design document; i.e., it assists design activity while capturing rationale to later be able to generate explanations following a model that represents the designer.

The Parametric Domain Model

A project is developed by a designer (the active agent) in phases, known as versions, which are composed by alternatives. ADDVAC reflects this procedure in the way of organizing its data and knowledge. The versions and alternatives are organized as a tree structure. Each node in this tree contains the context in which a set of decisions were made and evaluated. Any project is composed by one or more versions, that are disposed in a sequential way, showing the project evolution. However, only the last version can be edited. The previous ones are available only to be consulted. A new alternative is generated from an existing one. An alternative can have an unlimited number of derived alternatives. However, once an alternative derives another, it can no longer be changed. Therefore, the designer is always working on the alternatives of the most recent version of a project. This implementation strategy allows each alternative to contain the history associated with all design actions/decisions related to it.

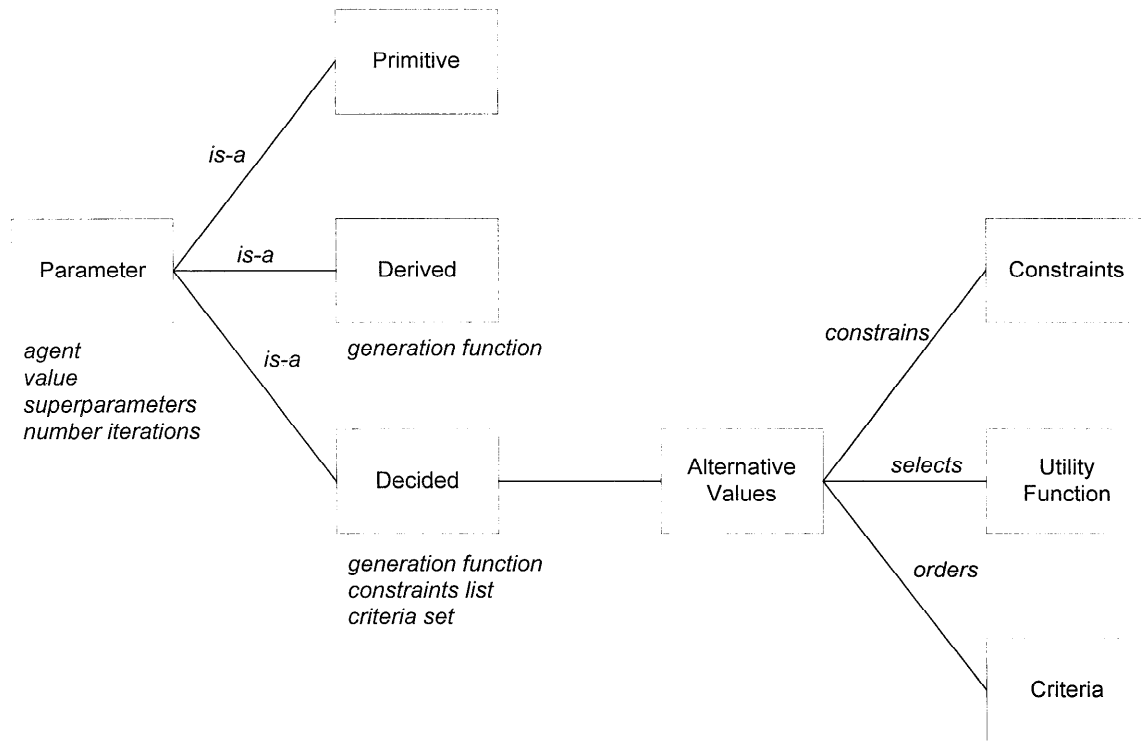


Figure 1: Design Domain Parametric Schematic Model

ADDVAC's Decision-Making Process

The decision-making procedure follows the following steps:

1—Generate a set of alternatives.

The idea is to specify a valid domain for that parameter. This process may involve database retrieval or calculations. Alternatives can also be pre-defined values. For example, when an engineer specifies an engine to provide air conditioning for all platform rooms, he may define different compartment groupings.

Whenever lack of enough information is detected diagnosing an inefficiency in using first-principles or even heuristic methods, case-based reasoning was used. Consequently, there are always alternative values to be considered. We have a Database (DB) of previous cases indexed by selected relevant parameters.

2—Eliminate alternatives due to constraints.

After generating alternatives, ADDVAC evaluates them according to active constraints, representing physical, normative, or institutional limitations on values. An alternative either satisfies or not a constraint (no fuzzy logic was used).

For example, a new oil production unit could be built on a ship, a fixed platform or a floating platform. Assuming a designer wants the list of VAC equipment for this new

unit, ADDVAC can use previous cases to generate such a list. It only includes cases with production unit type.

3—Score each alternative

Valid alternatives must be ordered according to the set of active criteria. Each alternative value receives a score according to each criterion evaluation.

Continuing the previous example, a designer would prefer to use a equipment list from a the most similar platform oil field characteristics, crew, and oil production.

4—Select the best alternatives

After ordering the alternatives, ADDVAC selects the best one; i.e., the one with the best overall score. We use a linear utility function to combine the criteria evaluation, as presented below.

$$FG_j = \sum_i W_i \cdot S_j$$

where (i) is the index of each criterion, (j) is the index of each alternative, W_i is the weight of each criterion and S_j is the score of each alternative according to step number 3.

ADDVAC's Interface Design

The ADDVAC interface objective is to assist the designers to develop a VAC project. Since a VAC project is based on a platform layout, it is important to know the dimensions, place of installation, type of compartment (galley, cabin, etc.) involved, so that the knowledge/rules may be used during the decisions phase. For this reason, a

computer aided design system (CAD) was included as part of the design interface. Although the drawing requirements for this CAD were not complex, the reasoning over the drawings were. For example, any compartments were represented as either rectangles or L-shaped objects, but they have, many different possible functions and were treated using a 3-D reasoning strategy.

After the input of the platform layout, the designer can enter in the main activity; i.e., deciding on the parameters that configured the VAC system. These decisions can be made at any time and in any order. Whenever a decision needs the results of a previous one, the system makes that decision, without any designer interference.

Figure 2 presents a screen dump of the ADDVAC design interface. In the lower part of the interface, there is a status bar with all the necessary information always available (compartment name, air flow, cooling load, type

of compartment, etc.). In addition to creating, deleting and modifying compartments, ADDVAC's multimodal design interface offers all zooming and positioning capabilities of most commercial drawing tools.

In spite of having a sequential arrangement, our case study has shown that designers make decisions in a randomized order and iterate many times. The system automatically checks the decisions and propagates the impacts to the entire project.

ADDVAC Rationale Acquisition

The knowledge acquisition interface allows designers to modify values and restrictions of ADDVAC's design knowledge and data bases. The changes on the knowledge bases are valid for the current project and become available for all future ones.

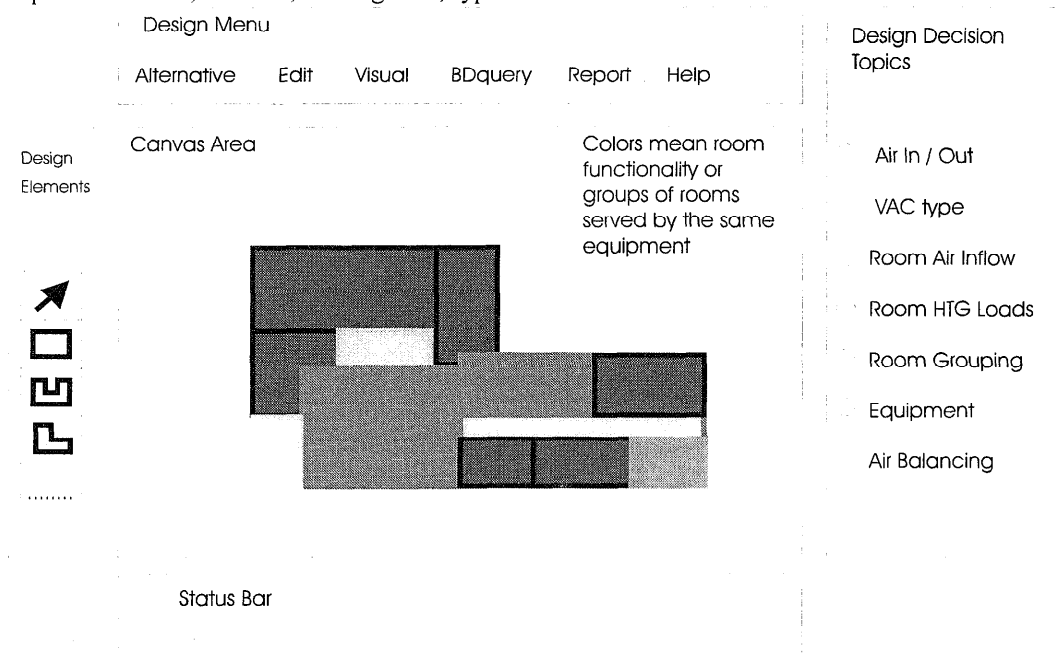


Figure 2: ADDVAC's Design Interface schema

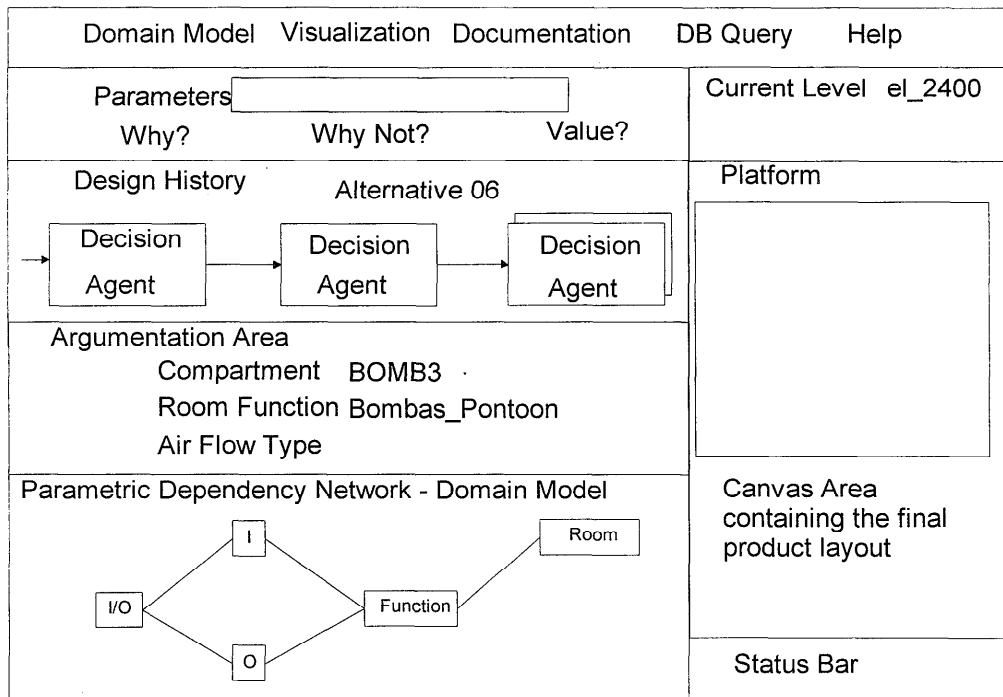


Figure 3 : ADDVAC Explanation Interface

Conflicting changes resulted from different designers interacting with the tool needs to be integrated in a manual manner. ADDVAC does not offer yet any mechanism to deal with merging different knowledge bases. A designer can choose from which knowledge base he or she will start working.

ADDVAC Rationale Retrieval Interface

The main objective of ADDVAC's interface is to provide an environment in which a VAC designer and any VAC documentation user can retrieve explanations (rationale) over a conceived project. Design rationale assists users to understand a project, in order to criticize or to learn from it.

The explanation interface user formulates queries using a combination of menu selection, and graphical object selection. The explanation interface allows 3 types of questions over the decisions: "WHAT?", "WHY?" and "WHY NOT?". Additionally, the user can probe the model and engage in a "WHAT IF" type of question.

The answer is presented as a series of displays containing: the design HISTORY, the designed PRODUCT (platform layout), the decision local ARGUMENTation and the DOMAIN MODEL, as illustrated in Figure 3.

The HISTORY is a text, automatically generated, presenting the designer's actions sequence up to the query parameter focus. To simplify the visualization of this text, the decisions were represented as boxes, in which only

basic information (type of decision, agent, and if there were any restrictions violations) is shown. The user obtains further information, such as a designer attached note, through direct graphic manipulation. The decision sequence indicates the designer's strategy. Recognizing decisions sequence patterns to provide interpretation to the flat information offered by the history is the focus of our current research. For example, in Figure 3 a cascade box indicates a sequence of repeated decisions applied to different objects. We are organizing the HISTORY information using a hypertext structure that enriches user interaction.

The ARGUMENT screen presents a table containing the local evaluation of alternatives for the probed parameter. This is a read only display that may be associated with a canned text that rhetorically organizes the designer's discussion over a parameter value.

The DOMAIN MODEL screen displays the dependency graph among the parameters. The total domain dependency graph is presented only in the case the user probes the design model. Generally, a partial view is shown, in which the probed parameter defined the portion that should be presented.

The PRODUCT screen displays the status of the project in a given version and alternatives. It offers zooming and deitic features. It is the major display to present answers to WHAT questions.

Implementation Aspects

The system was developed using multiple platforms of hardware and software:

Hardware:

- IBM RISC 6000™, hosting most of the program and where the program runs;
- Intergraph server 6700, hosting the database.

Software

- PROKAPPA™ for AIX™, Intellicorp, to develop the knowledge base;
- C language to develop the ADDVAC's decision-making reasoning;
- IUP and GKS libraries, PUC-Rio, to develop the graphical interfaces; and
- Oracle™ version 7.0 to record data from (1) previous cases and (2) equipment catalogs information.

Discussion

Active Design Documents have affected the way designers develop, document and use their projects. Case studies using ADDVAC have shown impacts on design process and product quality [Garcia, 95].

The impacts of using ADDVAC on the design process were objectively measured in terms of VAC systems development time and number of explored alternatives. Even though we did not use any objective metrics, we noticed a better design understanding among the development team.

The initial results from 5 VAC platform design indicates:

- a substantial decrease on the project's development time, from months to weeks;
- a decrease on the total VAC project cost; about 80 % of savings;
- an increase on the number of analyzed alternatives; and
- a decrease on the documentation inconsistencies.

We consider a recently developed project using a equally qualified VAC designer. His solution without ADDVAC was compared with the one developed within the ADDVAC environment. For that specific VAC design case, for a 100,000 BOPD oil platform, the reduction was about 800 man-hours. This benefit would be much greater, if the active design document were developed for a critical path discipline.

The most important benefit, however, was the exploration of six alternatives, instead of the usual two, within a reduced time period, which lead to the choice of an optimized solution. The total cost of the final solution presented a cost reduction of US\$ 300,000.00 from the US\$ 3,400,000.00 estimated budget.

In addition, ADDVAC has allowed users to create, and spread the company's standards and patterns. Automating

routine tasks frees designers to explore more design alternatives leading to optimum design, as well as to try creative solutions. The benefits of using ADDVAC have stimulated designers to use it as an assistant design tool.

References

Bradley, S., & Agogino, A (1991). Design Capture and Information Management for Concurrent Design. *International Journal on System Automation. Research and Application journal*, 1(2), pp. 117-141, 1991.

Conklin, J. & Begeman, M. (1988). gIBIS: A Hypertext Tool for Exploratory Policy Discussion. *ACM Transaction on Office Information System journal*, 6(4), 303-331, 1988.

Favela, J., Wong, A., & Chakravarthy, A. (1993). Supporting Collaborative Engineering Design. *Engrg. with Computers journal*, 4(9), pp. 125-132, 1993.

Fischer, G., McCall, R., & Morch, A. (1989). Design Environment for Constructive and Argumentative Design. In the Proceedings of Computer Human Interaction CHI '89, pp. 269-276, 1989.

Garcia, A. C. B., Stefik, M. J. & Howard, C. H. (1992). Acquiring Design Knowledge through Design Decision Justification. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing journal*, 6(1), pp. 59-71, 1992.

Garcia, A. C. B. (1995). ADDVAC Documentation System Manual. PETROBRAS technical report, 1995.

Klein, M. (1992). DRCS: An Integrated System for Capture of Designs and their Rationale. In the Proceedings of the Artificial Intelligence in Design '92 (Gero, J., Ed.), pp. 393-412, Kluwer, London, England.

Kuokka, D., Jefferson, S., Barford, L. & Frayman, F. (1995). A Parametric Design Assistant for Concurrent Engineering. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing journal*, 9(2), pp. 135-144.

Mittal, S., & Frayman, F. (1989). Towards a Generic Model of Configuration Tasks. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, pp. 1395-1401. San Mateo, CA: Morgan Kaufmann.

Peña-Mora, F., Sriram, R. D. & Logcher, R. (1995). Conflict mitigation system for collaborative engineering. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 9(2), pp. 101-124.