

CONF-840614--28

AUTOMATED REASONING APPLICATIONS TO DESIGN ^{Analysis} ~~VALIDATION AND~~
~~SNEAK FUNCTION ANALYSIS~~

CONF-840614--28

DE84 006413

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

by

R. C. Stratton

EBR-II Project
Argonne National Laboratory
P.O. Box 2528
Idaho Falls, Idaho 83401

The submitted manuscript has been authored by a contractor of the U. S. Government under contract No. W-31-109-ENG-38. Accordingly, the U. S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U. S. Government purposes.

Submitted for Presentation
at the
American Nuclear Society
1984 Annual Meeting
June 3-8, 1984
New Orleans, Louisiana

MASTER

* Work Supported by the U.S. Department of Energy
under Contract W-31-109-38.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

mp

INTRODUCTION

Argonne National Laboratory (ANL) is actively involved in the LMFBR Man-Machine Integration (MMI) Safety Program. The objective of this program is to enhance the operational safety and reliability of fast-breeder reactors by optimum integration of men and machines through the application of human factors principles and control engineering to the design, operation, and the control environment, Vaurio et al (1982). One aspect of the MMI work at ANL, and its implications, is reported in this paper.

Validating that the system design function incarnation (the physical product of design) matches the functional requirements and identifying sneak functions (unplanned element functions) are necessary for ensuring that a system meets reliability criteria, Wojcik (1983) and Ehrlich (1983). Classically, validation and sneak function analysis are performed by the design engineer. The effectiveness of this validation and analysis is directly proportional to the engineer's training, experience, and thoroughness. Generally, the better trained and the more experienced the engineer, the more functional the design and hence, the better the transition to the physical incarnation with reduced or eliminated sneak functions.

ANL is developing methods to apply automated reasoning and computerization in the validation and sneak function analysis process. If the proper relationships and definitions of design functions and components are provided, Stratton et al (1983) and Gabriel (1983), then validation of system incarnation and sneak function analysis can be achieved via automated reasoners (AR) such as Logic Machine Architecture, Lusk et al (1982), and Prolog, Clocksin et al (1981) and Kowalski (1982).

This project develops the element definitions and relations necessary for an automated reasoner (AR) to reason about design validation and sneak function analysis. This project also provides a demonstration of this AR application on an Experimental Breeder Reactor-II (EBR-II) system, the Argonne Cooling System (ACS). The initial demonstration will be limited to one of the subfunctions provided by the ACS, the reactor subassembly cooling function. This will be discussed in greater detail in later sections dealing with performing validation and sneak function analysis.

ACS System Description

The Argonne Cooling System (ACS) performs a subfunction for the Fuel Handling System (FHS). The FHS functions to transfer nonirradiated fuel into the reactor core and irradiated fuel from the core to the fuel subassembly shipping cask. The ACS principally functions to provide thermodynamic integrity of the subassembly when it is external to the reactor pool. The ACS circulates argon gas through fueled subassemblies for preheating (during transfers into the primary tank) or cooling (during transfers out of the primary tank).

Functional considerations and constraints that must be understood and accounted for in the design of the ACS are as follows. Subassemblies are either irradiated or nonirradiated. Irradiated subassemblies are radioactive, generate decay heat, and contain radioactive sodium residue. Nonirradiated subassemblies require preheating prior to immersion into the reactor pool sodium environment. The refueling environment contains sodium which is radioactive, highly reactive, and will foul and plateout on system components. All components that come into contact with sodium must subsequently be cleaned of sodium residue. The purity of the primary pool argon cover gas environment must be maintained. Argon must not be released to the atmosphere external to the FHS. The integrity of all interfacing processes must be maintained.

The above functions and constraints must be included in the design specification. The design specification will express the essential functions and incarnations required to provide the process. The functions will be of two classes, design functions and technical functions. The manifestations of the design specification are the physical components and structure of the ACS.

RELATIONSHIPS AND DEFINITIONS

Element relationships and definitions are required to define the system model and rules for the automated reasoner. Elements are the physical entities that comprise the designed process and are defined as components, subsystems, systems, and the process. The relationships and definitions are manifested as element functional definitions, state relationship to functions, function relationship to direction, element connectivity, and functional hierarchical configuration. "State" defines the elements condition of existence, Seeman et al (1982), Colley (1982), and Henley et al (1981). These relationships and definitions are further defined below.

Justification for elements in a system is that elements satisfy functions. Each element, therefore, represents single or multiple functions. These functions exist in a hierarchical network of functions that satisfy a required process. The elements can be characterized as providing functions at required design capacity or limits. This paper deals with the functional aspect of the element. Later research will incorporate design capacity.

Element state has a direct relationship to element function. Therefore the element function is further defined and bounded by the state of the element. An element with multiple functions is capable of providing a defined function only when it resides in a specific state and the element may provide certain functions when in one state while providing other functions in yet another state.

In addition to element functions being bounded by the state, the functions are also bounded by medium flow direction. The direction of flow through an element can alter the element's functional characteristics.

The alteration can either be neutral, negative or positive relative to the element's function.

Element connectivity characterizes the physical connection between elements. Connectivity defines the serial and parallel functions of the process as well as the function integration and provides the mechanism that allows function evaluation of the integrated elements for comparison with the design function requirements.

The functional relationships necessary to provide for the overall (top level) design function are described in what is called a "functional hierarchical configuration." The top level design function is the highest node in the functional hierarchy. Each successively lower level of the hierarchy defines functions required to build the top level function.

Validation and Sneak Function Analysis

Validation is the act, process, or instance of assuring the compliance of an object to a standard. "Validation," as used herein, is used to mean the process that consists of verifying that the incarnation of a design functionally complies with the design specification. Therefore, the product of this validation ensures that, at a minimum, the design incarnation provides for the functions designated in the design specification. Sneak function analysis is the reasoning about an incarnation to determine unplanned element functions that may result.

Automated Reasoner Algorithms

AR algorithms in this project provide for the generation and association of element parameters (state, function, and type) and functional hierarchies. AR algorithms also provide for validation and sneak function analysis. These algorithms are divided in the following categories; knowledge base, path, path function, validation, and sneak function analysis.

Performing Validation and Sneak Function Analysis

The procedure for performing validation and sneak function analysis for an actual design is as follows. Specify and represent the design functions in a hierarchy based on physics, engineering principles, objectives, and constraints. Represent the incarnation of the design as an extension of the design functional hierarchy. Define each element of the incarnation with respect to its functional attributes and interelement relationships. Then build an incarnated functional hierarchy using the graph representation of the design and the AR application algorithms. And finally, compare and analyze the incarnated hierarchy with the design hierarchy to determine validation properties and sneak functions.

DISCUSSION

The relations, definitions, and methodologies developed in this project potentially have wide application in addition to validation and sneak function analysis. These relations, definitions, and methodologies can be applied to design, operation, and training. Transformation of a design concept for a process into a design specification is defined and bounded by physics laws, engineering principles, objectives, constraints, and technology. When presented in a functional hierarchical construction, the functions tend to layer themselves from top to bottom in the order of laws and principles, objectives, constraints, and technology. That is, the hierarchy tends to place the essential function toward the top and the physical functions towards the bottom. These hierarchies can be developed for specific processes by experts in that process field. The expert process hierarchy can then be utilized by less experienced engineers and consultant aids in future design of similar and same processes. These hierarchies can also be utilized to document the physical incarnation functionally and to analyze the effect of future design changes. The hierarchy and path characteristics can be used to help develop normal and off-normal operations procedures for the process and can be used to analyze and process off-normal situations both diagnostically and prognostically. As a training aid, the functional hierarchy would explain and reinforce the functional aspects of the components and higher elements to the trainee. The path and pathfunction characteristics will help develop the trainee's ability to functionally analyze physical paths within the design incarnation, P&ID. That is, it will allow the trainee (either operator or engineer) to learn, review, and test the system concept of function and incarnation relationships.

Two other areas of potential application of these methodologies are fault tree analysis and alarm handling. Fault tree analysis requires the determination of all singular and integrated faults derived from electrical and mechanical designs. The design can be transformed into a graph consisting of arcs that represent the design elements, where each element has a defined number of states. Algorithms would then determine paths and path combinations associated with faulted elements states and conclude with a fault tree for the design in question. Alarms represent states of process element, i.e. systems, subsystems, and components. The matrix of alarms that result from a process fault is a function of the fault and the process elements associated with the fault. By defining the fault/ component/function relationship, alarm states can be mapped into a graph representing alarm relations and hierarchy. These graphs can then be analyzed by "path" and "path function" algorithms to determine the hierarchical fault relationships during a giving alarm situation.

CONCLUSION

Given the necessary relationships and definitions of design functions and components, validation of system incarnation (the physical product of design) and sneak function analysis can be achieved via automated reasoners. The relationships and definitions must define the design specification and incarnation functionally. For the design specification,

the hierarchical functional representation is based on physics and engineering principles and bounded by design objectives and constraints. The relationships and definitions of the design incarnation are manifested as element functional definitions, state relationship to functions, functional relationship to direction, element connectivity, and functional hierarchical configuration.

REFERENCES

1. Clocksin, W. F. and Mellish, C. S., "Programming In Prolog," ISBN3-540-11046-1 Springer-Verlag Berlin Heidelberg, New York, (1981).
2. Colley, R. W., "A Transition Control System for Procedure Prompting," 1982 ANS Summer Meeting, ISSN: 0003-018X, Vol. 41, p. 529, (1982).
3. Ehrlich, S. M. and Gabriel, J. R., "Cutsets With Required Arcs: A Prolog-Based Approach," ANL/MCS-TM-11, (1983).
4. Gabriel, J. R., "Algorithms for Automated Diagnosis of Faults in Physical Plant," ANL-83-70, (1983).
5. Henley, J. E. and Kumamoto, H., "Reliability Engineering and Risk Assessment," Prentice-Hall, Inc., 1981, pg. 80-89.
6. Kowalski, R., "Logic for Problem Solving," Elsevier North Holland, Inc. (1982).
7. Lusk, E. L., and Overbeek, R. A., "An LMA-Based Theorem Prover," ANL-82-75, (1982).
8. Seeman, S. E., Colley, R. W., and Stratton, R. C., "Optimization of the Man-Machine Interface for LMFBRs," Nuclear Safety, Vol. 24-4, pp. 506-512 (1983).
9. Stratton, R. C. and Lusk, E. L., "Automated Reasoning in Man/Machine Control Systems," 1983 ANS Winter Meeting, ISSN: 0003-018X, Vol. 45, P. 204 (1983).
10. Vaurio, J. K., et al, "Man-Machine Interface Program Plan," LMFBR Safety Program, Fast Reactor Safety Technology Management Center, Argonne National Laboratory, August 1982.
11. Wojcik, A. S. "Formal Design Verification of Digital Systems," 20th Design Automation Conference, 0738-100X/83/0000/022B, 1983 IEEE, p. 228-231 (1983).