DART - For Design Basis Justification & Safety Related Information Management

A. Billington[1], P. Blondiaux[1], J. Boucau[1] , B. Cantineau[1], A. Mared[2]

ABSTRACT

DART is the acronym for Design Analysis Re-Engineering Tool. It embodies a systematic and integrated approach to NPP safety re-assessment and configuration management, that makes use of Reverse Failure Mode and Effect Analysis in conjunction with a state-of-the-art relational database and a standardized data format, to permit long-term management of plant safety related information. The plant design is reviewed in a step-by-step logical fashion by constructing fault trees that identify the link between undesired consequences and their causes. Each failure cause identified in a fault tree is addressed by defining functional requirements, which are in turn addressed by documenting the specific manner in which the plant complies with the requirement. The database can then be used to generate up-to-date plant safety related documents, including: SAR, Systems Descriptions, Technical Specifications and plant procedures. The approach is open-minded by nature and therefore is not regulatory driven, however the plant licensing basis will also be reviewed and documented within the same database such that a Regulatory Conformance Program may be integrated with the other safety documentation.

This methodology can thus reconstitute the plant design bases in a comprehensive and systematic way, while allowing to uncover weaknesses in design. The original feature of the DART methodology is that it links all the safety related documents together, facilitating the evaluation of the safety impact resulting from any plant modification. Due to its capability to retrieve the basic justifications of the plant design, it is also a useful tool for training the young generation of plant personnel. The DART methodology has been developed for application to units 2, 3 and 4 at Vattenfall's Ringhals site in Sweden. It may be applied to any nuclear power plant or industrial facility where public safety is a concern.

[1] Westinghouse Electric Europe, Boulevard Paepsem 20, 1070 Brussels, Belgium
[2] Vattenfall DART Project Manager, Ringhals Nuclear Power Plant, S-430 22 Väröbacka, Sweden

INTRODUCTION

During the forty or so years since their first commercial introduction, the requirements for the design and operation of nuclear power plants have undergone tremendous evolution, responding to experience gained from operational and accident situations, and reflecting advances in scientific knowledge and engineering techniques.

In recent years, more and more emphasis is placed on operating NPPs to demonstrate, via traceable and up-to-date documentation, that they continue to meet the currently accepted safety standards, in the way they are designed, operated and maintained.

Typically, the relevant plant data are contained in separate, cross-referenced paper documents, which may be owned and maintained by different functional departments within a utility's organization. Managing all these documents, ensuring their coherence and up-to-date status, can therefore become a cumbersome and resource-intensive task, further complicated by the ever changing world of text processing (software and hardware). Furthermore, efficient use of the information contained in these different documents may often be achieved by experienced and specialized personnel only.

The method and tool described here allow one to reconstitute the safety-related features of a plant in a systematic and logical manner, using Reverse Failure Mode and Effect Analysis (RFMEA). These features are documented in a time-resistant electronic format inside a relational database, which becomes the centralized source of all relevant plant information and thus allowed the generation of the plant safety documents (safety analysis report, technical specifications, system descriptions, procedures, etc.).

The RFMEA approach facilitates the evaluation of changes in plant design or operation, while the centralized relational database ensures that such changes be efficiently reflected throughout all impacted documents. Furthermore, the database may be accessible through the utility's computer network, via a user-friendly browser software, so it may be used as a training tool or as a day-to-day consultation tool by utility personnel.

The RFMEA approach has been developed to be applied to the Ringhals 2 PWR in Sweden, under the name "Design Analysis Ringhals Two" (DART), in the framework of a comprehensive safety re-assessment program undertaken by Vattenfall.

The methodology, and the associated tool, turned out to be applicable to the other Ringhals units, and in fact to any nuclear power plant or risk-concerned industrial facility. Therefore, the DART acronym has been redefined to mean “Design Analysis Re-engineering Tool [*] ”.

[*] Patent application filed

BARRIERS AND RELEASE PATHS

As a basic criterion for a nuclear power plant, DART must prevent radioactive release to the environment.

As a first step, the methodology must identify all activity sources, as well as the successive barriers supposed to prevent such activity from reaching the public. In a nuclear power plant, the main activity source (the core) is shielded by four successive barriers (Figure 1): the fuel matrix, the fuel cladding, the Reactor Coolant System (RCS) boundary and the containment building boundary. This multiplication of the barriers represents the well known defense-in-depth approach.

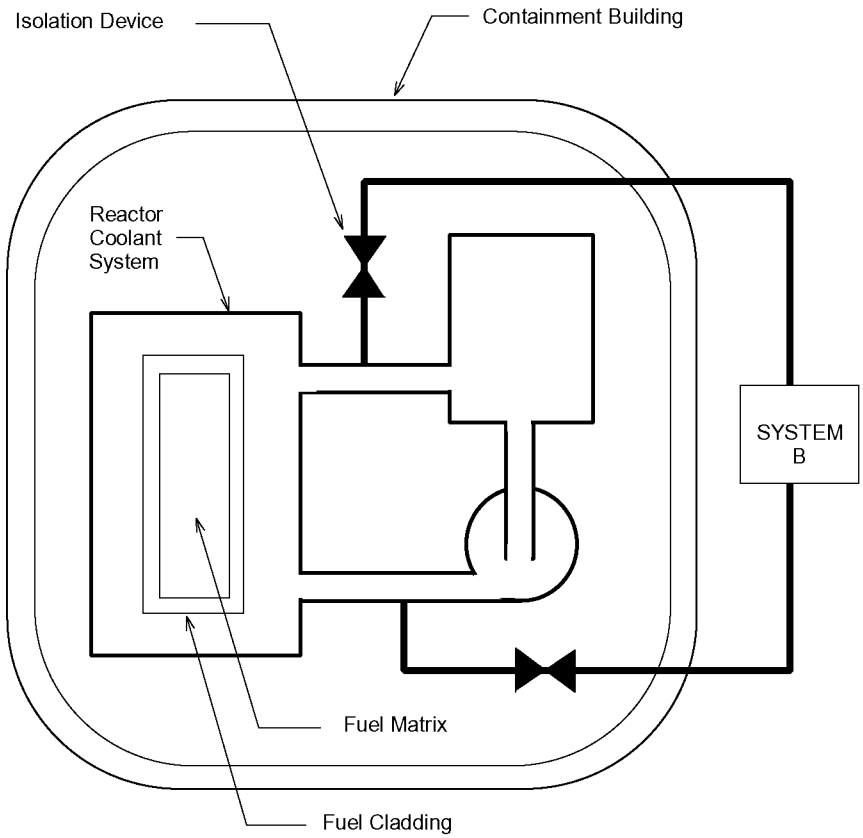

Figure 1: As a first step, the methodology must identify all activity sources, as well as the successive barriers supposed to prevent such activity from reaching the public. In a nuclear power plant, the main activity source (the core) is shielded by four successive barriers.

Other systems than the defined barriers can become radioactive and thus become additional activity sources, due to their connections with a adjacent barrier (Figure 1). System B may become radioactive if its isolation devices are opened when the upstream barrier (the RCS in this case) contains radioactivity. The boundary of System B then becomes the barrier for that activity source and constitutes a second possible release path to the environment.

REVERSE FAULT TREE APPLICATION

To retrieve and document the plant safety features that serve ultimately to limit the release of radioactivity, one uses the Reverse Failure Mode and Effect Analysis (RFMEA). Starting from the undesired consequences, one constructs logical fault trees that systematically identify successive causes of failure until the root causes have been identified. This approach originates from one used for the functional design of safeguard systems for protection against external events, for certain European NPPs [1]. Because it avoids the treatment of trivial effects, it offers the advantage of being more efficient than its counterpart, cause and effect analysis.

One identify the causes of the ultimate undesired consequence (radioactive release to environment) by logically connecting the different barriers according to each release path. For each barrier, fault trees are constructed by progressive identification of failure modes (called *gates*) which could cause the given barrier to fail. The logic is developed down to a very detailed level, with each gate in turn becoming the undesired consequence of its own causes, until the root causes have been identified.

As an example, consider a simplified fault tree identifying two possible paths leading to radioactive release to the environment (Figure 2), via the barriers shown in Figure 1. The release may come from the containment or from System B. Release from the containment requires both the presence of radioactivity in the containment and a containment failure. Radioactivity in the containment is due to releases from systems inside it or connected to it: the Reactor Coolant System (RCS) or System B. In turn, the reasons for the presence of radioactivity inside these systems is broken down by a systematic series of causes and subcauses that introduce links to other barriers and systems.

As shown in Figure 2, trees can be linked via the top gate or via intermediate level gates. A high-level operational issue can be split into very detailed causes, at the component level (for example, isolation device failure). By reading the tree from causes towards consequences, one can also determine all the significant effects of a component malfunction in the plant.

The fault trees associated with the barriers and with the systems supporting these barriers are stored in a relational database that maintains the logical links between gates and trees. One addresses each failure cause, identified by a unique gate, by defining functional requirements for measures that will prevent the failure from occurring, mitigate its effects or reduce the likelihood of failure probability. The requirements are introduced in the database in the form of structured text fragments, as described in the following section.

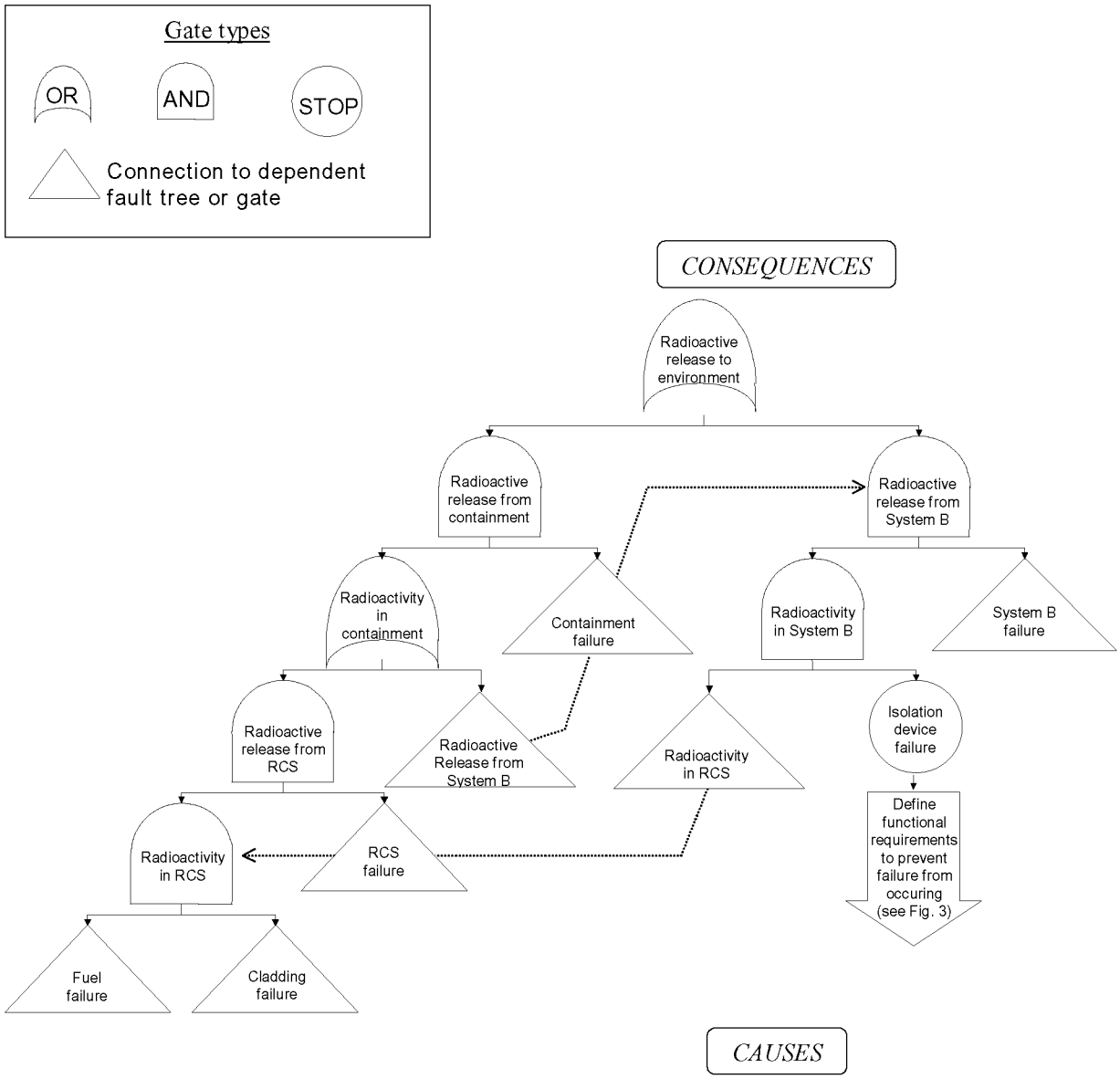

Figure 2: The fault trees start from the undesired consequence (radioactive release) and identify successive causes of failure.

STRUCTURED FUNCTIONAL REQUIREMENTS

A structured file (or *text fragment*) is associated to each gate within the database. The text describes the nature of the failure concern and the measures required (*functional requirements*) to prevent its occurrence or mitigate its effects. Such requirements, which may be as numerous as deemed necessary, are introduced as subfragments within the gate. The specific manner in which the plant complies with each functional requirement is documented within the gate file in the form of a compliance statement.

The measures required to show that the given failure has been accounted for in the plant design may be generic or specific, such as requirements for systems, analyses, procedures or technical specifications. They may also be regulatory requirements.

To distinguish clearly between the various sorts of requirement, and to allow retrieval and grouping of like information from different gates, each high-level functional requirement fragment may be divided into other types of subfragment. The rules for structuring the fragments within a gate and those for entering information within each fragment are defined by a template, which may be tailored to any project.

As an example, consider a simplified template that corresponds to the gate structure described above (Figure 3). The information in each gate may consist of entities created in the tool (such as text, equations, lists or tables) or of links to database files containing graphics or standard definitions. Links may also be made to other trees and gates in order to identify logical relationships that are not necessarily evident at the tree level.

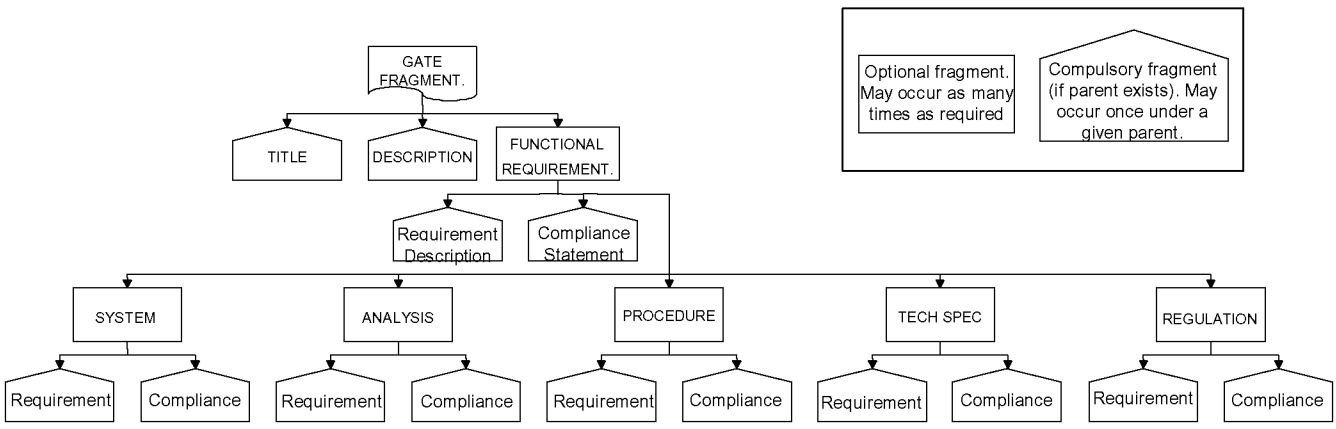

Figure 3: Dividing the functional requirements into subfragments makes it easier to distinguish between various sorts of requirements.

In practice, the specific requirement fields (system, analysis, procedure, technical specifications, regulation) may usefully be substructured, thus establishing a hierarchy between these requirements. For example, an *analysis* fragment may be added under *system* to specify the need for a system sizing analysis, or a *regulation* fragment may be added under *analysis* when rules exist for performing such analyses.

DATA FORMAT AND MANAGEMENT

A relational database management system, used as the central depository for the data, contains and controls the access to each piece of information produced during the project, including the tree structure, the gate information and all the text fragments.

The long-term accessibility and portability of the electronically stored information requires a neutral format to make the data independent from a particular software or hardware platform. The most suitable and universal format that meets these needs is the Standard Generalized Markup Language (SGML), defined in an ISO standard [2].

In an SGML file, the text is encoded as ASCII and tagged (or marked up) by SGML codes to indicate text style and formatting.

SGML allows links to other files (SGML or other, such as graphics) through "Hypermedia/Time-based Structuring Language" (HyTime) [3], another ISO standard that allows the end-user to jump (or browse) between connected pieces of information stored at different locations. An SGML file may also be reused from within another SGML file, such that the same information may appear at different locations, while being entered only once. When the reused SGML file is updated, the modifications are automatically implemented wherever this file has been reused, with no risk of inconsistencies.

Rules are necessary for structuring the information that is entered into each gate. In SGML, the rules for entering information are contained in a separate file, known as a Document Type Declaration (DTD). The DTD for a gate defines the relationship between the different text fragments (description, functional requirements, etc.), according to the structure shown in Figure 3. The DTD contains sub-structures that may need to be repeated. These substructures are defined as SGML elements, which in turn contain the rules for entering information at the next level down.

For example, the gate structure shown in Figure 3 requires five levels of SGML element definition of the following type ("*" designates optional elements – used zero, one or several times – while "|" represents an exclusive *or*):

1. gate (title , description, functional requirement*);
2. functional requirement (requirement description, compliance statement, system*, analysis*, procedure*, technical specifications*, regulation*);
3. system (requirement description, compliance statement);
4. requirement description (content*);

5. content (text | list | table | figure).

DTDs may be made as flexible or as rigid as required by the application. In the example shown on Figure 3, a gate must contain a title and a description field, whereas functional requirements are optional. However, once a functional requirement (general or specific) has been opened, it requires both description and compliance fragments. This approach ensures consistency between authors and reduces the risk of incomplete information.

The information contained within the database may be accessed and viewed with an SGML browser, such as the graphical interface used for creating the fault trees. The data can also be converted to HTML format so that it may be viewed on an intranet using a common Internet browser.

One can also create reports (as RTF files) of selected pieces, or combinations, of information extracted from the database. The report format is defined in a "style-sheet" encoded according to "Document Style Semantics and Specification Language" (DSSSL), yet another ISO standard [4]. A document may thus be regenerated at any time. Its contents then reflects the state of the database at that instant.

The DART system (Figure 4) is by nature open and flexible in the sense that it focuses on defining the content of the information. Thus, it can connect to intelligent plant information systems, like Documentum®.

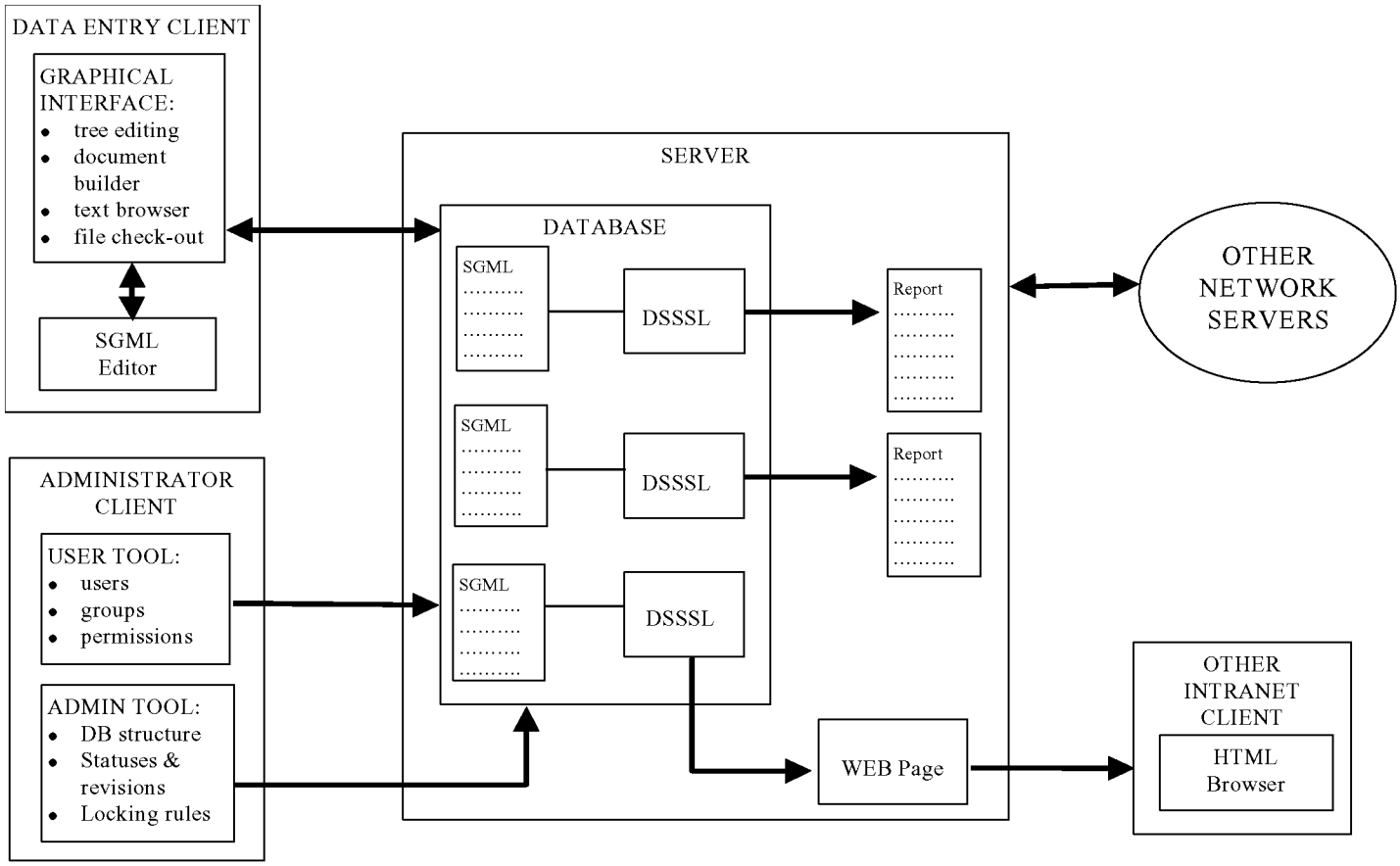

Figure 4: The DART data management system is open and flexible. It allows the user to browse connected pieces of information stored at different locations.

PLANT SAFETY RELATED DOCUMENT INTEGRATION

Once the trees and gates have been completed, the plant safety-related documents (safety analysis report, systems descriptions, technical specifications and regulatory conformance program) are regenerated from the DART database. Like for the fault trees, SGML provides a perfect means to share information between the different documents, without duplication. It ensures that all documents are updated as soon as the information itself is modified in the central database, which eliminates any risk of using obsolete information.

The flow of information between the fault tree gates and the plant safety documents avoids the duplication of information and allows the user to browse the different documents (Figure 5). The structure of every document is built-up according to its own DTD. One may insert SGML

fragments that have been created within gates into the document structure by reusing them, at as many locations as required.

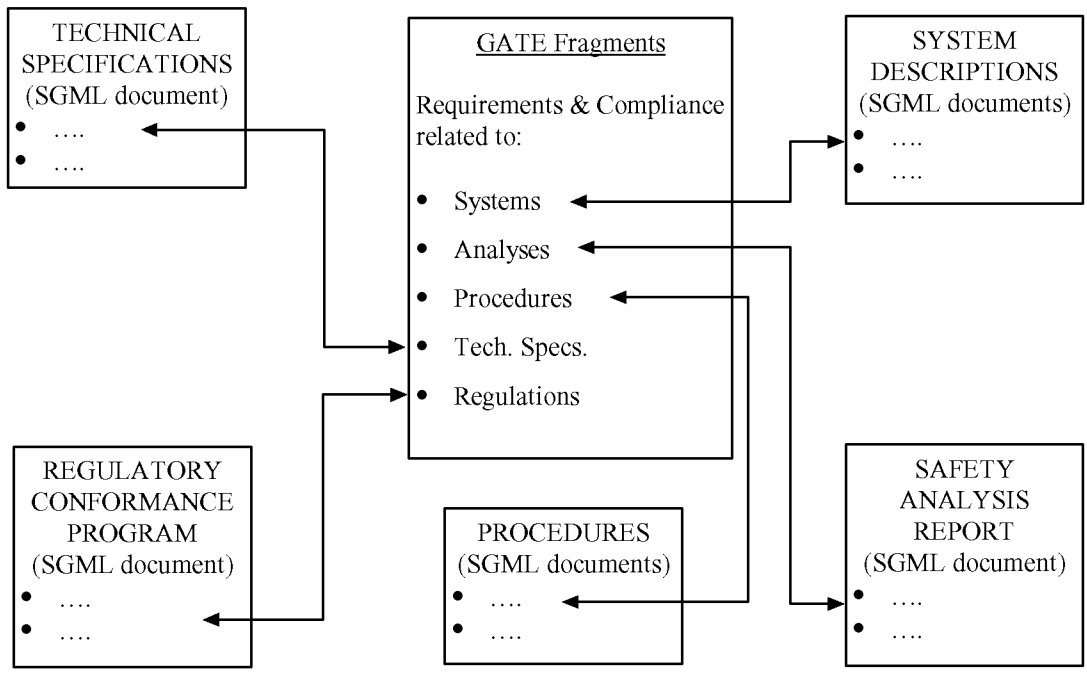

Figure 5: The flow of information between the fault tree gates and the plant safety documents avoids the duplication of information and allows the user to browse the different documents

The documents are constructed in SGML, in parallel with the creation of the functional requirements within the gates. When they are separated, specialists in each area (regulations, technical specifications, safety analysis reports, etc.) can work in parallel with the gate text authors, to create the framework and context that is specific to each document. As fragments are created on either side, they are marked by flags which define keywords that relate document fragments to gate fragments. The keywords might define the name of the barrier, system or component to which the fragment applies, or they might specify a barrier protective safety function (such as heat removal or overpressure protection).

Database searches on files with specific combinations of keywords allow one to locate fragments where information is to be shared. The necessary links may then be easily established to create an integrated and coherent set of electronic documents that form intersecting sets with the information contained in the fault trees.

Connecting all safety-related documents allows one to determine the safety impact of a given plant change by examination of the relevant fault trees and gates. Consequently, the

impacted text may be modified in a comprehensive way throughout those documents, without any duplication of information.

CONCLUSION

DART provides a systematic and methodical way of reviewing and documenting the design bases of a nuclear power plant in a format that ensures the long-term manageability and coherence of the plant safety-related documentation. It organizes the information into in a system that is readily accessible to utility personnel for day-to-day consultation or training purposes. As such, it is a powerful tool for maintaining the plant safety documentation up-to-date in a cost-efficient way.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the significant contribution to this work of the DART project team members, and especially of D. Ballant and J.P. Chaboteaux at Westinghouse and of B. Elam, J. Gällsjö and O. Johansson at Vattenfall.

REFERENCES

1. Cantineau, B. and Cecchi,T. , "Systematic Assessment of Necessary Functions for Protection Against External Accidents" Transactions of the American Nuclear Society ENC'79 Conference, May 1979, Volume 31, pp.351-353.
   (ENC = European Nuclear Conference)

2. ISO 8879:1986 “Information processing -- Text and office systems -- Standard Generalized Markup Language (SGML)” International Organization for Standardization, Geneva.
   (ISO = International Standards Organization)

3. ISO/IEC 10744:1997 “Information technology -- Hypermedia/Time-based Structuring Language (HyTime)” International Organization for Standardization, Geneva.
   (IEC = International Electrotechnical Commission)

4. ISO/IEC 10179:1996 “Information technology -- Processing languages -- Document Style Semantics and Specification Language (DSSSL)” International Organization for Standardization, Geneva.