Conversion From Tree to Graph Representation of Requirements

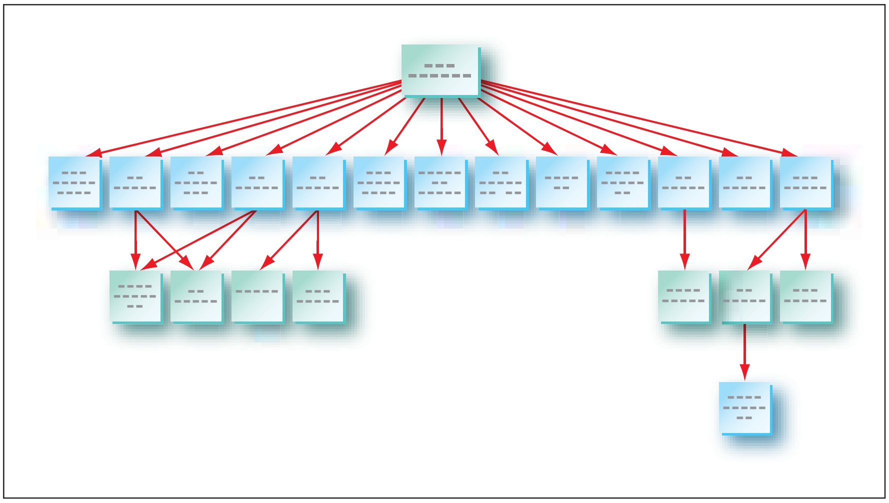
A complete representation without duplicate nodes is generated automatically.

Goddard Space Flight Center, Greenbelt, Maryland

A procedure and software to implement the procedure have been devised to enable conversion from a tree representation to a graph representation of the requirements governing the development and design of an engineering system. The need for this procedure and software and for other requirements-management tools arises as follows: In systems-engineering circles, it is well known that requirements-management capability improves the likelihood of success in the team-based development of complex systems involving multiple technological disciplines. It is especially desirable to be able to visualize (in order to identify and manage) requirements early in the system-design process, when errors can be corrected most easily and inexpensively.

Heretofore, largely for computational convenience, hierarchies of requirements have been visually represented as trees. However, the tree representation is inadequate because the underlying data structures of sets of requirements are graphs. A complete tree representation of a set of requirements usually involves repetition of some requirements nodes. In a case in which there are only a few requirements, the duplication of requirements nodes is not a significant barrier to complete understanding of relationships in the requirements document. In a typical real-world case involving a large number of requirements, the presence of many duplicate nodes makes it difficult or impossible for designers to be certain that all relationships among requirements have been identified. This leads to decision-making based on an incomplete picture of the role of a given requirement in the design of the overall system. In contrast, the graph representation provides a complete picture of relationships among requirements.

The procedure is embodied in algorithms that, in turn, are implemented in a Java-language computer program called "PaladinRM." The input to the algorithms is an Extensible Markup Language (XML) representation of requirements, generated by a previously developed computer program called "SLATE," that is equivalent to a tree representation. Exploiting portions of a body of work known among software experts as the Resource Description Framework, some of the algorithms first process the input into an equivalent graph data structure that does not contain duplicate nodes. Then a modified version of a previously proposed hierarchical-graph-layout algorithm is used to construct a block diagram equivalent to the graph data structure. The resulting display on the computer screen is a layered, directed graph (see figure) that assists in visualization of the hierarchy of requirements applicable to the system at various levels of abstraction (e.g., system, subsystems, and components).



This **Requirement Graph** is a simplified example of the display generated by the procedure summarized in the text.

The display, the algorithms, and the underlying procedure enable enhancement of the management of requirements with the following features:

- Visual representation of a subsection of the requirements document;
- Differential update of the database of requirements, based on the changes made during visual display of the requirements; and
- Annotation of requirement nodes with attributes of interest.

This work was done by David Frank Everett of Goddard Space Flight Center and Vimal Mayank, Natalya Shmunis, and Mark Austin of the University of Maryland.

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

University of Maryland
6200 Baltimore Ave., Suite 300
College Park, MD 20742

Refer to GSC-14951-1, volume and number of this NASA Tech Briefs *issue, and the page number.*