CEN 5055 Software Project Management

STATEMENT OF NEED

Project Data Dictionary System (PDDS)

GOAL:
The procurer needs a project data dictionary system having the following characteristics:
- network accessible to authorized network users
- write access by authorized network users
- versioning and rollback to previous versions
- journaling of all transactions
- syntax checking of entries according to Demarco data structure notation
- completeness and consistency checking of the system model
- report generation (at least as much as DDU version 1.0)

BACKGROUND:
Data analysis identifies the data within a system, and culminates with a data model consisting of entities, attributes and relationships. Entities are described by a collection of attributes, i.e., an entity can be described as a simple record data structure, where the record fields are the entity’s attributes. Other data identified in the analysis may also be described as data structures; e.g., a NAME may consist of a FIRSTNAME, LASTNAME and INITIAL. Each attribute has specific data properties such as data type, display format, domain (i.e., range/set of valid values). At minimum, a data dictionary captures a specification of all the elements of data model.

A data dictionary can be extended to include a complete logical system model, including system externals (screens, reports, menus), system internals such as files/tables, and system operations. Certain relationships between system elements can be specified algebraically using the Demarco data structure notation, and stored in the data dictionary. Thus, a DD can be used to document the evolution of a system through the analysis and design phases. More importantly, the DD becomes a single repository in which a system is modeled.

The types of system element types may vary from project to project, depending upon the analysis/design technique used. For example, with a structured analysis approach, system element types include records, files, fields, functions, reports, menus and screens; with an object-oriented approach, the system elements will be entities, relationships, attributes, use-cases, reports, menus and screens. A general purpose data dictionary will permit the DD administrator to configure the DD element types to conform with the analysis/design technique being used.

The value of a data dictionary, besides storing the system's logical model, is the automation of the tedious and difficult process of checking the DD contents for correctness, completeness, and consistency. An example of completeness is the requirement that a DD entry exist for each attribute of an entity, or that certain properties of a DD entry must be specified (e.g., the description fields). An example of correctness is the use of the well-formed DeMarco expressions for data structures. These analysis functions help the analysts construct a solid system model.

A final benefit of a DD is computation of project metrics. Simple things such as the number of entities, attributes, relationships, reports, etc., go a long way to describe a system. Additional measures of potential system size include function points, related to the volume of data flowing across the system boundary and retained within the system. The DDU application computes such a measure, called the "data volume count". Other automated measures are possible.

An existing legacy tool, the Data Dictionary Utility (DDU), runs on a PC in DOS and performs many of the features expected in the PDDS.