Figure 6 - Dynamic Relationship Between Different Levels of a DSS

Figure 7 illustrates the use of the architecture for DSS in the domain of capital budgeting. The specific DSS shown in figures 2 to 4 are used to illustrate the striking characteristics of the assemblage process.

Figure 7(a) shows the frameworks of different levels selected by the DSS developer. Any GUI Framework could be used for the dialog part (e.g. Borland Object Windows, Microsoft Foundation Classes). The general-purpose frameworks for the presentation part chosen are Table and Tree (Figure 2 and Figure 3), the latter modeled as a specialization of Graph framework. For the decision part, the Capital Budgeting framework was created representing the concepts and
relationships in this domain theory. Finally, the DAM framework, representing the resolution technique of spreadsheets, was chosen to solve capital budgeting problems in this DSS. At this point, every framework was designed independently one of another. We assume they already exist and are available in a framework repository.

Figure 7.(b) shows how the architecture guides the connection between decision situation and resolution frameworks. Most classes of framework Capital Budgeting were specialized, represented generically in the picture as Capital Budgeting with DAM. The new subclasses know how to create a correspondent model using the DAM modeling paradigm, and how to solve models in terms of DAM resolution technique.

Figure 7.(c) shows how the presentation part is connected to the semantics part, using Tree framework example. The procedure is similar for Table. The adaptations in Tree must result in a specialized type of Tree (Item Tree) capable of dealing with the special graphical appearance of a budgetary item tree (Figure 3), as well as with the responsibilities of forwarding to Capital Budgeting with DAM objects the modeling and resolution operations to be performed by a user. Figure 7.(d) illustrates objects created during DSS execution.

4. Using Design Patterns to Identify and Solve the Problems Found in the Architecture

Two case studies were developed in [BEC93] to validate the architecture for DSS framework-based development. The approach was validated as feasible, but the process of adaptation and connection was not as flexible as expected. Indeed, the process forces the developer to create too many subclasses to customize and combine the frameworks of distinct levels, and to understand frameworks internal implementation in detail.

In [GER97], we used design patterns to evaluate the problems and to redesign the frameworks for the Capital Budgeting case study. This analysis enabled us not only to identify the main problems with these specific frameworks, but more importantly, to generalize these problems to the whole architecture. Basically, the main problems found were:

- Frameworks represented analysis (real world) concepts, and not a design targeted at reuse;
- Emphasis on white-box frameworks;
- Lack of standard interfaces to connect frameworks of different parts;
- Difficulty for understanding the architecture and frameworks developed due to the lack of adequate documentation.

In the remaining of this section, we discuss how the design patterns described in [GAM94] have helped in the identification of these problems (Section 4.1), and illustrate through a very simple example how they are being used for framework and architecture redesign (Section 4.2).