• **guidelines**: decision makers interact with presentation structures and domain concepts guided by a dialogue component. It helps the user on selecting concepts for model formulation, instantiating and combining instances, and also guides the execution of models. The Figure 4 shows the dialog component that guides a user in the definition of the investment project lifetime, by instantiating a Time object. Notice that this concept is created by manipulating the visual representation given to it, in this case, a column of the capital budgeting table.

To organize the frameworks that are used to build specific applications, the architecture divide them into parts or levels, according to their commonalities. The two main parts are **semantics** and **interface**, as shown in Figure 5. The separation between semantics and interface is nowadays a common practice in interactive software applications, allowing them to evolve independently. In our DSS architecture, the **semantics part** is developed using frameworks related to the **specification** and **resolution** of decision problems. The **interface part** is built with frameworks that provide visual representations for the semantics part objects, through which the user interacts with the semantics part objects. The interface part also includes guidelines (dialog components) that help the user to interact with the DSS using the visual presentation objects.
The Semantics part is itself divided in two other parts, Decision Situation and Resolution. Decision Situation part organizes the frameworks responsible for the model formulation using concepts derived from the cognitive world of decision makers. Thus, frameworks in the decision situation part concern just specification of problems, using an existing theory for a decision problems class. This theory is called Domain Theory [BEC93], of which the concepts and relationships can guide the specification and structuring of the decision problem. The architecture assumes that the classes of problems addressed must have a domain theory.

Considering the Capital Budgeting domain, the frameworks in the decision situation part would represent concepts like incomes, expenses, investments, profitability criteria, cost of capital, etc., and the relationships and collaborations between them.

The methods for solving problems are organized in the resolution part, which organizes the frameworks representing different method or techniques for solving problems. This separation between specification and resolution allow independent evolution of the two parts, and enable the use of different resolution methods to solve the same problem. This issue is known as model independence in the DSS field [DOL84][BLA87]. Some frameworks for the resolution part would represent mathematical solving methods like those in spreadsheets (DAM - Descriptive Algebraic Model), dynamic programming, linear programming, simulation, etc.

The architecture guides developers in the selection and assemblage of frameworks of different types for building a specific DSS [BEC96]. The assemblage involves adapting and combining the original components in order to establish the cooperation protocol defined in Figure 6. Basically the user interacts with the DSS through dialog objects, and manipulate exclusively presentation objects. These must be adapted in order to be capable of forwarding to the next level (decision situation objects) the user modeling or resolution operations. Decision objects must be in turn related to a specific resolution technique. Their adaptation also includes the responsibility of forwarding the modeling and resolution operations to last level (resolution objects).

Further details about the Architecture and the assemblage process can be obtained in [BEC93][BEC96].