
Abstract
ET: Enhanced tool (solution strategy)
The effectiveness of specification-gathering in supporting the design of software has been demonstrated. An enhanced tool / method is described for the design of software based on the authors’ technique of using specification templates to make partial specifications more commensurable. Examples are provided confirming the effectiveness of its support for software development in design.

Why ET?
The authors essentially have proposed a way of making various partial specifications commensurable. Based on the “Related Work” section, there clearly had already been some work along this line, so this is clearly not a “radical” solution. Although the authors demonstrate their technique using a “Library World” example, they intend for it to be widely applicable to a variety of specification problems, so this is not a single “enhanced solution.” Finally, although they have implemented a tool to facilitate the use of their technique, the tool itself is not the main contribution but rather an embodiment of the main contribution, which is a solution strategy.

Question - [Method/means of development]
<model-type / solution strategy>
What is the most tractable way of combining partial specifications?

<artifact-type>
The artifact being produced is a software implementation. The question is the best way to ensure that the implementation results from a consistent and complete software specification.

Results - [Procedure / technique]
<model-type / solution strategy>
The authors propose a “ViewPoints” structure for combining specifications gathered through disparate processes. The main steps in this technique seem to be the following:
• Define a ViewPoints template for each specification style x process.
• Repeat until the specification is complete
  o Identify a bit of the specification to cover next
  o Select the template with the style and process that suits this bit best
  o Fill out the template
  o Follow any steps mandated by the process tied to this template
    ▪ This may involve forking new templates covering sub-bits
    ▪ Or it may involve redacting existing templates to make them more consistent with the current bit’s template, or other templates
• There may be a final redacting step, though that is not clear from the article.

While the authors attempt to formalize some of the mathematical structure behind the ViewPoints approach, they don’t carry it too far. Moreover, they only offer a hint as to how consistency relations would be propagated through the structure of the specification hierarchy; that is, they mention automation as one approach, and they mention a tool prototype that may automate some hand-checking of consistency relations. They also offer reference [16], which may provide more detail in this regard.

Validation – [Example]
<model-type / solution strategy>
The authors outline a toy example (a library) demonstrating their approach. The tool mentioned at the end seems to serve as some additional proof-of-concept, although the authors consign the details of validation using this tool to another paper (reference [34]).