

Software Creation:  
Various knowledge during detailed design

H. アボルハサニ      陳慧      高野英樹      B.H. ファー      河野善彌  
 Hassan Abolhassani   Hui Chen   Hideki Takano   Behrouz Homayoun Far   Zenya Koono  
 埼玉大学 工学部情報システム工学科

Department of Information and Computer Sciences, Saitama University

## Abstract:

This paper reports the result of a study to formulate two software components named "Design Direction Finder" and "Design Rule Inventor" in the context of software creation project. Design rule inventor is responsible to detail current design pattern when there is not any past experience for that pattern. Design direction finder, on the other hand, must decide about the correct way that detailing process should forward. The results shows that by studying micro steps of well human made designs it is possible to recognize rules that formulates both of these components.

## 1 Introduction

This paper reports on the next phase study after Intelligent CASE Tool [CHEN98]. For this study we assumed:

1. A software development organization with high maturity as the expert model, where their design process is solid and designer's actions are correct.
2. Their hierarchical work process intersected by hierarchical documents is as the knowledge model.

In Intelligent CASE tool, only automatic hierarchical detailing reusing the past design has been made, thus following two kinds of human design works have been left unautomated:

1. To create a new hierarchical detailing,
2. To direct design toward the direction that it should goes.

The former is named 'Design Rule Inventor' and the later is named 'Design Direction Finder', and they are the object of this study.

## 2 The Method of Analysis

In order to analyze human mental operations during designing a software using hierarchical detailing, following method has been used.

The top of Figure 1 shows a hierarchical design process, and successive documents in the center are hierarchical detailing by a designer. Based on this hierarchical detailing, one can analyze the hierarchical

expansion in three steps of *Cognition*, *Decision* and *Transformation* by an I.P.O. chart as shown at the bottom. Since the logical distance of two consecutive documents during a hierarchical detailing ('design rule') is small, the analyst's estimation of lower level mental operations of each design rule conforms generally well to show what happened during actual design.

After showing each design rule as an I.P.O., the next step is to estimate the mechanism of 'Transforming' a design input to the hierarchical output(s). Detailing of 'Cognition' specifies the preconditions for starting of the transformation. Thus a design rule may be detailed to a 'if-then' rule.

Figure 2. is a design trace for a 'clock' program. At level 1 of the figure, the first data flow is detailed to the second one (shown at level 2), and both forms a design rule. As this shows the input side data flow is detailed to three links of the lower level data flows, it can be regarded as an application of Myer's S-T-S<sup>1</sup> [MYE78] division, and the mechanism estimated is to divide the flow by finding two most abstract data

<sup>1</sup>S-T-S division is a method for structured design proposed by Myer in the context of Data Flow Design to decompose a function. The major steps of this decomposition method is: 1) Identifying major data flow, 2) Finding two points in data flow where input and output are existed in most abstract forms, 3) Inserting three functions between: a) Input and it's Most abstract form, b) Most abstract input and Most abstract output and c) Most abstract output and Output.

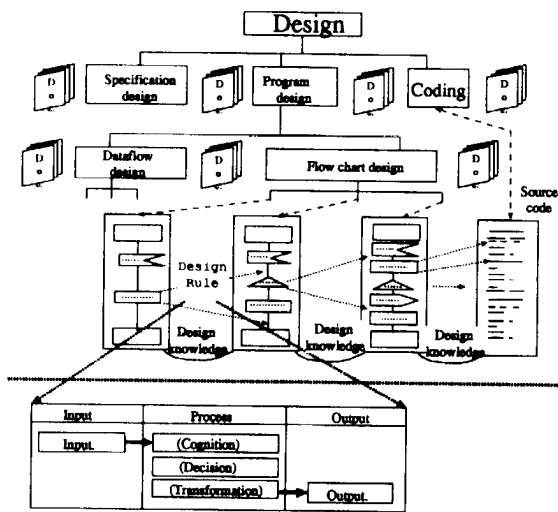


Figure 1. Assumed work process

points and then insert a boxes to each corresponding link of data-flow. The estimate for cognition can be *the existence of such abstract forms.*

Detailing 'Transform to hands' function at level 2 is another example. Here input data and output data are hierarchical and also a function is placed between each detailed input and the corresponding detailed output. This case can be related to an application of J.S.P.<sup>2</sup> [JAC75] division, the mechanism estimated for the transformation is first to find correspondence between input and output hierarchical data structures, and base on it then insert boxes showing the sub functions in a hierarchical detailing. Cognition can be estimated as *existence of correspondence between input and output structures.*

Because the design steps selected for analysis is small enough to prevent large variations for interpreting the mechanisms, continuing this way repeatedly, it is possible to find good estimates for each step of a micro design process. Gathering these estimates of transformations for a number of samples derives the rules that constitutes 'Design Rule Inventor'. Similarly estimates for the cognition steps of design rules formulate the 'Design Direction Finder' to select the rule that must be applied on current design pattern.

<sup>2</sup>J.S.P. is another method for structured design proposed by Jackson in the context of design by data structures. The idea is that the correspondence of input and output hierarchical structures determines the decomposition of a function to more detailed sub functions, resulting in a hierarchical decomposition of a function.

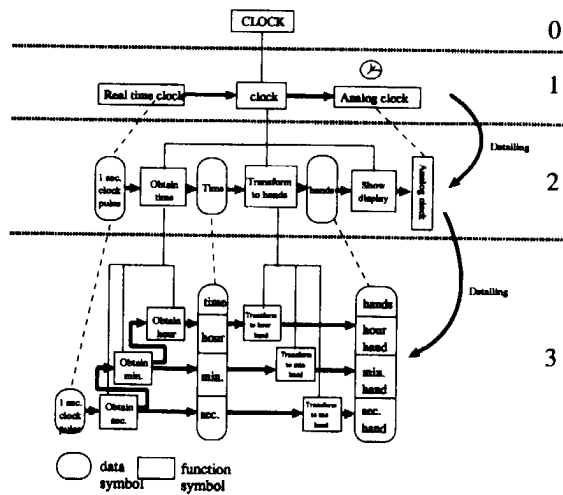


Figure 2. Design of 'a clock'

### 3 Conclusion

As the previous section described, a base for analyzing human mental operations related to design activities is established. The focus of current study is at detailed design, and the results achieved up to now show that S-T-S and J.S.P. detailing methods are the most hierarchical division methods in this level. By making this study for micro operations of a reasonable number of good human designed samples, we hope to be able to find estimations for almost all human mental operations related to design at detailed design level.

The next phase is to apply such analysis for upper levels of design process, to find estimates for whole mental operations related to design process, and then by simplifying make them appropriate for actual use.

### Acknowledgment

Authors wish to express their gratitude to students who built the basis for this study in the Laboratory.

### References

- [CHEN98] H. Chen, H. Takano, N. Tsutsumi, Z. Koono, "Software Creation: A prototype of Intelligent CASE tool for detailed design", Proc. 12th Annual Conference of Japan Society for Artificial Intelligence, 1998 (This Conference).
- [JAC75] Jackson, M., Principles of Program Design, Academic Press, 1975.
- [MYE78] Myers, G. J., Composite/Structured Design, Van Nostrand Reinhold, 1978.