Towards ontology-based support for reducing semantic errors in function decomposition

学籍番号：90152171　　溝口研究室　　Houda MESBAHY

1. Introduction

Although the importance of knowledge sharing among designers has been widely recognized, knowledge about functionality in the conceptual design phase is hard to capture and is often scattered across technical domains. As a tool for knowledge sharing among designers, our team has been developing a function decomposition editor (FDE). It provides users with means to describe and edit function decomposition trees. In order for a user to do so correctly, there are some guidelines to follow. The existing function decomposition tree editor allows semantic mistakes, and the users receive no clue about the errors they make. For reducing such semantic errors, this paper proposes a new framework of the new FDE to support editing functional models .

2. New Function Decomposition Editor

T[0]he new FDE provides the users with means to avoid semantic mistakes based on ontological definition of functional concepts. Fig.1 shows the system architecture. This study designed additional 3 modules to the current version of the FDE: An Assisting tools module, an Ontology access module and a Model access module. HOZO is an ontology editor developed in our laboratory, and works with the new FDE.

The Assisting tools module helps users to build semantically correct functional models by checking the input/output objects of each function. [0]The Ontology (Model) access module provides a means for accessing to Ontologies (Models) contained in HOZO.

Fig.2 shows the user interface of the new FDE. As shown in this figure, the user describes the functional model first. Then the Assisting tools check the input and output of the functional models constituting the function decomposition tree. In order to run this check, the Assisting tools access the model or ontology inside HOZO. The ontology includes class constraints for input/output objects and port numbers in the functional concept definition. This constraint has 2 patterns: (1) the class concurrence check and (2) the port numbers concurrence check. In the constraint check process, as shown in Fig.3, upon a query from the user, the model or ontology is shown on the API console. According to the above functions, the user can verify whether there is an error or not. The mistaken model can be corrected using the editing function.

3. Future work

Improvement of the new FDE by describing various functional models remains as a future work.

I also plan to improve the new FDE so that an automatic correction of the functional models can be made. By correcting I mean, if there is a mistaken function, providing alternative functional models to replace it.

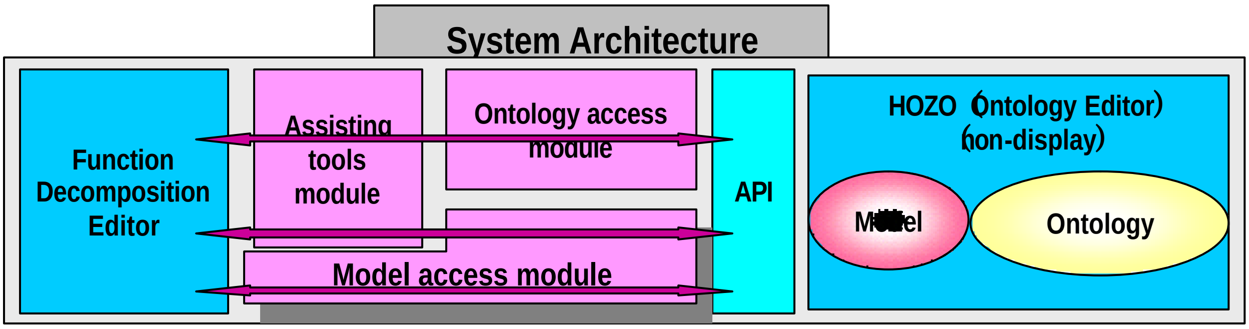


Fig.1: Architecture of the new function decomposition editor

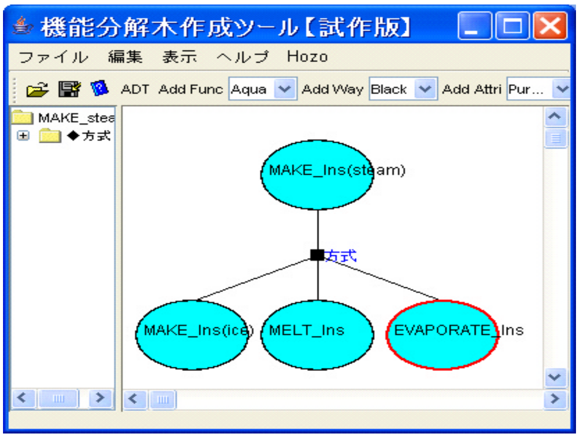
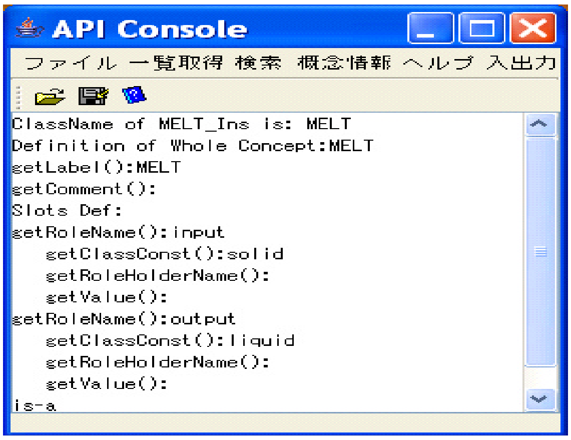

機能分解木作成ツール【試作版】
ファイル　編集　表示　ヘルプ　Hozo
ADT　Add Func　Aqua　Add Way　Black　Add Attri　Pur...
MAKE_stea
◆方式
MAKE_Ins(steam)
方式
MAKE_Ins(ice)
MELT_Ins
EVAPORATE_Ins

Fig.2: New Function Decomposition Editor



```
ClassName of MELT_Ins is: MELT
Definition of Whole Concept:MELT
getLabel():MELT
getComment():
Slots Def:
getRoleName():input
   getClassConst():solid
   getRoleHolderName():
   getValue():
getRoleName():output
   getClassConst():liquid
   getRoleHolderName():
   getValue():
is-a
```

Fig.3: Textual display of functional concept definition