Next Generation Astah: A platform for creating modeling tools equipped with tactful editors

Shin Kuboaki, Toshinori Takai, and Toshihiro Okamura

Change Vision, Inc.

Abstract. We are developing a platform for creating modeling tools equipped with tactful user-interface functions. Using this platform, we are aiming at providing a next-generation modeling tool supporting various types of models. In this session, we demonstrate that the proposed platform can provide modeling tools having practical usability capable for industrial use, assuming that some original modeling languages with original presentations are given to our platform as meta-models.

1 Requirements for modeling tools having tactful editors

In order to propose a new modeling language based on a new idea, any modeling tools for that language should also be constructed and presented. More precisely, lacking modeling tools with practical usability could prevent fair recognition for a new proposal even if it has innovative ideas. We call a modeling tool with practical-level usability by providing tactful features a *modeling tool with a tactful editor*. We consider the followings are the requirements of such tools:

- full-conformance to a given specification
- stress-free usability
- engineering features

The first means the specification of a supporting modeling language is explicitly given as a meta-model and the editor accepts models defined by the specification. Example features for stress-free usability include to manipulate properties of model elements without modal operations, to arrange model elements easily with the help of drawing guides, and to suggest possible candidates for easy entering and referring. Appropriate combinations of those features can allow users to construct models without stressful and redundant operations, e.g. superfluous direction changes of eyes. Engineering features imply to enable constructed models to be applied for code generation or model-translation for the post-processes or to be referred from other projects as reusable assets.

We have accumulated the knowledge about user-interfaces on both design and implementation aspects prevent from any obstruction for thinking during modeling processes. These experiences brought us to come up with the idea to provide a platform for creating modeling tools equipped with tactful editors, in which our knowledge on user-interface design and implementation can effectively be utilized. We call the platform *Next Generation Astah*.

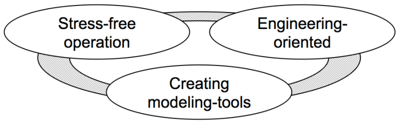


Fig. 1. A platform for creating modeling tools with tactful editors

Some tools have already been proposed aiming at the similar purpose of our platform. Enterprise Architect supports Model Driven Generation technologies [1], which enable users to obtain extended UML editors. MetaEdit+[4] provides an integrated platform with a modeling environment and a process for generating editors to support flexible construction of modeling tools. Our approach is nearer to the later one but focus on the stress-free usability of the obtained editors.

2 Features for constructing model editors

A definition of a modeling language consists of a meta-model of abstract syntax, definitions of symbols and/or diagrams, and a set of constraints on them. But those definitions do not induce suitable specifications of drawing functions for diagrams and manipulating functions for models. In order to give an effective way to identify and determine the specification of user-interfaces for generated editors, we consider a mechanism to reuse functions for drawing and manipulating diagrams. We explain the mechanism by the following use-cases: (1). to construct model repository from a given meta-model, (2). to construct operation functions to manipulate the model repository, (3). to construct presentation models from information about notations, (4). to support constructing editing functions of graphical elements for presentation models, and (5). to update model instances in a model repository through manipulating presentation models. The relation among those use-cases is shown in Fig. 2.

We have implemented a prototype of Next Generation Astah and using the prototype we construct model editors for some standardized modeling languages in UML, SysML and ER diagrams.

3 Stress-free usability

Modeling processes in modeling tools on the present day are not just to write out the final version. Instead, for example, a user first gives some possibilities of model elements and then organizes them. Such process is repeated and thus modeling tools are desired to have usability to prevent intercepting user thinking, or working rhythms. That is, modeling tools are required to have stress-free usability. For providing stress-free usability, modeling editors generated from Next Generation Astah can be associated with the following functions:

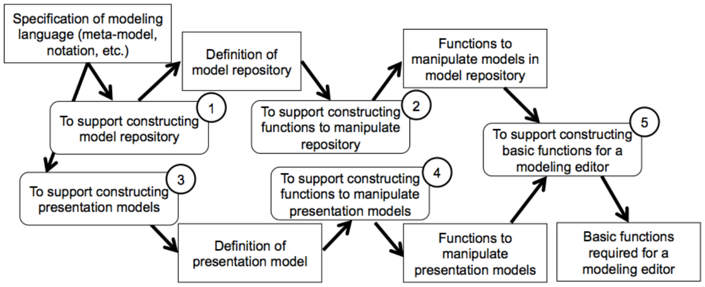
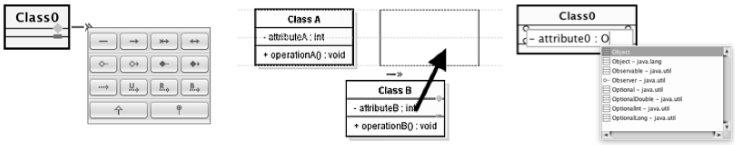

Specification of modeling language (meta-model, notation, etc.)
Definition of model repository
Functions to manipulate models in model repository
To support constructing model repository
1
To support constructing functions to manipulate repository
2
To support constructing basic functions for a modeling editor
5
To support constructing presentation models
3
To support constructing functions to manipulate presentation models
4
Definition of presentation model
Functions to manipulate presentation models
Basic functions required for a modeling editor

Fig. 2. Roles of features for creating modeling editors

- *mode-less property manipulation* feature
- *next suggest* feature (see [2])
- *alignment guide* feature (see [3])
- *content assist* feature

Manipulating properties of model elements with mode-less operations means to set and edit properties without any dialog-box appeared according to modes. Dialog-boxes can hide other model elements already exist on the drawing canvas and imply to intercept user thinking process. Content assist is a feature to display a list of possible candidates that can be added to a focused model element that can be selected by users. With the help of this feature, users do not need to remember all the names of those candidate models or to refer to other diagrams. With the help of this feature, users do not need to remember names of types or operations and also do not need to refer another diagrams. Even in compare with editing through a property pane for the mode-less property manipulation feature, distances of user's eyes direction can be minimized.



(a) next suggest for a class (b) alignment guide (c) content assist

Fig. 3. Some features for stress-free usability

4 Engineering-oriented

For users of modeling tools, developing models is not the ultimate goal; the models are just intermediate artifacts during a development process. The obtained models are expected to be the input of the subsequent processes. Next Generation Astah provides the following functions to support engineering-oriented modeling:

- exchanging the artifacts with other projects,
- supporting multi-platform
- providing APIs for manipulating models
- providing a plug-in mechanism
- providing a script-engine mechanism

A modeling editor obtained by Next Generation Astah has functions to import and reuse existing models written in the same modeling language with the help of standardized model exchange data format like XMI.

For practical use of modeling tools in industry, another important aspect other than usability is a function to allow other tools to derive model repositories and editors without manual operations by human beings. An editor constructed by Next Generation Astah provides APIs to manipulate meta-models and presentation models. Moreover, Next Generation Astah itself has components to provide a plug-in mechanism, script-engines and the constructed editors easily use those components. Applying these functions, a user of the platform can obtain a modeling tool equipped with advanced tool-chain features from very early stage of the development.

5 Future work

We will compare with existing platforms for generating modeling tools from the view of usability of the obtained editors. The plan contains comparison by features [5], the number of actions for typical use-cases by modeling editors [6], and an empirical approach [7].

References

1. http://www.sparxsystems.com/enterprise_architect_user_guide/9.3/standard_uml_models/mdgtechnologies.html
2. http://astah.net/features/suggest-feature
3. http://astah.net/features/alignment-guide
4. J.-P. Tolvanen, R. Pohjonen, S. Kelly: "Advanced tooling for domain-specific modeling: MetaEdit+," in Proc. of the 7th OOPSLA Workshop on Domain-Specific Modeling, Montreal, pp. 48-55, 2007.
5. B. Langlois, C.E. Jitia, E. Jouenne: "DSL classification", in Proc. of the OOPSLA 7th Workshop on Domain Specific Modeling, 2007.
6. M. Auer, L. Meyer, S. Biffl: "Explorative UML Modeling - Comparing the Usability of UML Tools," Proc. of the 9th International Conference on Enterprise Information Systems, pp. 466-473, June 2016.
7. N. Condori-Fernández, *et al.*: "An empirical approach for evaluating the usability of model-driven tools," Sci. Comput. Program, vol. 78, no. 11, pp. 2245-2258, 2013.