

---

# Automated user documentation generation based on the Eclipse application model

Marco Descher, Thomas Feilhauer, Lucia Amann

Fachhochschule Vorarlberg – Forschungszentrum für Prozess- und Produkt-Engineering,  
Hochschulstrasse 1, A-6850 Dornbirn, AUSTRIA

---

## ABSTRACT:

An application's user documentation, also referred to as the user manual, is one of the core elements required in application distribution. While there exist many tools to aid an application's developer in creating and maintaining documentation on and for the code itself, there are no tools that are complementing code development with user documentation for modern graphical applications. Approaches like literate programming [1] are not applicable to this scenario, as not a library, but a full, user interface capable, application is to be documented to an end user. Documentation generation on applications, up to now, was only partially feasible [2,3] due to the gap between the code and its semantics. The new generation of Eclipse rich client platform developed applications is based on an application model, closing a broad semantic gap between code and visible interface. We use this application model to provide a semantic description for the contained elements. Combined with the internal relationships of the application model, these semantic descriptions are aggregated to well-structured user documentations that comply with ISO/IEC 26514 [4].

## 1 INTRODUCTION

The project is focused around two core elements: the Eclipse application model, providing a comprehensive model regarding the applications code, and the ISO/IEC 26514 standard for user documentation generation.

### 1.1 Eclipse application model

Eclipse is one of the most popular development environments for software. Generation 4 of the Eclipse Rich Client Platform (RCP) defines an application model, as the base for the application's user interface. This application model allows to define, amongst others, visual representation and acting elements. On the actual creation of an application (the development process), each of these elements is created out of the necessity of a specific user story; by attaching the information contained in this user story to the resp. element we gain a documented application model.

### 1.2 ISO/IEC 26514

The main standard, specifying the requirements and structure of a software documentation, targeted at the end user is ISO/IEC 26514. This specification hence sets the structure of the documentation to be generated, out of the documented application model.

## 2 THE CORE ISSUES

The core research question was: *What are the potentials and limitations of user documentation generation based on the Eclipse application model?* In the course of the project this question was broken down in four separate issues to be treated and a software toolkit for documentation generation was created. The respective questions are elaborated in [5].

### 3 THE ÉCRIT TOOLKIT

The Écrit toolkit [6] consists of two separate software packages: an extension of the Eclipse application model editor to allow semantic description of model elements, and the actual project generating the documentation. The toolkit is further described in [7, sect. V].

#### 3.1 Eclipse application model extension

The Eclipse application model editor is a graphical editor to configure the application model. It supports all the application model elements, and allows altering their respective properties. Depending on the model element to be edited, different properties have to be presented in order to satisfy the documentation model requirements. Fig. 1 shows an example for the application model element *Command*. Here the developer provides the description of the command, the precondition for the commands execution and the post-condition after executing the command. The semantic description is then stored within the application model.

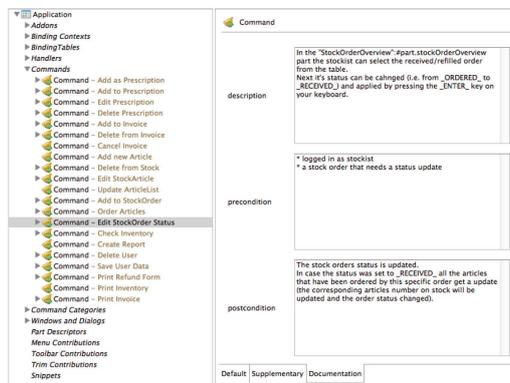


Figure 1. Editing the semantic description of an application model *Command* element.

#### 3.2 Eclipse Écrit plugin

The Écrit plugin tightly integrates into the Eclipse IDE. By executing it on an application model or product element and selecting an Outputter (currently there exist HTML and LaTeX Outputters), we receive a generated application, user documentation.

Analyzing and transforming the documented application model into a document model, which acts as the base for a text template system, realizes the required output. While the elements in the application model are contained as tree, they are “enriched” with information about their embedding in the document model, providing the required information to embed the respective element in its required documentation context. Fig. 2 exemplarily shows this for the visual representation of a perspective and its respective embedding as an image.

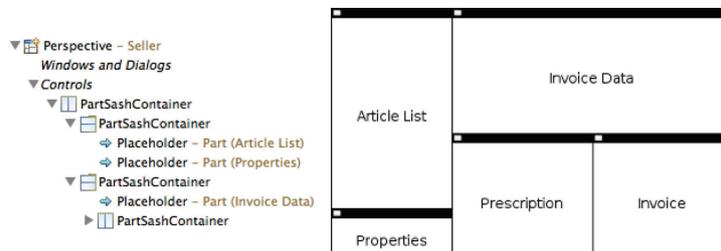


Figure 2. Sample perspective and it's representation as a depiction image

A sample for generated user documentation can be found on the projects homepage [6].

## REFERENCES

- [1] Donald E. Knuth, Literate Programming, The Computer Journal, 27 (2), 1984, 97 – 111.
- [2] W. L. Johnson, Dynamic (Re)Generation of Software Documentation, Proc. 4th Systems Reengineering Technology Workshop, J. Hopkins University, 1994.
- [3] Paris et al., Automatic Document Creation from Software Specifications, Proceedings of the 3rd Australian Document Computing Symposium (ADCS), 1998.
- [4] ISO/IEC 26514, Systems and software engineering – Requirements for designers and developers of user documentation, International Standard, First Edition, 2008, ISO/IEC 26514:2008(E)
- [5] M. Descher et al., Towards automated application software documentation generation based on the Eclipse application model, Proceedings of the IASTED International Conference Software Engineering (SE 2014) (M.H. Hamza, ed.), 39 - 45, 2014.
- [6] M. Descher, T. Feilhauer, and L. Amann, The Écrit project – Who writes the user manual? <http://www.ecrit.at>, 2014
- [7] M. Descher et al., Automated user documentation generation based on the Eclipse application model, <http://arxiv.org/abs/1411.3818>, Nov. 2014