SOFTWARE DESIGN AUTOMATION IN AN IPSE

Stephen Beer and Ray Welland,
Department of Computer Science,
University of Strathclyde,
Glasgow G1 1XH, Scotland.

Ian Sommerville,
Department of Computing,
University of Lancaster,
Lancaster LA1 4YR, England.

ABSTRACT

This paper describes an editing system which is explicitly designed to support the production of graphical representations of a software design. The novel features of the system are that it is table-driven, so that it may be tailored to support most graphical design notations, and that it explicitly includes a means of defining the rules of the design method. These rules may be enforced automatically or checked at user request by the design editing system.

Furthermore, the system is intended to operate within the context of an integrated project support environment called ECLIPSE and the designs generated by the system are stored as objects in the ECLIPSE database. These objects have a defined structure and may be manipulated by other tools (such as a code generator). The editing system is implemented in C and runs on a Sun workstation.

Keywords: Design diagram construction, Method description, Design checking, Design method support, Integrated Project Support Environment.

Automation de la Conception du Logiciel dans un Environnement Intégré

Résumé

Cette communication décrit un système d'annoter, qui est explicitement construit pour supporter la production de représentations graphiques d'une conception du logiciel. Les traits nouveaux du système sont qu'on le fait marcher avec des tables pour qu'il soit possible d'adapter pour supporter la plupart des notations graphiques de la conception et aussi qu'il embrasse explicitement les moyens de définir les règles de la méthode de la conception. Le système peut automatiquement observer ces règles ou il peut les vérifier sur la demande de l'utilisateur.

De plus, le système a l'intention de fonctionner dans le contexte d'un environnement intégré qui s'appelle ECLIPSE et les conceptions, que le système produit, sont gardées comme des objets dans la base de données d'ECLIPSE. Ces objets possèdent une structure bien déterminée et des autres outils (comme un générateur de code) peuvent les manipuler. Le système est écrit en 'C' et il marche dans un poste de travail de SUN.

Mots-clés: La construction de notations graphiques de la conception; la description des méthodes de développement du logiciel; la vérification de la conception; les méthodes de développement du logiciel; l'environnement intégré de soutien de projet.
INTRODUCTION

The work described here is taking place in the context of an integrated project support environment (IPSE) called ECLIPSE (1) where we hope to support different approaches to the software process and provide tools to assist with software design.

Although they have been criticized for lack of formality, graphical approaches to software design such as Structured Design (2) and JSD (3) are widely used and reportedly successful. Thus, it was a requirement of ECLIPSE that it should be possible to incorporate support tools for such methods and to allow designs produced with such tools to be stored and manipulated in the ECLIPSE database. ECLIPSE is an open IPSE so the methods to be supported could not be predefined - indeed as the system construction progressed we changed our mind on which methods should be supported in the initial release of ECLIPSE. A further requirement on ECLIPSE was an integrated approach to the user interface so we came to the conclusion that the most effective way to provide a design support tool for graphical design methods was to produce a generic tool which could be tailored by the system builder for whatever methods were supported in any single release of ECLIPSE.

However, we did not simply want a graphical editing system which allowed us to produce neat diagrams. We also wished to provide as much design checking as possible as the design was created - in essence we wanted to provide a syntax-driven editing system to support whatever graphical method was in use. Thus, we decided that we should define a notation for defining the method to be supported and use this to generate tables to drive the design editor. Furthermore, as we wished to support notations with arbitrary symbolism we also needed a tool with which we could define symbols and relate them to their names used in the design definition language.

The design definition language which we developed is called GDL (4) and is a notation for defining the syntax and partial semantics of software designs which are expressed as directed graphs. This means that most graphical design methods may be supported. The relationship between GDL, the symbol editing system, the design editor and the database is shown in Figure 1.

![Figure 1. The ECLIPSE design editing system](image)

The mode of use is as follows:
1. The IPSE tool builder defines the syntax and the semantics of the design method to be supported using GDL.
2. The GDL compiler generates tables for input to the design editor.