Towards a Systematic Building of Software Architecture: the TRIDENT Methodological Guide

François Bodart, Anne-Marie Hennebert, Jean-Marie Leheureux,
Isabelle Provot, Benoît Sacré, Jean Vanderdonckt

Facultés Universitaires Notre-Dame de la Paix, Institut d'Informatique
Rue Grandgagnage, 21, B-5000 Namur (Belgium)
Tel : + 32- (0)81- 72 50 06 - Fax : + 32- (0)81- 72 49 67 - Telex : 59.222 Fac. Nam.B
E-mail : fbodart@info.fundp.ac.be

Abstract. When designers are facing the question how to build an application architecture practically, they often have to consider various arguments and factors coming from different perspectives: decomposition criteria in architecture design, dialog independence in user interface (UI) design, methodology to follow in a development team. These factors are not easy to conciliate, forcing designers to make trade offs or unbalanced choices. In this paper, we discuss an architecture model, which is part of TRIDENT project, that addresses these issues. It consists of a generic architecture model for highly interactive business oriented applications. It is accompanied with a practical task-based methodology for building an architecture that automatically preserves desired criteria. Assumptions made for the architecture model, its content and the semantics of relationships are explained. The systematic approach is exemplified by a complete architecture case throughout the paper. Software Architecture Analysis Method (SAAM) is finally applied to prove the benefits of this architecture and to evaluate it with respect to relevant criteria. This paper suggest first steps towards a systematic building of a software architecture.

1 Introduction

Several architecture models which are convenient for an interactive application have already been introduced [1,6,7,14,15]. Though they can be examined according to a generic metamodel [16], it is not always clear to judge the appropriateness and the quality of these models [8,9]. Up to now, gained experience shows that a multiplicity of factors is to be considered :

- traditional architectures coming from software engineering do not support UI design easily : architectures should provide relevant abstractions needed for configurable UI components, particularly when UI is adaptable [4];
- application architectures should satisfy usual decomposition criteria : e.g., high internal cohesion, weak coupling degree, easy component reusability [1];
- application architectures should meet more advanced criteria needed by interactive application development : e.g., conceptual simplicity, reuse of code, quality of resulting UI, easy maintenance [16];

P. Palanque et al. (eds.), *Design, Specification and Verification of Interactive Systems '95*

- separating UI functionality from application semantics is one of the most important goals of architecture design [16];
- particular architectures should adhere to a generic model that supports this separation and respects mentioned criteria [6,7,14,15,16].

These opposed considerations are not effortless to manage simultaneously so that the quality of resulting architectures are not easy to judge. SAAM has been introduced for this purpose [8,9]. Whereas existing architecture models can be described, understood and evaluated through SAAM, such models do not suggest any methodology to follow for reaching an architecture

- that is intrinsically compatible with the architecture model,
- that automatically preserves expected criteria,
- that rests on task analysis rather than simply technical considerations,
- that guarantees compatibility with task model rather than application model,
- that allows systematic generation and easy change of interaction techniques.

However, several works already consider task aspects in their approach [11,12]. In this paper, we report on TRIDENT (Tools foR an Interactive Development ENvironmenT) which utilizes an architecture model and promotes a systematic methodology that addresses these issues.

2 Background on TRIDENT

The main goal of TRIDENT is to allow interface designers to generate as automatically as possible a user interface to a highly interactive business oriented application from specifications describing different models (e.g., task model, object oriented entity/relationship model) [2,3]. Multiple aspects such as selection of presentation units, choice of appropriate interaction objects, computer-aided placement of interaction objects, computer-aided window sequencing, basic conversation generation are covered [3].

This generation also encompasses a code skeleton of the application containing the required abstractions for a practical and effective architecture. This paper is dedicated to the methodological steps required for building practically such an architecture in this context. This methodology supposes that task analysis and presentation design have already been performed. We show now the contents of these pre-requisite. More details are found in [3].

2.1 Task Analysis

One of the results of task analysis is a model describing the interactive task the user has to carry out. This task behavior model can be graphically represented with an *activity chaining graph* (ACG). An ACG depicts an information flow between functions chained to define a business logic in order to reach the goal(s) assigned to the interactive task.