Abstract. The increment of the complexity of systems requires new techniques that allow manipulating it adequately. Software architecture is becoming an important part of software design, which helps developers to handle the complexity of large systems. In addition, the management of the evolution as well as the maintenance of complex systems are two of most important problems to be solved by software engineering. Several solutions have been considered, one of them being the separation of concerns. These concepts have been extended along the life cycle and thus, Aspect Oriented Software Development (AOSD) arose. In this paper the architectural design phase and Aspect Oriented concepts are considered jointly. A proposal introducing aspects modelling in the architecture design phase is presented. The research is based on the combined use of a conventional architecture description language and an exogenous co-ordination model. When new requirements are going to be included in the system, the proposal provides the required steps to allow its evolution and maintenance by specifying an Aspect Oriented Architecture, which will permit us to change the system easily.

1 Introduction

More and more the complexity of systems to be developed increases and new design techniques are necessary. In addition, the structure of the systems changes throughout their life cycle and evolves to adapt it to new situations, making it necessary to develop the systems keeping an easy maintenance.

On the one hand, new approaches propose the Software Architecture as an important part of the design phase helping to manage the systems complexity as well as to define the system structure in such a way that its maintenance and evolution are easy.

On the other hand, AOSD, which extends aspect oriented (AO) programming concepts to the early stages, gives approaches to allow for the early identification of the aspects; their extraction, representation, and composition are their main goals (early aspects [1]). It recommends considering crosscutting concerns throughout the life cycle to obtain a clear designs and untangled code. Crosscutting concerns can be

* This research work is partially supported by CICYT under the TIC2002-04309-C02-01 project.
encapsulated into so-called aspects. AOSD provides techniques for modularising and composing concerns which are difficult to untangle using the traditional ones.

Software Architecture is an appropriate moment in the life cycle to consider aspects, because it is when the structural definition of the systems is done. The suitability of the definition of an AO architecture design appears when it is observed that crosscutting concerns cross the architectural components, contaminating components and connectors, and the final design become complex.

The aspects can be considered as first class elements of architectural development. For this, having mechanisms to identify and specify them during the architectural design is needed; as well, its interaction with other architectural elements must be defined. Recently, a great number of research whose results show the benefits of an architectural approach have been presented [1], [2]. Therefore, the complex systems design is easier.

Systems, throughout their life, need to be maintained due to the changes in the world that they represent. This means that the requirements defining the changes of the system need to be considered. The design of systems can be redefined by adding, eliminating or changing elements of the current design structure.

In this paper a proposal to model aspects at architectural design phase is shown. Some methodological considerations about how to deal with the integration of the aspect separation at the architectural level are made. This allows us to manage the evolution of systems by considering the changes as aspects. To achieve it, the aspects are extracted during the architectural design stage. The model, which proposes to give a structural specification of an AO system, is based on the combined use of a conventional architecture description language (ADL) and an exogenous co-ordination model. In particular, LEDA [3] has been chosen as ADL and Coordinated Roles [4] as co-ordination model. The formal basis of the language will allow us to reason about the properties of the software architecture as well as to execute a prototype of the system from its architecture design. The generated systems have a clear design, easy evolution and maintainability. In addition, the language needs to be extended to express the AO concepts.

The proposal is a contribution to obtain an AO life cycle: a methodology is defined to evolve architectures, based on Aspect orientation. An ad-hoc architectural reconfiguration is obtained when a new requirement is added by a software architect, who can obtain a “new design” from the current system by applying the proposal. In others words, it is possible to apply aspactual concepts to a system, which need to evolve due to new requirements or unanticipated changes. It is assumed that the new requirements that will be modelled as aspects at the software architecture phase have been identified in earlier steps. So, from an initial system, a tool allows us to obtain the design of the extended one with new requirements (considering them as aspects). The definition of an AO-ADL gives the necessary support to do it.

The article is structured as follows: in section 2, the proposal is presented, considering the development process of the AO systems during the architectural stage. The proposal includes the definition of an architectural structure with two levels, the component level and the aspect level. Also, how to manage the aspect

1 New requirements considered as aspects can crosscut or not the architectural modules of current system.