Evaluating Alternative COTS Assemblies from Imperfect Component Information

Hernán Astudillo[1], Javier Pereira[2], and Claudia López[1]

[1] Universidad Técnica Federico Santa María, Departamento de Informática,
Avenida España 1680, Valparaíso, Chile
{hernan, clopez}@inf.utfsm.cl
[2] Universidad Diego Portales, Escuela de Ingeniería Informática,
Av. Ejército 441, Santiago, Chile
javier.pereira@udp.cl

Abstract. Component-based approaches to elaborate software must deal with the fact that in practical settings, components information may be incomplete, imprecise and uncertain, and requirements may be likewise. Architects wanting to evaluate candidate architectures regarding requirements satisfaction need to use whatever information be available about components, however imperfect. Imperfect information can be dealt with using specialized analytical formalisms, such as fuzzy values for imprecision and rough sets for incompleteness; but if used, evaluations need to compare and rank using non-scalar, non-symbolic values. This article presents an approach to systematically describe components' imperfect information, and to evaluate and rank whole component assemblies, by using credibility values-based "support scores" that aggregate imperfect information about requirements, mechanisms and components. The approach builds on the Azimut framework, which offers progressive refinement of architectural entities via architectural policies, architectural mechanisms, components, and component assemblies. An example of the proposed approach and "what-if" analysis are illustrated.

1 Introduction

Component-based software development proposes building systems by using pre-existing components, to reduce development time, costs and risks and to improve product quality; achieving these goals requires an adequate selection of components to reuse. Most current methods of component evaluation and selection are not geared to support human specialists in the systematic exploration of design spaces because they require rather complete and/or consistent descriptions of components behavior, connections and prerequisites. Unfortunately, in practice architects have at hand incomplete, imprecise, and uncertain information about components, and perhaps even about requirements. Even more, mathematical formalisms that describe imperfect information are not directly amenable to comparing alternatives.

This article builds on the Azimut approach [12], which proposed progressive refinement of architectural abstractions and artifacts via architectural policies,

C. Hofmeister et al. (Eds.): QoSA 2006, LNCS 4214, pp. 27–42, 2006.

mechanisms, components and assemblies. It presents an approach to evaluate and compare whole component assemblies, even if the available information is imperfect (possibly incomplete, imprecise and uncertain), by defining "support scores" that reflect the "strength" of architectural artifacts (mechanisms, components and assemblies) to satisfy specific sets of requirements (expressed as architectural policies). Support scores aggregate imperfect information about requirements and components, using as parameters the minimal credibility degree that architects are willing to accept, thus providing a means to explore design spaces using different risk preferences.

The reminder of this article is structured as follows: Section 2 characterizes the problem; Section 3 surveys some related work; Section 4 presents an intuitive formulation of the key Azimut concepts; Section 5 characterizes the architectural imperfect information we can handle; Section 6 reformulates the key Azimut concepts to deal rigorously with imperfect information; Section 7 introduces support scores; Section 8 illustrates the use of the support scores with a realistic example; and Section 9 discusses future work and conclusions.

2 Motivation

The construction of software systems using components offers great promise of reducing development times and costs while increasing quality, but its realization requires that architects be able to choose among alternative solutions composed from available components that best fit the requirements.

In practice architects have at hand incomplete, imprecise, and uncertain information about components, and perhaps even about requirements. Imperfect information can be dealt with using specific mathematical formalisms, such as fuzzy values for imprecision and rough sets for incompleteness. Since architects wanting to compare candidate architectures need to use whatever information is available, however imperfect it may be, they need to compare and rank the non-arithmetic values used to characterize their components.

Thus, a practical and scalable approach to component-sets comparison must have these properties:

1. **Relate component-sets to requirements (specially to NFRs).** It is sometimes quite complex to relate components (and sets thereof) to specific requirements, and specially to NFRs (non-functional requirements) due to their systemic nature.
2. **Record imperfect information.** In real situations, architects have at hand incomplete, imprecise and uncertain component information; indeed, even requirements may be likewise.
3. **Allow direct comparison of whole component sets.** Architects need to play what-if and sensitivity games, which rely on being able to compare alternative solutions, i.e. component sets as wholes agains each other.

This article presents a threefold technique that meets the above requirements: