Chapter 8

Description, Classification and Discovery Approaches for Software Components:

A Comparative Study

Sofien Khemakhem
CNRS, LAAS, & Université de Toulouse, France; University of Sfax, Tunisia

Khalil Drira
CNRS, LAAS, & Université de Toulouse, France

Mohamed Jmaiel
University of Sfax, Tunisia

ABSTRACT

The successfulness of the Component-Based Development (CBD) process relies on several factors, including: the structuration of the component repositories, and the comparison procedures for interface exploring while comparing the expected and the provided services. Both functional and non-functional features should be considered. This chapter presents three key factors to guarantee the successfulness of the repository reuse: the description, the classification and the discovery of components. Through this study, first, the type of information is specified. Next, the style of adopted search is indicated, and then the type of comparison, and the level of specification. Then, the chapter analyzes the importance of non-functional constraints in the description of the components, and studies the advantages and the disadvantages of existing techniques.

DOI: 10.4018/978-1-60960-215-4.ch008

INTRODUCTION

Component-based and service-oriented software architectures are likely to become widely used technologies in the future distributed system development. Component reuse is a crucial requirement for the development process of component-based and service-oriented software.

Components are developed as important and big autonomous and customizable software units. The successfulness of the reuse is important and depends on the efficiency of the search procedure. The search step is essential in the development process, since the developer is generally faced with a significant number of various component types. The search step may fail if the explored component repositories are not appropriately structured or if the required and the provided services are not correctly compared. The use of a component repository, having a well-defined structure, is crucial for the efficiency of the CBD approach. This allows the developer to easily seek and select the component which perfectly meets his/her needs.

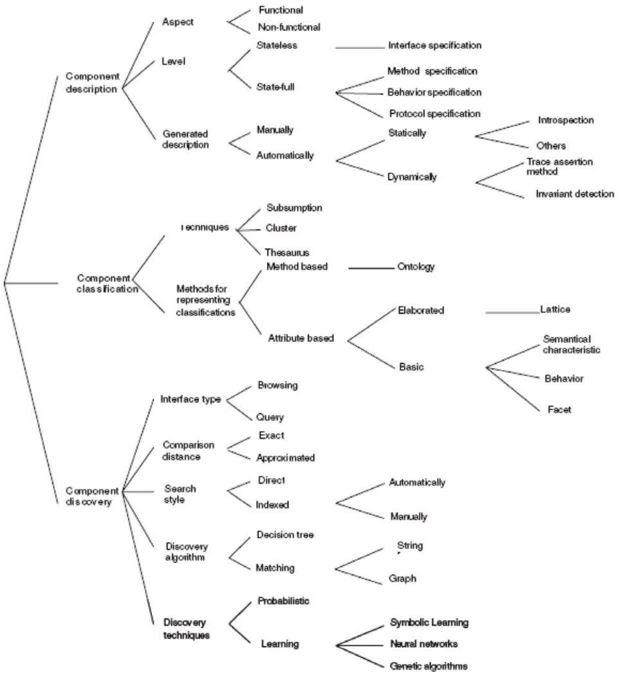
Through this study, we analyze the key factors that are necessary for obtaining a well-organized software component repository and software components having a pertinent description for the search procedures (see Figure 1). These factors act not only on the precision of the specified request but also on the component resulting from the search process.

For component description, two generation approaches of description are distinguished:

Manual generation (Erdur & Dikenelli, 2002) and automatic generation. The automatic generation is relying on different methods such as introspection (Sessions, 1998), (ONeil & Schildt, 1998), trace assertion (Whaley et al., 2002) and invariant detection (Perkins & Ernst, 2004).

The second part identifies and describes five categories of methods for representing component classification. The first is the adhoc method, called also behavioral method (Podgurski & Pierce,

Figure 1.


Component description
Aspect
Functional
Non-functional
Level
Stateless
Interface specification
State-full
Method specification
Behavior specification
Protocol specification
Generated description
Manually
Automatically
Statically
Introspection
Others
Dynamically
Trace assertion method
Invariant detection
Component classification
Techniques
Subsumption
Cluster
Thesaurus
Methods for representing classifications
Method based
Ontology
Attribute based
Elaborated
Lattice
Basic
Semantical characteristic
Behavior
Facet
Component discovery
Interface type
Browsing
Query
Comparison distance
Exact
Approximated
Search style
Direct
Indexed
Automatically
Manually
Discovery algorithm
Decision tree
Matching
String
Graph
Discovery techniques
Probabilistic
Learning
Symbolic Learning
Neural networks
Genetic algorithms