I. Introduction

Software Tuning Panels for Autonomic Control (STAC)
• Automatically re-architects legacy source code for autonomic control
• Source transformation rules written in TXL generate a control panel to centralize access to and modification of tuning parameters
• XML mark-up is inserted manually to identify tuning parameter declarations

What are tuning parameters?
• Scalar fields and scalar properties of structured fields
• Explicit declarations in source code
• Used to influence or observe the behaviour of the system
• Related to metrics such as performance and security

Where are tuning parameters?
• Tuning parameters are not always documented, intentionally or unintentionally
• Variable names provide clues but can be misleading or ambiguous
• Not always explicit (e.g., cache hit rate, tree depth)

Objective
• Automate tuning parameter identification and mark-up using patterns of use

II. Case Studies

Creating a taxonomy
• Studied four server-oriented applications implemented in Java:
  • Apache Tomcat/Jetty (Web/Servlet)
  • Apache Derby/Berkeley DB Java Edition (Database)
  • Catalogued tuning parameters from manuals, source comments, JMX
  • Classified according to usage patterns (Fig. 1)

III. E-R Model

Building the model
• Need to know where and how data are transferred and how data are compared
• Entities abstracted from the Eclipse JDT program model to represent:
  • Types
  • Fields
  • Local variables
  • Methods
  • Relationships represent data transfers and data comparisons between entities (Fig. 2)

IV. Fact Extraction

Abstract Syntax Trees
• Eclipse JDT parser used to generate AST for each compilation unit
• Nodes of AST visited to extract instances of relationships between entities
• Extracted relationships form a directed graph of tuples (Fig. 3)
• Graph patterns used to identify tuning parameters

V. Graph Manipulation

Manipulating the facts
• Operations on binary relationships used to manipulate facts
• Isolated facts from each compilation unit are combined to elicit program understanding
• Inferences create new edges in the graph
• Graph patterns require, allow or disallow particular edges from particular nodes along a path to identify tuning parameters

Example - Statistical Maximum
• Statistical maximums are characterized by an expression being assigned to a variable only when the value of that expression would not cause the value of the variable to decrease

VI. Conclusions

• Tuning parameters can be automatically identified by matching patterns of use from the taxonomy
• Fact extraction and graph manipulation can be used to get existing relationships from source code and infer new ones
• Patterns identify tuning parameters while also providing clues about related expressions

VII. Future Work

• Integration with management frameworks based on standards such as Web Services Distributed Management (WSDM)
• Refine taxonomy based on expert feedback and study of other application domains
• Orthogonal tuning parameter classification based on resource stereotypes

References


This work is supported by an IBM CAS Fellowship and by the Natural Sciences and Engineering Research Council of Canada.