An approach for the evolutionary discovery of software architectures

Aurora Ramírez, José Raúl Romero, Sebastián Ventura


XXI Jornadas en Ingeniería del Software y Bases de Datos (JISBD) Salamanca (Spain). 13-16 September 2016
Content

1. Introduction
2. The search problem
3. Evolutionary approach
4. Experimental study
5. Conclusions

Introduction

- Software architects face complex design decisions
  - Software structure, platforms, styles...
  - Functional and non-functional requirements
  - Few information at this stage of the development

- Search Based Software Engineering
  - Support in decision making
  - Exploration of design alternatives

An approach for the evolutionary discovery of software architectures. SBSE@JISBD’16.
We want to automatically identify the underlying architecture from an analysis model (represented as a class diagram).

It can be a too demanding, complex, and time-consuming task.

Evolutionary algorithms may serve to (semi-)automate the process of finding optimal software architectures.

A extremely high combinatorial problem.
RQ1: Can single-objective evolutionary algorithms help the software engineer to identify an initial candidate architecture of a system at a high level of abstraction?

RQ2: How does the configuration of the algorithm influence both the evolutionary performance and the quality of the returned solution?
Evolutionary approach

Key elements

A flexible and comprehensive tree encoding

A fitness function based on three design metrics

A mutation operator able to perform five architectural transformations

An adaptive mechanism to deal with constraints

fitness(ind) = \begin{cases} 
  r(ICD_{ind}) + r(ERP_{ind}) + r(GCR_{ind}) 
  & \text{if ind is valid} \\
  \#individuals \cdot \#metrics + 1 
  & \text{if ind is invalid} 
\end{cases}

ICD = \frac{\#classes_{ind} - \#classes}{\#classes_{total}} 
\sum_{i=1}^{n} ICD_i

ICD = \frac{1}{n} \sum_{i=1}^{n} ICD_i 

ERP = \sum_{i=1}^{n} \sum_{j=1}^{n} \left[ \begin{array}{c} \cdot n_{ij} + w_{ij} \cdot n_{ij}^q + W_{ij} \cdot n_{ij} + w_{ij} \cdot n_{ij} \end{array} \right]

GCR = \frac{\#groups}{\#components}

An approach for the evolutionary discovery of software architectures. SBSE@JISBD’16.

[6/9]
## Experimental study

### Parameter study

<table>
<thead>
<tr>
<th>Selection</th>
<th>deterministic / tournament / roulette</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replacement</td>
<td>best / competition / elitism / elitism (10%) / binary tournament</td>
</tr>
<tr>
<td>Mutation</td>
<td>[0.1, 0.6]</td>
</tr>
<tr>
<td>Population Size</td>
<td>50, 100, 150, 200</td>
</tr>
<tr>
<td>Stopping criterion</td>
<td>convergence every 1200 evaluations: 20000-24000</td>
</tr>
</tbody>
</table>

### Experimental results

- Optimal or near optimal values for **GCR**
- ICD and ERP require to **strike a balance**
- Without assuming any structure, it can **identify related functional blocks**
- Importance of the **number and types of relationships** among classes
Conclusions

- Evolutionary Computation as an exploratory mechanism to decision support
  - Identify blocks of related functionality
  - Without assuming any structure

- The search approach is close to the architect
  - Flexible and comprehensible representation
  - Architectural transformations with heuristic information
  - Fitness function based on design metrics
An approach for the evolutionary discovery of software architectures

Thanks!

XXI Jornadas en Ingeniería del Software y Bases de Datos (JISBD)
Salamanca (Spain). 13-16 September 2016

Aurora Ramírez

Email. aramirez@uco.es