Pat-Evol: Pattern-Driven Reuse in Architecture-Based Evolution for Service Software

Aakash Ahmad and Claus Pahl

Although architecture-centric maintenance and evolution is useful for adjusting software structure and behaviour at higher abstractions, this approach lacks the potential for systematic change reuse. The Pat-Evol project focuses on managing a constructive architecture-based evolution process for service software. It enables continuous automated identification of evolution patterns from the architecture change log with support for pattern specification and instantiation provided by a pattern library. A pattern library acts as a repository to enable pattern-driven change execution, supporting the notion of an off-the-shelf architecture evolution.

Service Oriented Architecture (SOA) is a business-centric architectural approach representing business processes as technical software services. In this context, architecture-centric maintenance and evolution can be exploited to adapt service process structure and behaviour at higher levels of abstraction. The emergence of change patterns or evolution styles promotes the build-once use-often philosophy in component-based architectural evolution, but often falls short of addressing frequent, demand-driven process-centric change in SOAs. We focus on exploiting recurrent evolution patterns that are specified once and instantiated multiple times thus providing a generic, reusable solution to re-current SOA evolution problems.

The Pat-Evol Project

Pat-Evol - an SFI funded project at Lero, the Irish Software Engineering Research Centre – focuses on enabling pattern-driven reuse in architecture evolution for service software. Within Lero competency areas, the project aims at providing a formal foundation and tool support for the evolution of business-critical software. It enables an automated identification of evolution patterns with repository support for pattern specification and instantiation during evolution. The novelty in the project lies in:

- Exploiting the architecture change log (a history of sequential change) to continuously identify evolution patterns that provide a generic solution to recurring architecture evolution problems.
- Support for pattern specification and instantiation through a **pattern library** that comprises of a *continuously validated and updated collection of patterns* as reusable solutions to architecture evolution problems.
- An **evolution application framework** to enable pattern-based reuse during change execution to support the notion of *off-the-shelf evolution* in service architectures.

At the core of the Pat-Evol project is an evolution pattern identification technique that continuously detects and updates patterns and allows them to be utilized from a library.

**Evolution Pattern Identification**

Pattern identification aims at capturing the recurring primitive architectural changes that occur with change operations (add, remove, modify etc.) on architecture elements (configurations, components, connectors etc.). In order to achieve this, fine-grained instances of sequential architectural changes are retrieved from the architecture change log (ACL). The changes are modelled as a typed attributed graph providing formal semantics with its node and edge attribution to capture change operations on architectural elements. Sub-graph mining in the architecture change graph identifies exact instances of evolution patterns, but also inexact matches where only central pattern features suffice for identification.

In order to facilitate automation, the developed algorithm - G-Pride (Graph-based Pattern Identification) - uses an a priori-based approach to (i) generate and validate the pattern candidates and (ii) determine the occurrence frequency of candidates in an architecture change graph.

**The Pattern Library**

The library is a collection of evolution patterns which provide empirically determined generic and potentially reusable pattern-based solutions to a collection of architectural evolution problems. The primary benefit lies with i) automated population - to eliminate time-consuming and error-prone manual efforts for continuously validating and updating the pattern collection and ii) assisted retrieval - to assist the architects in selecting and instantiating appropriate pattern(s) for a given evolution context.

The Pat-Lib library prototype supports semi-automated specification of identified patterns using pattern templates. This allows architects the query-based retrieval of concrete pattern(s) instances during architecture evolution.
Reusable Change Execution

A framework for pattern-based reuse of change and evolution steps during architecture change management is based on constrained composition of evolution operations on architecture elements enabling their structural evolution. The architecture meta-model is represented as a typed attributed graph with its node and edge attribution capturing configuration among components and connectors. This allows for modelling the evolution operations as (constrained) graph transformation rules to enable structural evolution of architectures.

We are currently developing the eADL (Architecture Evolution Description Language) that allows sequential composition of change to enable pattern-based change execution. The architecture (graph) model is expressed as an XML structure that allows for the execution of evolution operations on architecture elements as the XSLT rules.

Conclusions

Automating the change pattern detections allows us to tailor change management to particular environments and to validate and update these change patterns continuously. The Pat-Evol project aims to provide graph-based formalism and tool support to manage change reuse during structural evolution in service architectures. It provides automation and tool assistance for architects to obtain properly documented and up-to-date descriptions of architectures and their evolution based on common evolution patterns.

Links:

http://www.lero.ie/research
http://www.computing.dcu.ie/~cpahl/

Please Contact:

Claus Pahl
School of Computing, Dublin City University, Ireland

Tel: +353-17005620

E-mail: Claus.Pahl@computing.dcu.ie