FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Diplomarbeitspräsentation

Distributed Systems Group

Decentralized Run-time Architecture Tracking

Masterstudium:
Software Engineering & Internet Computing

Bernd Rathmanner

Technische Universität Wien
Institut für Informationssysteme
Arbeitsbereich: Distributed Systems Group
BetreuerIn: Univ.Prof. Dr. Schahram Dustdar
Mitwirkung: Christoph Mayr-Dorn, Ph.D.

Introduction

Setting

- A distributed application,
- whose architecture might be rapidly changing.

Problems

- Monitoring the local and distributed runtime architecture of such applications.
- Providing probes for different environments to allow for their monitoring.
- A distribution mechanism for the runtime architecture aggregation on specific granularity levels.

Achieving Decentralized Runtime Architecture Tracking

We have created an approach which provides the monitoring and aggregation of an application's runtime architecture using the following steps:

Extraction of architectural properties

Our approach extracts the following properties:

- components and connectors
- local and external links between architectural elements
- the runtime status of an architectural element
- properties about the host where the application is running

These properties are directly used to create the resulting runtime architecture of the monitored application.

Propagation of extracted properties

The extracted properties are:

- Packed into simple events containing all required information of an architectural property.
- Propagated to the aggregation process.

Aggregation of the runtime architecture

Our aggregation process is:

- Directly based on the mentioned events, thus on our architectural properties.
- Using a xADL document to store the resulting runtime architecture.
- Heavily relying on a number of xADL extensions we have created:
  - *xArch Instance Mapping*
  - *xArch External Identified Links*
  - *xArch HostProperty*

Proof of Concept Implementation

We have created a monitoring framework which implements the described features. It is composed of two loosely coupled applications named *Monitor* and *Aggregator*.

Monitor

- Handles the instantiation of an Myx based application using *myx.fw*.
- Extracts all automatically extractable architectural properties.
- Provides methods which enable the extraction of other architectural properties, namely external links and properties about the application's environment.
- Propagates the extracted architectural properties.
- Provides extensions to *myx.fw* that allow for dynamic runtime architecture adoption.

Aggregator

- Aggregates the resulting runtime architecture and persists it using xADL.
- Allows hierarchal composition by utilizing the publish-subscribe pattern which enables a subscriber to receive information on specific granularity levels.

Evaluation

We have evaluated our proof of concept implementation using:

- An audio based publish-subscribe system,
- that runs in a cloud based environment.

We have evaluated our application using the following scenarios:

- Replaying different audio streams.
- Using the message broker as the basis for the *Aggregator* application.
- Handling a rapidly changing architecture with many publishers and subscribers.

Conclusion

We have created an approach which:

- Allows for the instantiation of Myx based applications.
- Provides capabilities for monitoring and
- aggregating the runtime architecture of such applications (including distributed applications).
- Offers methods that support overall runtime aggregation on specific granularity levels.
- Includes utilities for the dynamic creation of architectural elements.

DOWNLOAD

Kontakt: bernd.rathmanner@alumni.tuwien.ac.at