

Jadex: A Generic Programming Model and One-Stop-Shop Middleware for Distributed Systems

Alexander Pokahr, Lars Braubach, Kai Jander

Distributed Systems Group, University of Hamburg
{pokahr, braubach, jander}@informatik.uni-hamburg.de

I. OVERVIEW

The developer of a distributed application is confronted with numerous intricate issues such as concurrency, heterogeneity and security. Many approaches and technologies exist, each of which focuses on addressing some subset of these issues. Yet, developing distributed applications remains a challenge due to a missing coherent conceptual model as well as due to technical and configuration problems. The middleware Jadex has been built to alleviate challenges on both parts by proposing an intuitive *programming model* as well as a mature *one-stop-shop middleware* for many different distributed applications.

II. PROGRAMMING MODEL

Jadex introduces a new approach called *active components* [1] that allows treating all aspects of distributed systems using a unified conceptual model. Active components (AC) combine characteristics of services and agents with a component view. Each active component forms a conceptual unit of state and concurrency and is defined by asynchronous interfaces of required and provided services. Thus, AC offer distribution transparency and a clean concurrency model based on the actor formalism. Dynamic scenarios are naturally supported by relying on runtime service discovery. Furthermore, the explicit component interfaces allow separating functionality from configuration aspects, such as security.

Internally, a component may be composed of service implementations as well as subcomponents. In addition, each component may possess an agent-like proactive behavior, specified using e.g. plain Java objects, reasoning engines for intelligent agents following the BDI (belief desire intention) model, as well as different workflow descriptions.

III. ONE-STOP-SHOP MIDDLEWARE

The Jadex middleware is responsible for all aspects of executing and administering active components. Execution is further facilitated by so called kernels that capture the logic of specific component types, such as BPMN workflows or Java agents. Besides execution in realtime, the middleware also allows for executing applications in simulation time. This is e.g. a powerful tool for testing applications or benchmarking alternative implementations of application functionality.

One main goal of the middleware is ease-of-use concerning both application development and middleware configuration. The middleware follows a configuration by exception strategy, which means that sensible defaults exist for all configuration

options. For example each Jadex instance by default automatically discovers globally (via public relays) and locally (via multicast) available peer instances to connect to. Yet, security settings are automatically set in a way that local testing of applications is not restricted while no sensitive services are exposed to untrusted hosts.

Application development is further supported by a suite of tools ranging from an administration view to specific debuggers for inspecting components at runtime. All tools are able to operate remotely, which is especially advantageous when debugging application parts, which run on back-end servers or mobile devices with limited user interface capabilities.

IV. EXEMPLARY USE CASES

Jadex is e.g. used as a basis for a commercial product, currently developed by the company Uniiue AG. The product, called DiMaProFi (Distributed Management of Processes and Files) will allow for defining and executing distributed ETL (extract, transform, load) workflows that are used to process and transport data and files scattered at different places in an IT infrastructure to a central data warehouse. Uniiue internally already uses Jadex to automatically execute their own ETL processes for several customers and in this way got rid of many manual activities that were concerned with executing shell scripts and copying files between different hosts. Features of Jadex that have proven especially useful in this scenario are the capability of modelling and executing distributed workflows in BPMN, automatically building an overlay network from different hosts also within highly restricted environments, and platform security shielding from outside access.

Another application has been developed together with Daimler AG in order to automate long-running production planning processes. The approach is based on two-levels of workflows: a stable control layer based on process goal hierarchies described with the goal-process modelling language (GPMN) [2] and a dynamic execution layer based on standard BPMN processes. In this project it was helpful that different kinds of components (here GPMN and BPMN) could be executed within the same infrastructure interacting seamlessly. Additionally, as some of the system users were equipped with mobile devices, awareness and a fast binary transport mechanism were considered advantageous.

REFERENCES

- [1] L. Braubach and A. Pokahr. Developing Distributed Systems with Active Components and Jadex. *Scalable Computing: Practice and Experience*, 13(2):3–24, 2012.
- [2] K. Jander, L. Braubach, A. Pokahr, W. Lamersdorf, and K.-J. Wack. Goal-oriented Processes with GPMN. *International Journal on Artificial Intelligence Tools (IJAIT)*, 20(6):1021–1041, 12 2011.