JACK autonomous agents extension

<!-- -->

Table of contents

1. Datasheet

Solution data	
Name	JACK autonomous agents extension
Result type	Platform
Description/functionality	This result covers extensions to the existing JACK agents platform to work with Autonomous Agents. The aim of the extended JACK agent framework for Web services is to provide a goal-oriented service composition and execution module within an SOA. Following the Belief Desire Intention (BDI) model, agents are autonomous software components that have explicit goals to achieve or events to handle (desires). Agents are programmed with a set of plans to describe how to go about achieving desires. Each plan describes how to achieve a goal under varying circumstances. Set to work, the agent pursues its given goals (desires), adopting the appropriate plans (intentions) according to its current set of data (beliefs) about the state of the world. The combination of desires and beliefs initiating context-sensitive intended behaviour is part of what characterises a BDI agent. BDI agents exhibit reasoning behaviour under both pro-active (goal directed) and reactive (event driven) stimuli. Adaptive execution is introduced by flexible plan choice, in which the current situation in the environment is taken into account. A BDI agent has the ability to react flexibly to failure in plan execution. Agents cooperate by forming teams in order to achieve a common goal. The JACK agent platform is not inherently ready for interaction within a Web service environment. Additional steps are necessary for enabling interactions between the agent platform and Web services, especially when the agents themselves offer services Following the MDA approach, a modeller specifies at design time a set of plans (PSM level) that constitute the workflow library of the agents. Web service calls are integrated as steps into plans. Workflows are modelled graphically and most of the common workflow patterns are supported. In order to prepare for a transformation from a

	PIM4SOA model to the JACK platform specific model (PSM), service providers are mapped to JACK agents/teams. The parts of the platform independent model (PIM) which define the processes involved are mapped to agent/team plans and correlated events, whereas the parts which define the interfaces are mapped to the modules which provide the client- and server-side code for the JACK agent platform.
Benefits to interoperability	Just like BPEL, the JACK framework supports fixed composition, where the structure and the components of the composition are statically bound, and semi-fixed compositions, where the structure is statically bound but the actual service bindings are performed at runtime. More explorative compositions, where both structure and components are created at runtime, are beyond what BPEL or BDI agents can offer. However, there are several advantages to BDI agent, especially when it comes to handling failures at runtime. A plan is executed in a context which specifies conditions for plan instances and also other applicable plans. An exception in one plan instance then leads to the execution of another plan instance for the next known service. The BDI agent approach supports this adaptive behaviour in a natural way, whereas a BPEL process specification which attempts to provide the same behaviour would require awkward coding such as nested fault handlers. Another advantage is that extending the behaviour by adding a new plan for a specific task simply means adding it to the plan library for it to be executed at the next opportunity. Similarly, customizing the composition is facilitated since the different plans clearly structure the alternatives of possible actions. Since the control structure is implicit, changes in a plan do not have impact on the control structure.
Supported models/methodologies	-
Supported input interfaces	-
Supported output interfaces	-
Validation/demonstration	Will be done together with the investigation of the model mapping in the evaluation of the PIM4SOA to Jack transformation.
Standards compliance	• http://www.fipa.org/http://www.fipa.org/ • http://www.agentlink.org/http://www.agentlink.org/

Availability	-
License	-
Status	-
Requirements/dependencies	-
Web references	-
ATHENA metadata	
Contact person	Klaus Fischer, DFKI
Contributors	DFKI
Provided by project/activity	• A5 – Planned and Customisable Service-Oriented Architectures • A6 – Model-driven and Adaptive Interoperability Architectures
Deliverables representing result	• D.A5.3: Architecture of SOA platforms (M21) • D.A5.4: Execution Framework(s) for Planned and Customisable Service-Oriented Architectures (M21)
Contribution to key result	• 8: Interoperability Infrastructure • 12: Service Composition Framework • 13. Model-driven and Adaptable Interoperability Framework and Infrastructure
Used in pilot	-
Deliverable providing evaluation	D.A5.5 “Validation of Research Results” (M24)