An Eclipse-based IDE for Agent-Oriented Programming in simpAL

Andrea Santi[1], Alessandro Ricci[1]
a.santi@unibo.it, a.ricci@unibo.it

Dipartimento di Informatica Scienza e Ingegneria (DISI),
Alma Mater Studiorum—Università di Bologna[1]

ECLIPSE-IT 21/09/2011, Pozzuoli

Outline

Developing Modern Software Systems: Challenges

- Developing modern software systems is an hard task
- Almost every modern application has to deal with issues such as
  - Concurrency
  - Distribution
  - Decentralisation of control
  - ...
- The free lunch is over [Sutter and Larus, 2005]
- Towards a paradigm change in computer science [?]

How to Program Such Systems?

- It is now important to introduce higher-level programming abstractions [Sutter and Larus, 2005]
  - Easing the development of modern software systems
  - Like OO abstractions help(ed) build large component-based programs
- Proliferation of new programming languages/frameworks aiming at ease this task
  - Rooted on not so mainstream (*yet*) programming models
    - The Actor [**?**] model one is a main example
  - Scala [**?**], Groovy [**?**], Clojure [**?**], etc.

Agent-Oriented Programming (AOP):the Current Situation

- The idea of Agent-Oriented Programming is not new
  - The first paper about AOP is dated 1993 [Shoham, 1993]..
  - .. and since then many APLs and platforms have been proposed [Bordini et al., 2005, Bordini et al., 2009, Bordini et al., 2006]
- Main acceptations are the (D)AI contexts
  - Agents as a special purpose technique to build intelligent systems
- No significant impacts on mainstream research in programming languages and software development
- Emphasis put on theoretical issues
- No focus on principles of general-purpose computer programming

Our Perspective

AOP can be exploited for programming modern software systems in general [Ricci and Santi, 2011]

- Extending object/function-oriented programming
  - Thanks to an higher-level of abstraction
- Tackling main challenges of modern software development
  - Concurrency
  - Decentralization of control
  - Autonomy
  - Adaptivity
  - ...

simpAL Project Overview 1/2

Background/motivations: the free lunch is over [Sutter and Larus, 2005]

- How to model & organize concurrent, reactive programs scaling with their complexity?
- Mainstream paradigms (e.g., OOP) are not enough
  - Looking for a proper level of abstractions

General Research Objective

- Exploring paradigms based on agent-oriented abstractions
  - From agent programming from the perspective of AI
  - To general-purpose software development viewpoint

Method

- simpAL programming language & related tools/ecosystem
- Strongly inspired from existing work in MAS
  - JaCa [Ricci and Santi, 2011] & *JaCaMo* [**?**]
  - simpA [**?**]

simpAL Project Overview 2/2

- simpAL programming language
  - First-class agent-oriented programming abstractions for defining the organization and control of MASs
  - General-purpose for concurrent & distributed programming
  - Compiled & statically typed language
- simpAL platform
  - Distributed runtime
  - IDE &tools
- simpAL core language & formalization
  - Agent-oriented core language & calculus for studying properties

AGERE! @ Splash Initiative

- International ACM Workshop on Programming based on Actors, Agents, and Decentralized Control
  - 2nd edition in 2012, running in October
  - http://agere2012.apice.unibo.it

Main Objective

Soliciting discussion & interaction among people interested in novel programming paradigms promoting a decentralized-control mind-set in solving problems & developing solutions

- Actor-based and agent-oriented approaches as "natural" starting points
- Focus on the theory and the practice of programming

IDE Requirements

What are the requirements for a JaCa IDE?

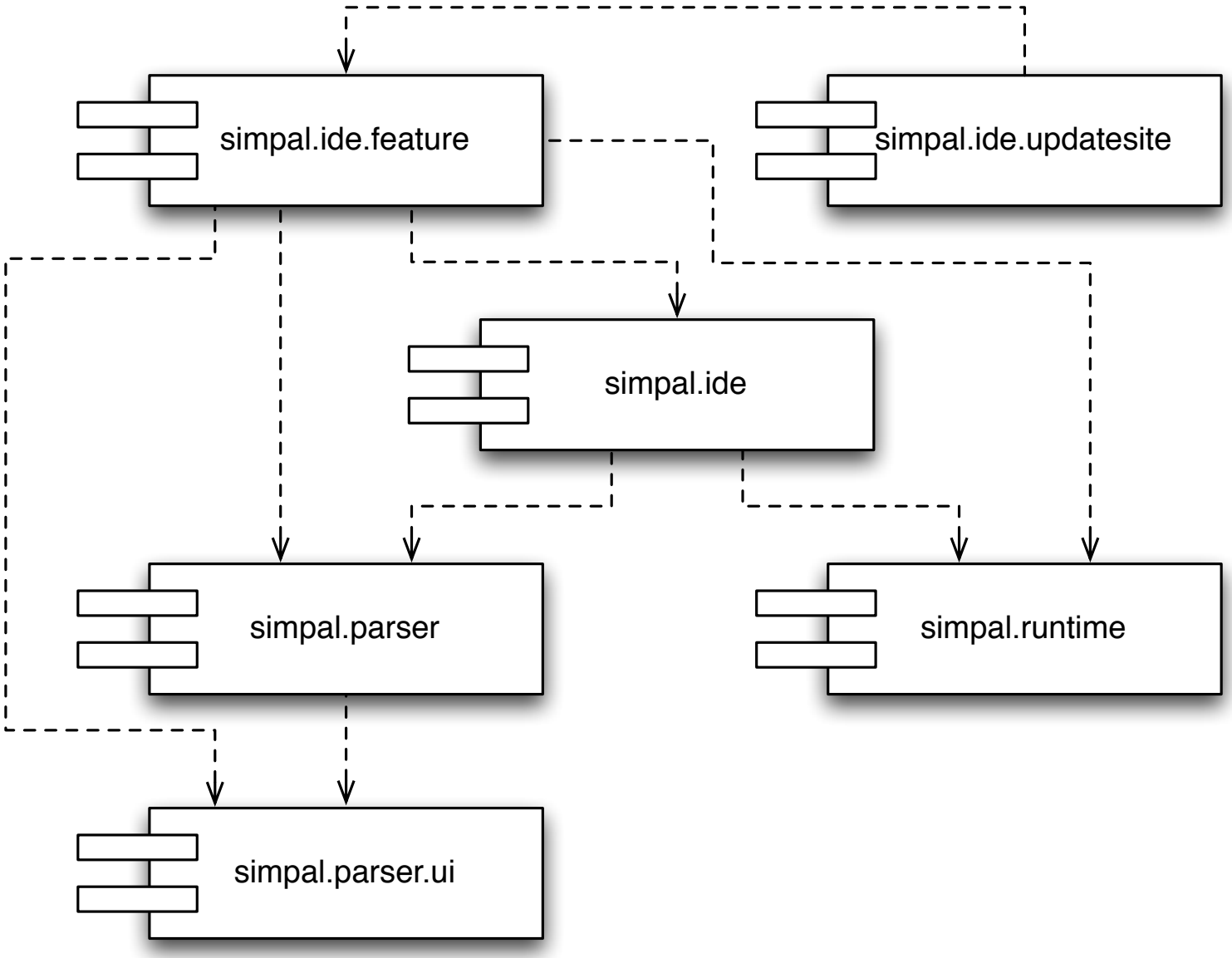
A group of classical requirements for a modern IDE

- Project management and exploring
- Full-fledged file editors
- A set of views able to improve coding experience
- ...

A group of advanced requirements specific to AOP (and JaCa)

- A powerful and innovative debugger
  - Exploiting AOP's abstractions to ease debugging tasks
- Management of distributed applications deploy/execution
  - To simplify and automatize tedious tasks
- Proper organisation and representation applications' topology

IDE Architecture



Bibliography I


Bordini, R., Braubach, L., Dastani, M., Seghrouchni, A., Gomez-Sanz, J., Leite, J., O'Hare, G., Pokahr, A., and Ricci, A. (2006).
A survey of programming languages and platforms for multi-agent systems.
In *Informatica 30*, pages 33–44.


Bordini, R., Dastani, M., Dix, J., and El Fallah Seghrouchni, A., editors (2005).
Multi-Agent Programming Languages, Platforms and Applications - Volume 1, volume 15 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*. Springer.

Bibliography II


Bordini, R., Dastani, M., Dix, J., and El Fallah Seghrouchni, A., editors (2009).
Multi-Agent Programming Languages, Platforms and Applications - Volume 2, Multiagent Systems, Artificial Societies, and Simulated Organizations. Springer.


Ricci, A. and Santi, A. (2011).
Agent-oriented computing: Agents as a paradigm for computer programming and software development.
In *Proc. of the 3rd Int. Conf. on Future Computational Technologies and Applications – Future Computing 2011*, Rome, Italy. IARIA.


Shoham, Y. (1993).
Agent-oriented programming.
Artificial Intelligence, 60(1):51–92.

Bibliography III


Sutter, H. and Larus, J. (2005).
Software and the concurrency revolution.
ACM Queue: Tomorrow's Computing Today, 3(7):54–62.