

ObjectWeb

An Introduction

Jean-Bernard STEFANI

INRIA



- ➔ **What is it ?**
- ➔ **History**
- ➔ **Motivations**
- ➔ **Objectives**
- ➔ **Organization**
- ➔ **Current Status**
- ➔ **Technical Perspectives**
- ➔ **Conclusion**

What is it ?

➔ Open Source initiative

- Origin FT R&D, Bull/Evidian, INRIA
- Initial code base: Jonathan, JOnAS, Joram

➔ Developing Open Source middleware

- Apache or Linux of distributed middleware
- Software available under LGPL-like licences

➔ Since Jan. 2002

- International Consortium, hosted by INRIA

➔ In the works

- ITEA Osmose
- FP6 Integrated Project

➔ Created end 1999

- Partners : France Télécom R&D, INRIA, Evidian (then BullSoft)
- Initial code base: Jonathan, JOnAS

➔ Extension in 2000

- New partners: Libelis, Lutris (Enhydra), Experlog
- New code: Joram, RMIJdbc

➔ Consolidation and development in 2001

- New partners: Schlumberger, Kelua, Scalagent, other start-ups, academic partners, ...
- New code: JORM, OpenCCM

➔ Interests of Open Source software

- Infrastructure and commodity software
- Means for distribution and experimentation, adaptation to various requirements
- Development of new business models around Open Source software

➔ Limitations of existing Open Source platforms

- Architecture
- Functional coverage
- Industrial quality



Objectives

➔ Technical objectives

- To develop a distributed component-based middleware technology
- In the line of the CORBA and Java standards

➔ Community objectives

- To develop a community of developers and users around ObjectWeb components and platforms
- In the Free Software / Open Source tradition (FSF, Apache, etc.)

Technical objectives

➔ Integration, manageability, composability

- Architecture and frameworks
- Components
- Management and re-configuration

➔ Compatibility with standards

- CORBA: CORBA 3.0, CCM
- Java: EJB 2.0, JDO
- Interoperability: HTTP, SOAP, JDBC, LDAP

Technical objectives

➔ Functional coverage

- Naming, trading
- Communication: remote operations, events, messages, multicast
- Availability and safety: transactions, persistence, replication, fault tolerance
- Load balancing
- Security

➔ Robustness, optimisation and code quality

- Benchmarks, tests, evaluations, demonstrators
- Development tools

Community objectives

➔ Education and teaching

- Support for courses, on-line documentation
- Projects
- Leveraging National and European universities

➔ Contributors and users

- Web infrastructure
- Seminars and conferences
- Leveraging National and European middleware R&D

Community objectives

➔ Industrial relations

- Standardization
- Distribution, integration, services
- Network of industrial and start-up companies

➔ Communication and promotion

- Web site
- Free Software / Open Source lobbying
- Lobbying



Organization

- ➔ **Projects**
- ➔ **Members (legal entities or individuals)**
- ➔ **Board**
- ➔ **College of architects**
- ➔ **Executive committee**



Organization

➔ ObjectWeb Project

- Development activity within ObjectWeb
- Project code base
- Project leader
 - Oversees the evolution of the project code base
 - Decides on new releases
 - Manages the development of the project code base

➔ Board

- Elected by all members
- Meets every 3 months
- Approves new projects
- Defines the Consortium's strategy
- Coordinates communication and promotion actions for the Consortium
- Appoints the Executive Committee

➔ Executive Committee

- Daily management of the Consortium
- Permanent representation of the Consortium
- Current composition
 - S. Krakowiak (INRIA – teaching and education)
 - A. Lefebvre (FT R&D - communication)
 - Mathieu Peltier (INRIA - infrastructure)
 - G. Vandôme (INRIA)

➔ College of architects

- Project leaders and co-opted individuals
- Defines the architecture and overall evolution of the ObjectWeb code base
- Decides the creation of new projects (in agreement with the Board)
- Resolves difficulties or technical conflicts that may occur between projects of the Consortium
- Approves major versions of ObjectWeb code base

➔ College of architects

➤ Current composition

- Joram : A. Freyssinet (Scalagent)
- JOnAS : F. Exertier (Evidian)
- Jonathan : B. Dumant (Kelua)
- RmiJdbc : P.-Y. Gibello (Experlog)
- JORM: P. Déchamboux (FT R&D)
- OpenCCM: P. Merle (LIFL)
- Proactive : D. Caromel (U. Nice)
- J.B. Stefani (Inria)
- S. Krakowiak (UJF)
- G. Vandome (Inria)



Current Status

➔ **Infrastructure**

➔ **Notoriety**

➤ Industrial use of ObjectWeb

➤ Usage

➔ **Current members**

➔ **Projects**



Infrastructure

➔ Web site <http://www.objectweb.org>

- General information
- Information on each project
 - documentation, downloads, history, etc.
- Common services
 - Source code management: CVS
 - archived mailing lists
 - FAQ
 - search engine



Industrial use

➔ Bull

- Evidian: uses JOnAS and JORAM in the Orchidee project (embedded in new Telco OEM products and platforms), commercial JOnAS support offer
- ISD: Open Source Linux + JOnAS offer

➔ Schlumberger

- JOnAS application for Smart Card management

➔ France Télécom R&D

- ObjectWeb components as experimentation platform

➔ Lutris

- Integration of JOnAS in Enhydra Enterprise

➔ Libelis

- Integration of JOnAS in Orcas



Industrial use

➔ Experlog

- Expershop product and services

➔ Kelua

- Mosaik tools and associated services

➔ Scalagent

- Framework for gathering and analysing distributed data

➔ ObjectWeb technologies used by numerous start-up companies such as:

- Kovair
- E2fb2
- Venetica
- High Sierra Technologies



Usage



➔ **Wide community of users and developers**

- Leading product: JOnAS
- Hundreds of developers and thousands of users
- Used in at least 50% cases in operational applications (Kovair, DeutschPost, etc.)
- ObjectWeb “inside”
 - “Enhydra Enterprise” = Enhydra + JOnAS (Lutris)
 - “Orcas” Libelis Server = JOnAS + Versant (Libelis)



Projects

- ➔ **Jonathan**
- ➔ **JOnAS**
- ➔ **JORAM**
- ➔ **JORM**
- ➔ **OpenCCM**
- ➔ **Proactive**
- ➔ **RmiJdbc**



Technical perspectives

- ➔ **Architecture**
- ➔ **Functional coverage**
- ➔ **Standards**
- ➔ **Steps (architecture)**
- ➔ **Conclusion**

➔ Vision

- component-based middleware code base
 - objects + dynamic configurability (needs introspection)
 - bringing together work on configurable distributed systems, software architecture and software configuration management
 - e.g. insights from architecture description languages, reflective systems, workflow management, etc.
- with common software frameworks and shared code base
 - used for building different distributed platforms (e.g. telecoms, transport, e-commerce, enterprise intranets, community nets, ...)
 - common frameworks applied to consistently solve recurring problems (e.g. multiprotocol support, resource multiplexing,...)

➔ Evolve code base into fine-grained middleware components

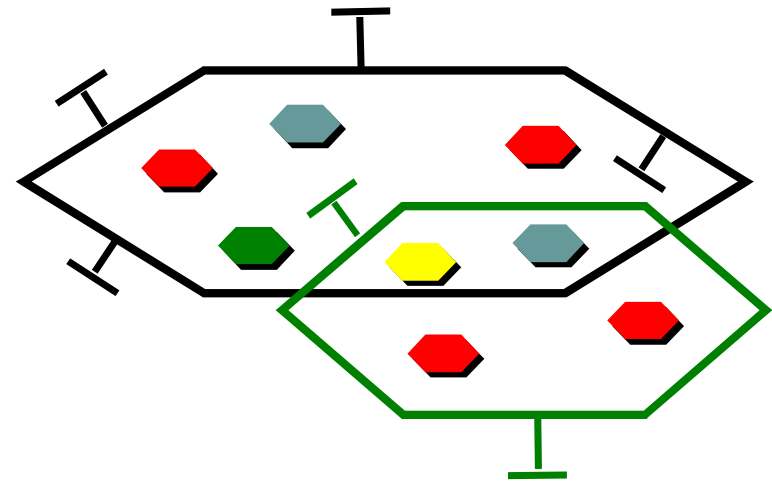
- Core component model (subsuming EJB, CCM, .Net)
- Common frameworks
- Re-engineering OW code base

➔ Improve code base properties

- Protection & Isolation
- Scalability
- Manageability
- Consistent flexibility

➔ Towards a recursive component model

- Controlled introspection
- Behavioral & structural reflection
- Component = {membrane, plasm}
 - Membranes
 - Filter, locate, control, abstract, compose
 - {interfaces, behavior}
 - Plasms
 - Interacting Components
 - May overlap



➔ Towards a recursive component model

- no pre-defined membrane behavior ; membranes generalize containers
 - membranes can be programmed
 - membranes implement composition and containment operations
- no pre-defined component life-cycle
 - membranes can support different semantics for deployment, installation, initialization, activation, suspension, passivation, etc.
- distributed components with no pre-defined granularity
 - from objects to sites and networks

➔ Common frameworks

- Naming
- Binding
- Component
 - including configuration, containment and life-cycle
- Resources
- Protocols

Functional coverage

➔ **Distributed system management**

- Monitoring
- Configuration management (software, components, resources)
- Fault & Performance management

➔ **Availability**

- Clustering & Load balancing
- Caches & Replication
- Recovery
- Extended transactions



Functional coverage

➔ Security

- PKI
- Secure Network Overlays

➔ Distributed persistence

- Heterogeneous data sources
- Requests
- Ad-hoc federation

Functional coverage

➔ Small systems

- “Lightweight” versions of main OW macro-components
- Embedded systems
- Real-time systems

➔ Workflow

- Extended transactions
- Distributed control and superimposition



Standards

➔ Sun

➤ EJB, JDO, JCA, etc.

➔ OMG

➤ CORBA 3.0, Components, etc.

➔ W3C

➤ UDDI, SOAP, WSDL, WSFL, etc.

➔ Pragmatism & Agnosticism

First architecture steps

➔ Component model

- OW specification with abstract model and concrete component framework instantiation
 - leverage : EJB and CCM models, Jonas & OpenCCM tools, Jonathan component framework

➔ Leverage existing frameworks

- Jonathan : naming & binding, protocols
- Document and exploit (e.g. Jonas & Joram multiprotocol support, support for SOAP and WSDL)



First architecture steps

➔ **Populate shared code base**

- Monolog framework (tracing)
- Component, naming, binding & protocols frameworks
- Meta-programming tools
- Transaction management engine

➔ Resources

- JVM technology
 - Isolation & protection
 - Real-time & embedded Java
- Connection to OS
 - Isolation & protection
 - Resource containers
 - Real-time, embedded



Conclusion

- ➔ **Establish ObjectWeb as open source reference for middleware standards (Java, CORBA, Web services)**
- ➔ **Architectural innovation & integrity**
- ➔ **Standards agnosticism**
- ➔ **Architecture and code base requires naturing and nurturing**

➔ **Ambitious initiative**

- Technology
- Community
- Coverage

➔ **Opportunistic initiative**

- Resources
- Themes
- Current offering
- Open Source notoriety

➔ **Difficult initiative**