Activity Service Overview

Mark Little

October 17-20, 2000

Bluestone SOFTWARE Enterprise Interaction Management Arjuna Labs

What is the “activity service”?

- Additional Structuring mechanisms for the OTS
  - OMG adopted specification (orbos/2000-06-19)
    - IBM, University or Newcastle (Bluestone Arjuna Labs), IONA, Inria, Alcatel, Vertel/Expersoft, Bank of America
- Main work provided by IBM and Newcastle University
  - 12+ months to develop and guide through OMG process
- Inria provided an example of use

EXI

Problem statement

- Transactions imply all ACID properties
- Good for “short” durations
  - Application specific
- Long-running transactions may impose constraints
  - Hours, days, months, …
  - Retain resources for duration of transaction
- Should build on existing transaction standard

Structuring transactions

- Could structure transactional applications from short-duration transactions
  - Release locks early
  - Resulting application may still be required to appear to have “ACID” properties
    - May require application specific means to restore consistency
- A transactional workflow system could be used to script the composition of these transactions

Extended transaction models

- There are a number of such models
  - Sagas
  - Compensations
  - Epsilon Serialisability
  - Versioning Schemes
  - Nested top-level transactions
  - Open-nested transactions
  - Glued transactions
  - Coloured actions

Which model to use?

- One size does not fit all!
- Business domains will impose different requirements on implementers
  - Essentially construct domain-specific models
  - Realtime
- The range and requirements for such extended models are not yet known
  - Do not restrict implementations because we don't know what we want yet!

Basic assumptions

- Models share a basic underlying assumption of “event signalling”
  - Specific model maps event into its “domain”
    - e.g., “prepare”, “rollback”
- OTS transactions may be used as building blocks
  - Transactions may be ignored if required
- Some level of interoperability between models may be possible
  - Signal-mapping

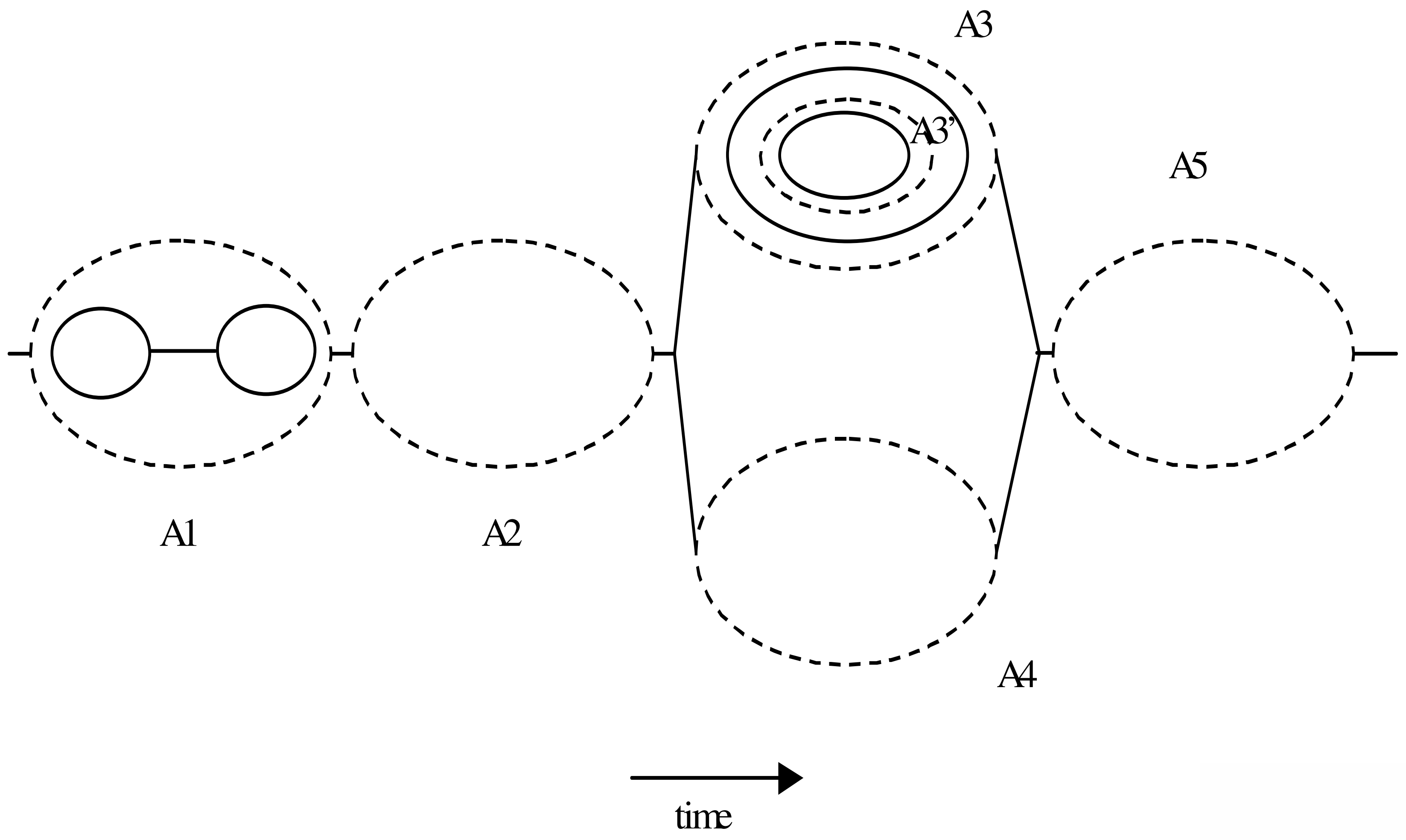
CosActivity module

- Defines a generic protocol engine
  - Support basic infrastructure for many extended models
- Pluggable coordination and control engine
- Activity service interfaces not typically for application programmers
  - Too low-level
  - Requires high-level API on top

Activities

- Activity is an entity that does some work
  - Activities can be nested
  - May use transactions during parts of its lifetime
- Ensure transaction contexts (and extended context data) flow between address spaces
- Threads can run within activity context as well as transaction context
- Recoverable

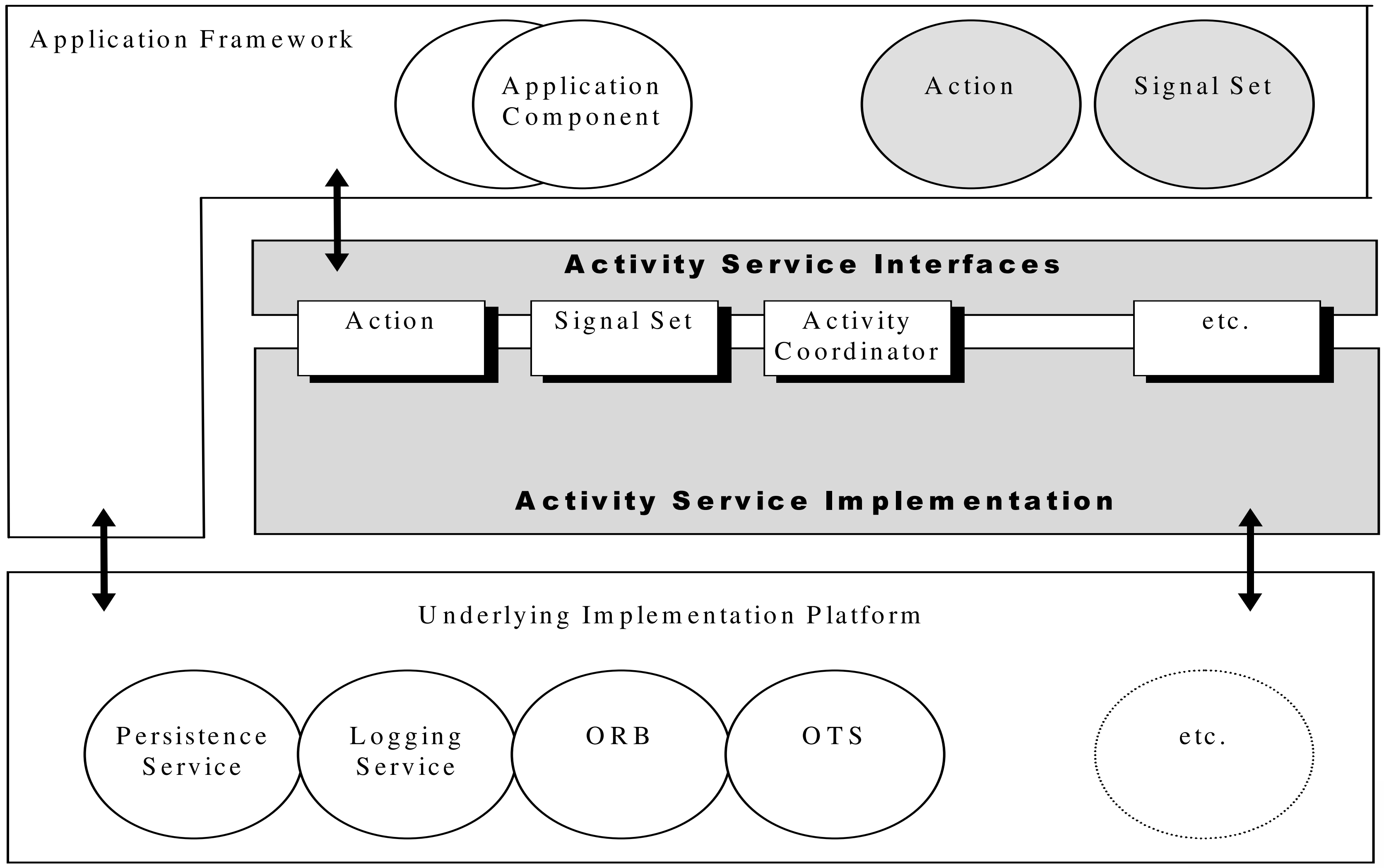
Activities and transactions

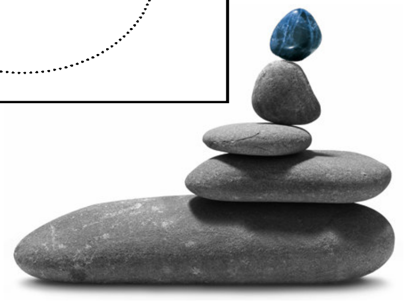


Actions and signals

- Send “signals” between entities (actions)
  - Some “signals” are pre-defined
  - Most are defined by the extended model and dealt with at that level
- Signal factory is pluggable and adaptable
  - Responsible for interpreting results of signal processing
- Infrastructure does not know how to interpret signals or responses to them

Architecture





Bluestone SOFTWARE Arjuna Labs
Enterprise Interaction Management

Future directions

- New transaction models being proposed
  - Domain specific (telecoms, banking)
  - Active area of interesting research
- Lots of interest within OMG members
  - Key participants already have requirements
  - Glue for transaction domains, b2b
- UML profile for transactions and extended transactions

Bluestone SOFTWARE Arjuna Labs
Enterprise Interaction Management

Conclusions

- JCP process already in the works
- OTS-NG
- General principles be the basis for other environments
  - Mobile
  - Web
- We have the first 100% pure Java implementation
  - Prototype completed and tested