Hermes:

An Agent-based Middleware for Mobile Computing

Flavio Corradini (1) and Emanuela Merelli (1,2)

(1) Dipartimento di Matematica e Informatica
Università di Camerino, IT
(2) Department of Computer Science
University of Oregon, USA

Bertino 30 April 2005

What is HERMES?

- Hermes is a software tool (80 kb of Java code)

- Hermes supports the execution of software applications. The applications it supports can be:
  - Activity-based
  - Distributed

Main Aim for Developing Hermes

For the application domains we are interested in applications must be as much as easy and intuitive possible to specify

The programmer of applications has to concentrate on the involved activities and can ignore implementation strategies or metodological development issues

Hence, we also demands to the (automatic) tool the bridging of the gap between an application specification and the application itself

Then…

Hermes can be easily used even by non computer practitioners…

and, indeed, Hermes was asked by those people.

… but at a lower level of abstraction can be used by experts in the area (Hermes as a middleware for Klaim specifications).

Hermes Design Decisions

Hermes has been designed according to **component-based methodology**. This guaranteed a needed modularity to meet the continuous changes of requirements coming from "non initiated" final users

To dominate "application complexity" Hermes relies on the "agent technology" as implementation strategy.

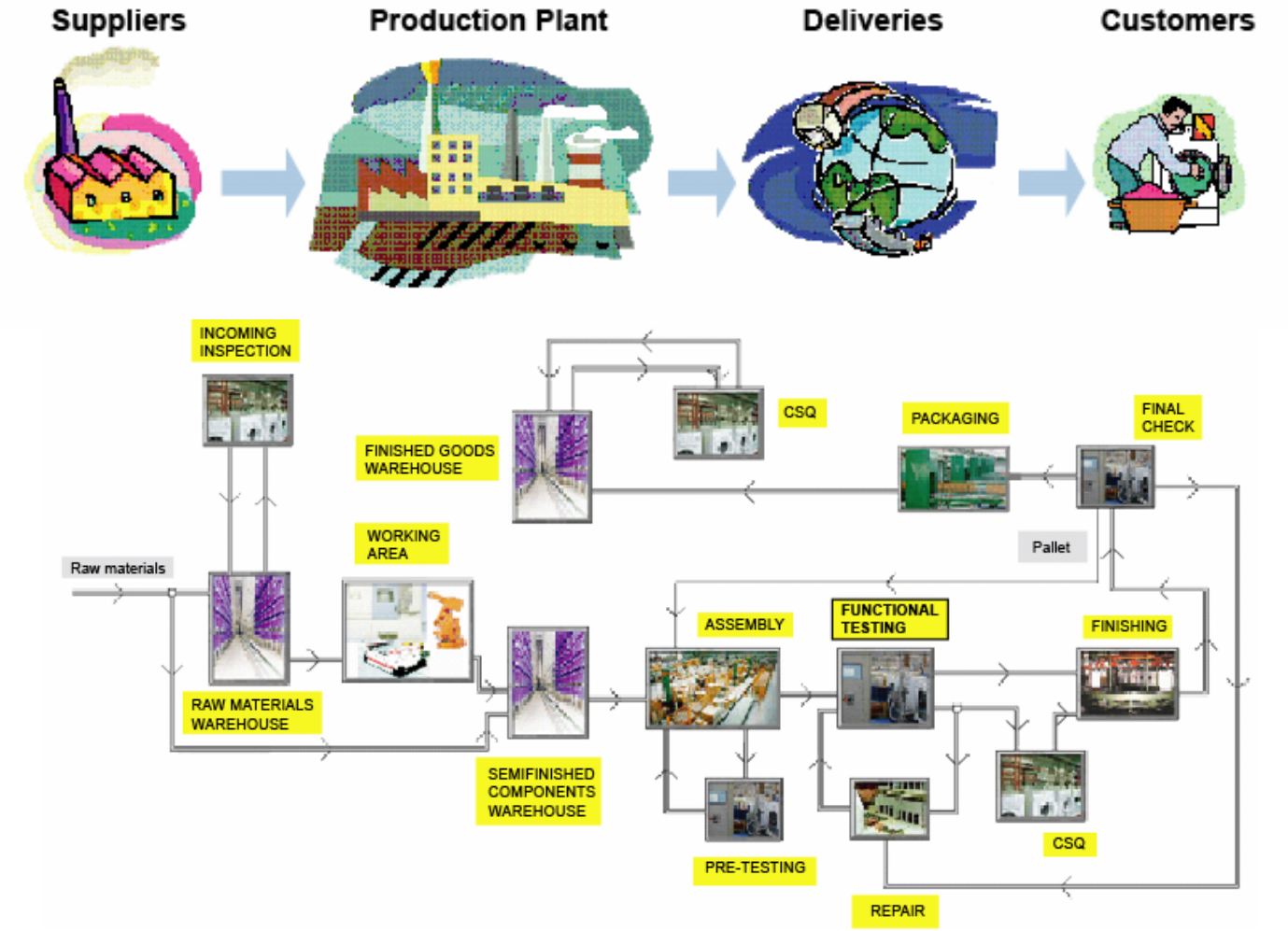
Agents are software components that:

- embed a "complete" behaviour in their control,
- are autonomous and react to external stimuli and environment
- can be mobile (in the sense that they can move from a site another)

They supports modular and compositional design of complex applications.

Code mobility was also very useful in specific application domains. E.g., applications that run in devices with very limited computational resources (e.g. embedded software)

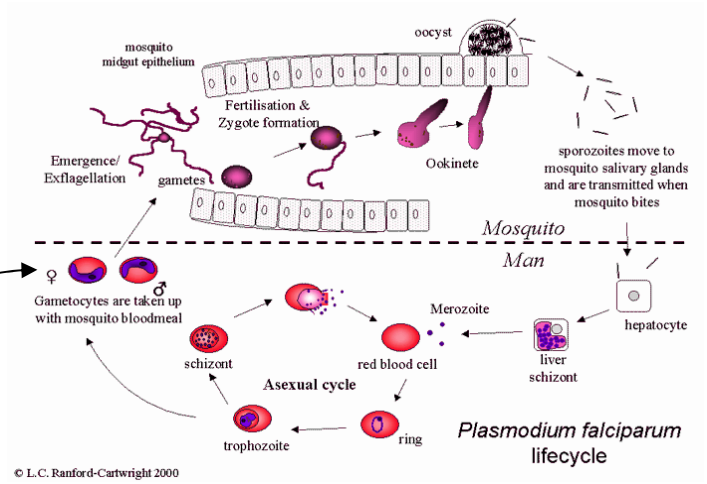
Scenario 1: Industrial control



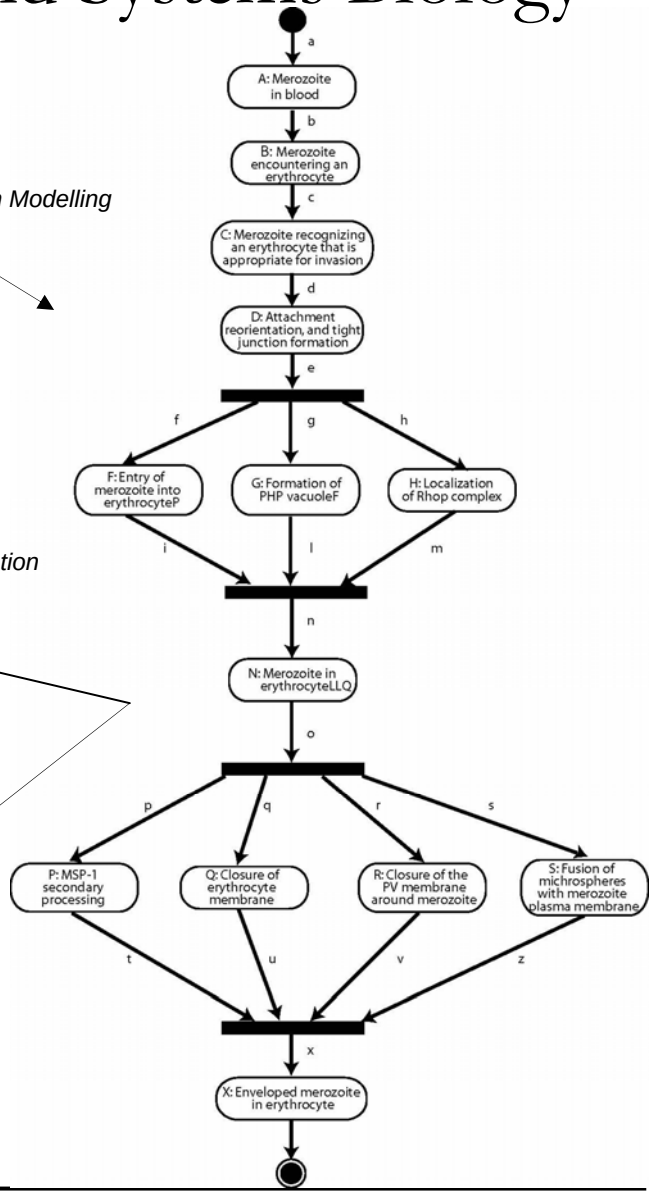
Scenario 1: Industrial control

- **Application Domain:**
  - testing of appliance (washing machine etc.)
  - telemonitoring & teleassistance
- **Execution environment**
  - federated enterprises geographically distributed
  - heterogeneous data format
  - complex industrial production plants
- **Why mobility**
  - tracing products
  - to update software in remote computational stations
  - to avoid transfer of proprietary data
  - to deal with resources with limited computational capabilities

Scenario 2: Bioinformatics and Systems Biology



System Modelling



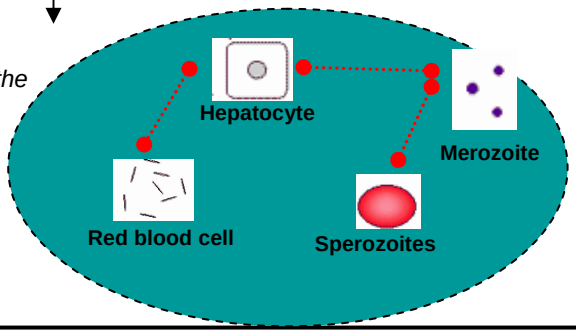
Model Validation

$$(\mathtt{a},\mathtt{A}) \longrightarrow \mathtt{SKIP};\ (\mathtt{b},\mathtt{B}) \longrightarrow \mathtt{SKIP};\ (\mathtt{c},\mathtt{C}) \longrightarrow \mathtt{SKIP}\ ;\ (\mathtt{d},\mathtt{D}) \longrightarrow \mathtt{SKIP}\ ;\ |||_{i=1}^{3} P_i;$$
$$(\mathtt{n},\mathtt{N}) \longrightarrow \mathtt{SKIP}\ ;\ |||_{j=1}^{4} Q_j\ ;\ (\mathtt{x},\mathtt{X}) \longrightarrow \mathtt{SKIP}$$

where $|||_{i=1}^{3} P_i = (\mathtt{f},\mathtt{F}) \longrightarrow \mathtt{SKIP};\ (\mathtt{g},\mathtt{G}) \longrightarrow \mathtt{SKIP};\ (\mathtt{h},\mathtt{H}) \longrightarrow \mathtt{SKIP}$
and $|||_{j=1}^{4} Q_j = (\mathtt{p},\mathtt{P}) \longrightarrow \mathtt{SKIP};\ (\mathtt{q},\mathtt{Q}) \longrightarrow \mathtt{SKIP};\ (\mathtt{r},\mathtt{R}) \longrightarrow \mathtt{SKIP};\ (\mathtt{s},\mathtt{S}) \longrightarrow \mathtt{SKIP}.$

MAS Model Validation

Validation of the experimental results

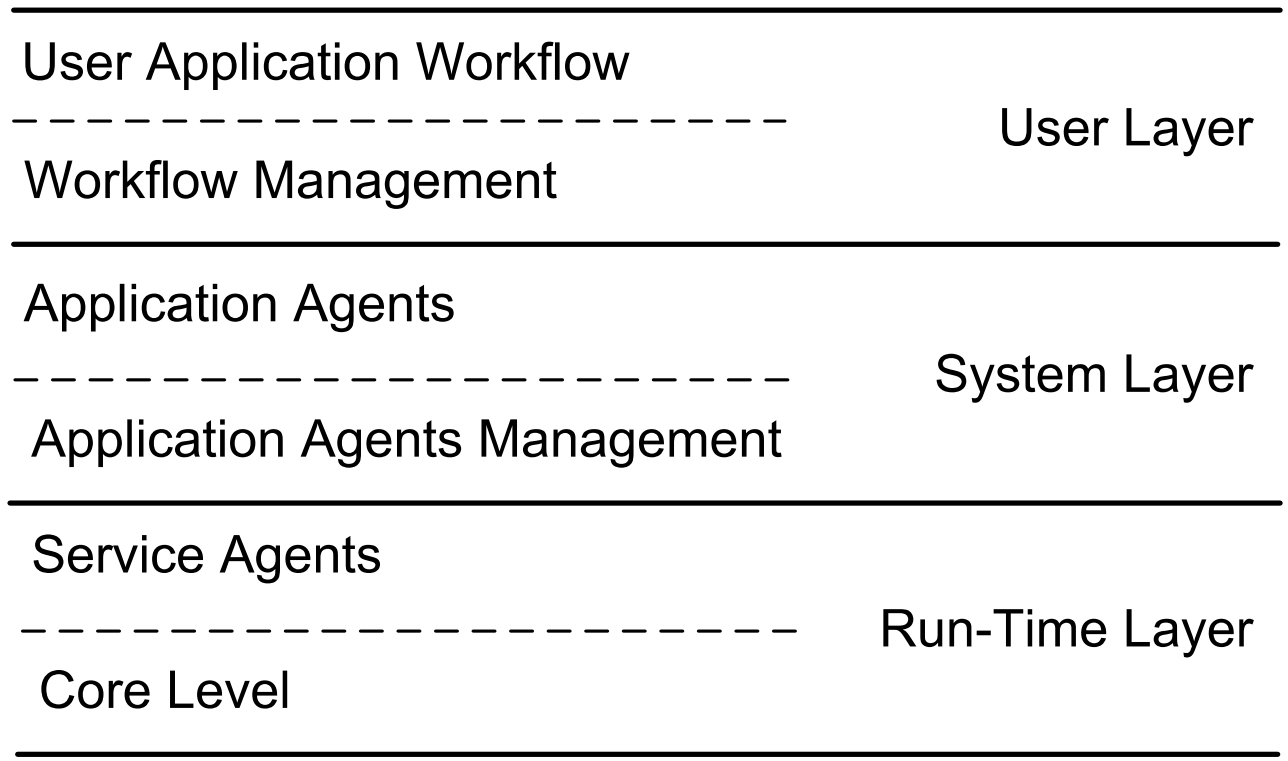


System Simulation

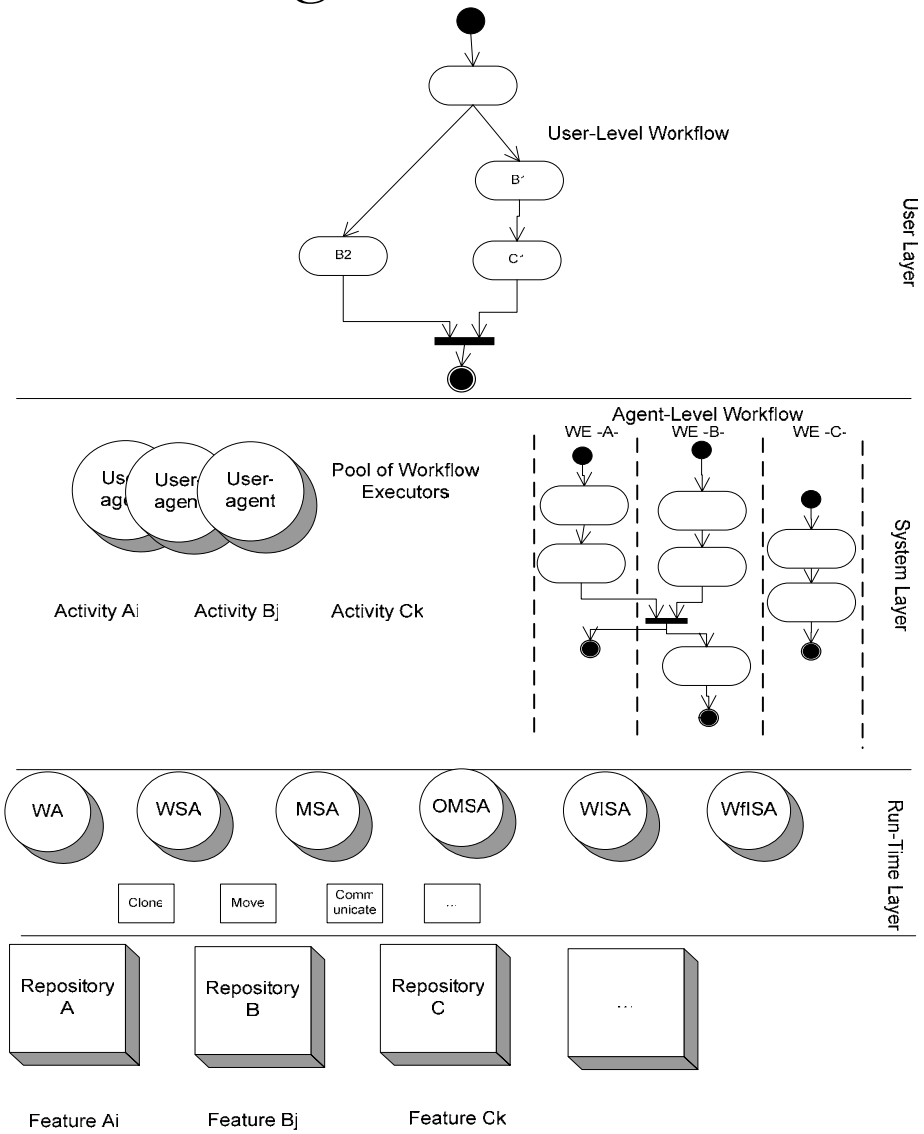
Scenario 2: Bioinformatics and Systems Biology

- **Application Domain**
  - Modelling and Validation of Biological Systems
  - Biological experiments
- **Execution environment**
  - Geographically distributed
  - Dynamic, redondant and huge data repositories
  - Different data schema and semantics
- **Why mobility**
  - to support a flexible workflow engine
  - to reduce network traffic
  - to support user mobility through the lab

Hermes' software architecture

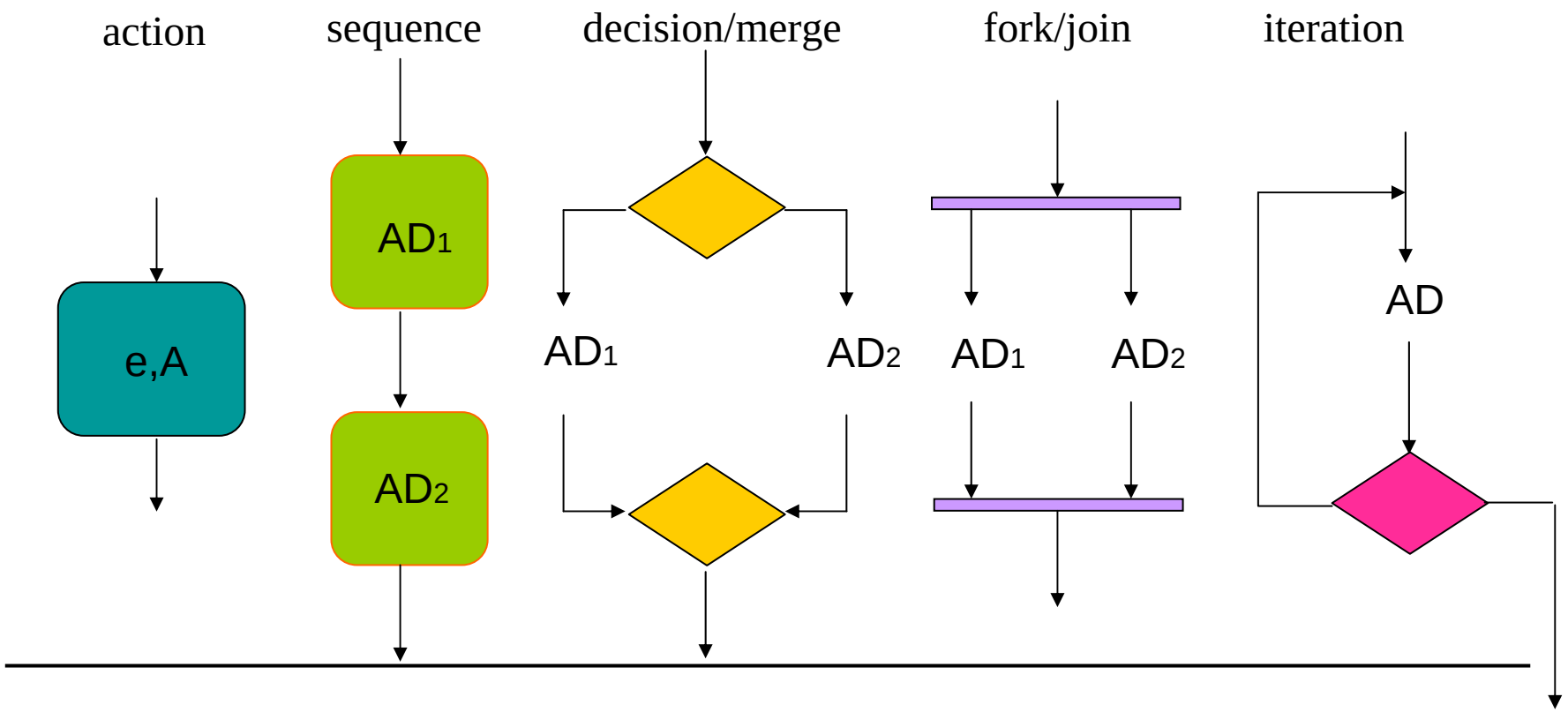


Hermes: The Agent-based Middleware



Workflows

Intended for applications that need to coordinate activities, control flows or object/data flow models ...



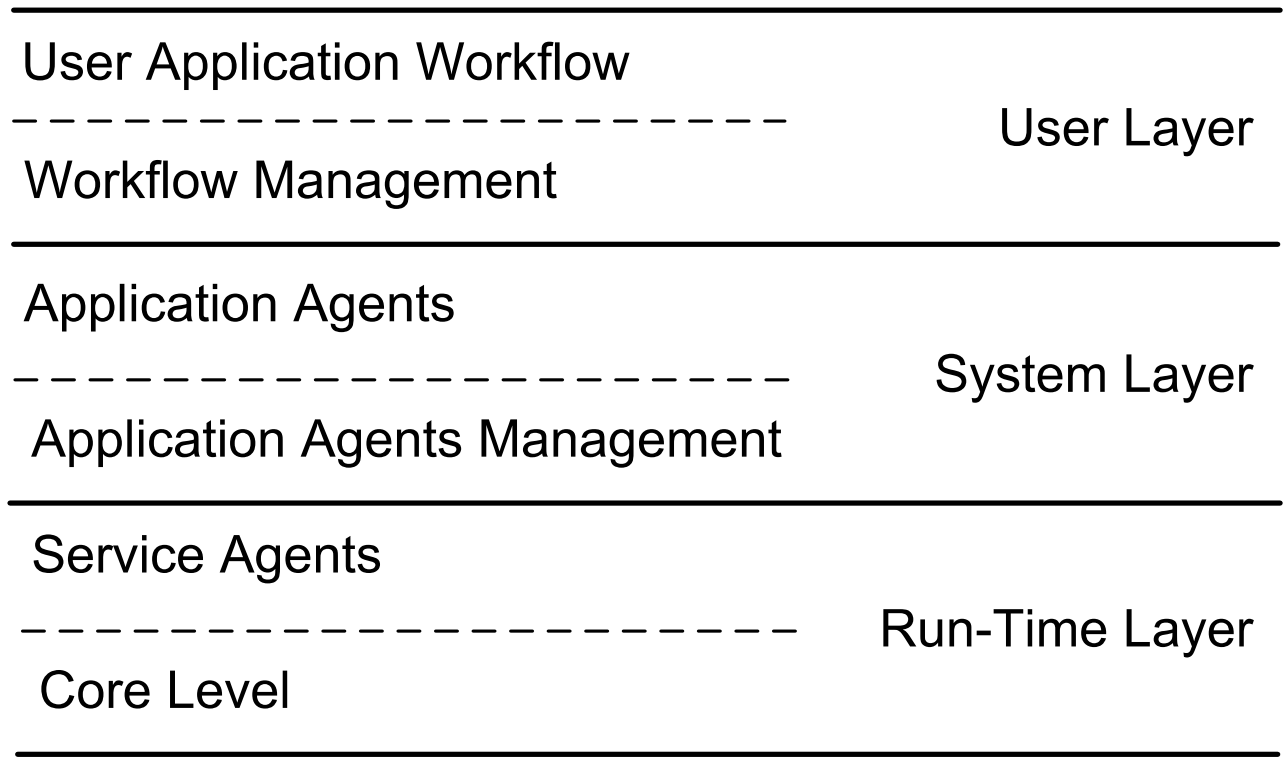
Kernel Primitives

- Clonation
  - Clones code and state of a running agent
  - The cloned agent starts its execution from a given point
  - Fix a communication channel between cloning and cloned
- Mobility
  - Moves an agent from a site to another
  - Movements happens in a proactive way
- Communication
  - based on message passing
  - can be either synchronous or asynchronous
  - between cloning and cloned

Outline

- *Introduction to Hermes and its Mobility*
- Hermes by Example
- Hermes in Action

System's software architecture



Workflows in Bioinformatics and Systems Biology

Definition

A workflow is a partial or total automation of a business process in which a collection of **activities** must be executed by humans or machines according to certain procedural rules.

Goals

- to design and implement a data analysis process (e.g. standardized protocols for oncology research)
- to simulate a high-level biological process (e.g. life cycle of the malaria parassite)

Advantages

- to reproduce the analysis, or simulation
- to reuse intermediate results, or system's components
- to create a transparent analysis environment, simulation environment
- to support a good practice,
- to free the bioscientist from repetitive interactaction with the web
- to verify system's properties at structural, functional and dynamic

Simple Workflow Example

Use context: MIS-Stanford -- verification of proteins mutation experiment by in-silico reproducing

Goal 1: Retrieve abstracts from a molecular biology **literature db** for identifying the best **cell line** for reproducing a human **TP53 mutation** experiment linked to a particular tumour-habits-sex combination

Activities: use Bioinformatics Services available on Internet in order to achieve the desired result

1. Retrieve all ***mutations (IDs)*** observed in the **7th exon** in **men** who are ex-smokers and drinkers
   - by searching p53 mutations database SRS (IST, Genova)

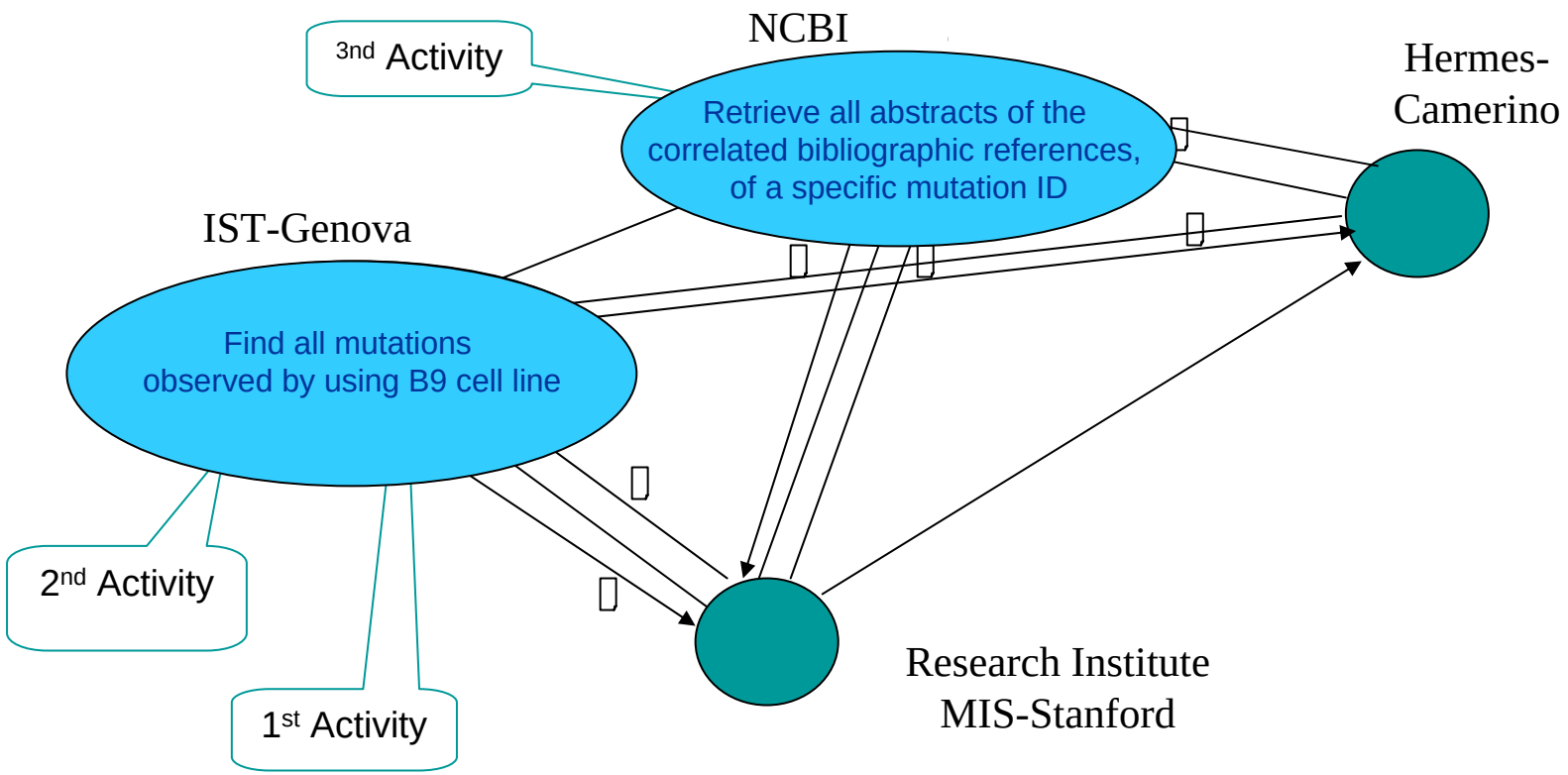
   1st Activity
2. Retrieve all ***mutations (IDs)*** observed by using **B9 cell line** as original resource
   - by searching p53 mutations database SRS (IST , Genova)

   2nd Activity
3. Retrieve all abstracts of the correlated bibliographic references, of a specific ***mutation ID***
   - by searching Medline (NCBI)

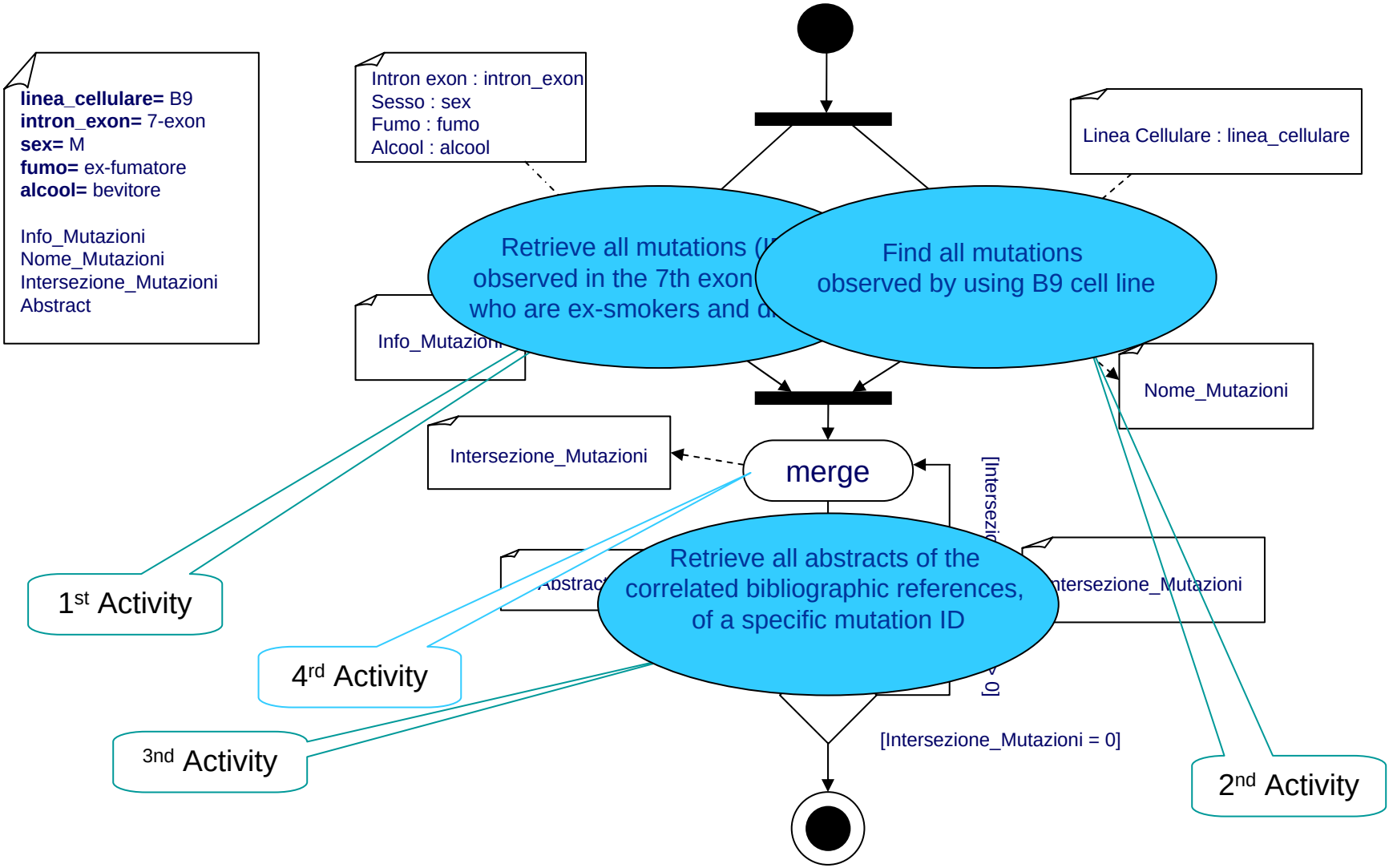
   3rd Activity

Achievement: To integrate on-line Bioinformatics data in a unique result freeing the Bioscientist from the need to personally interact with remote sites.

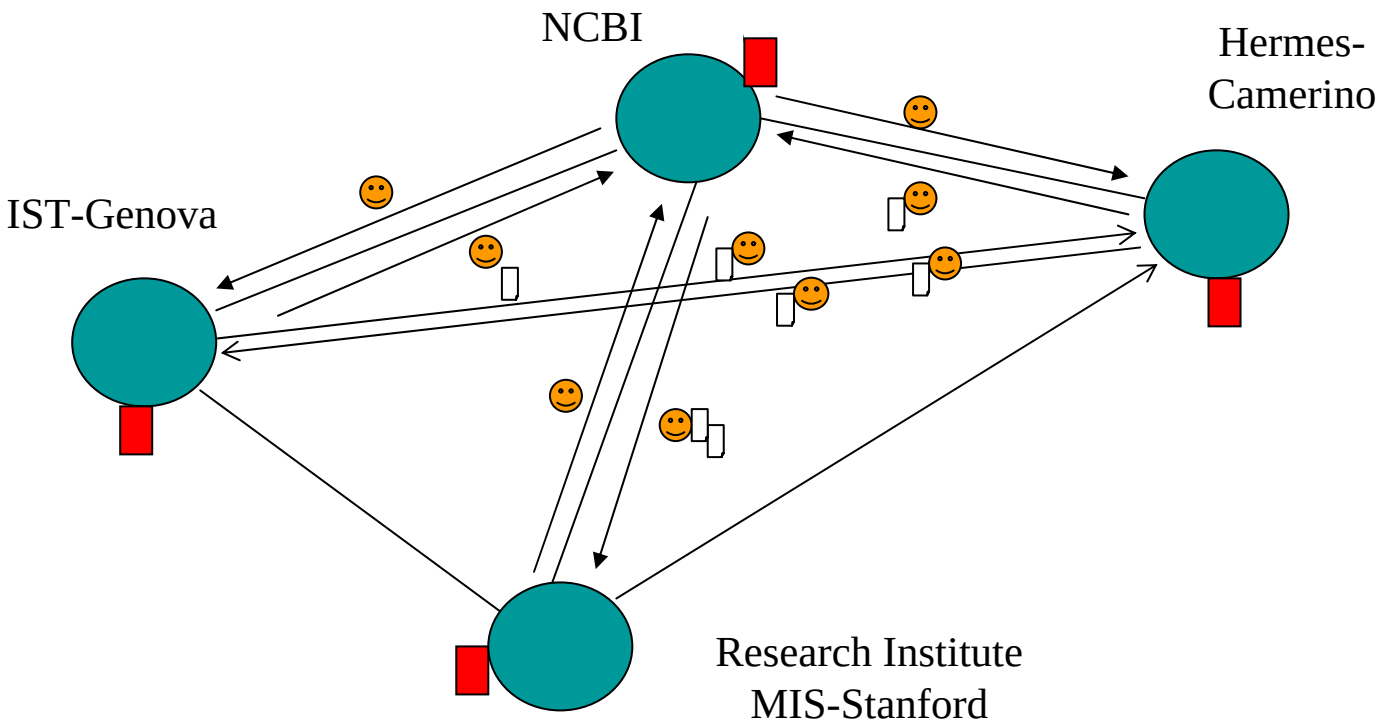
The Geographical Distribution



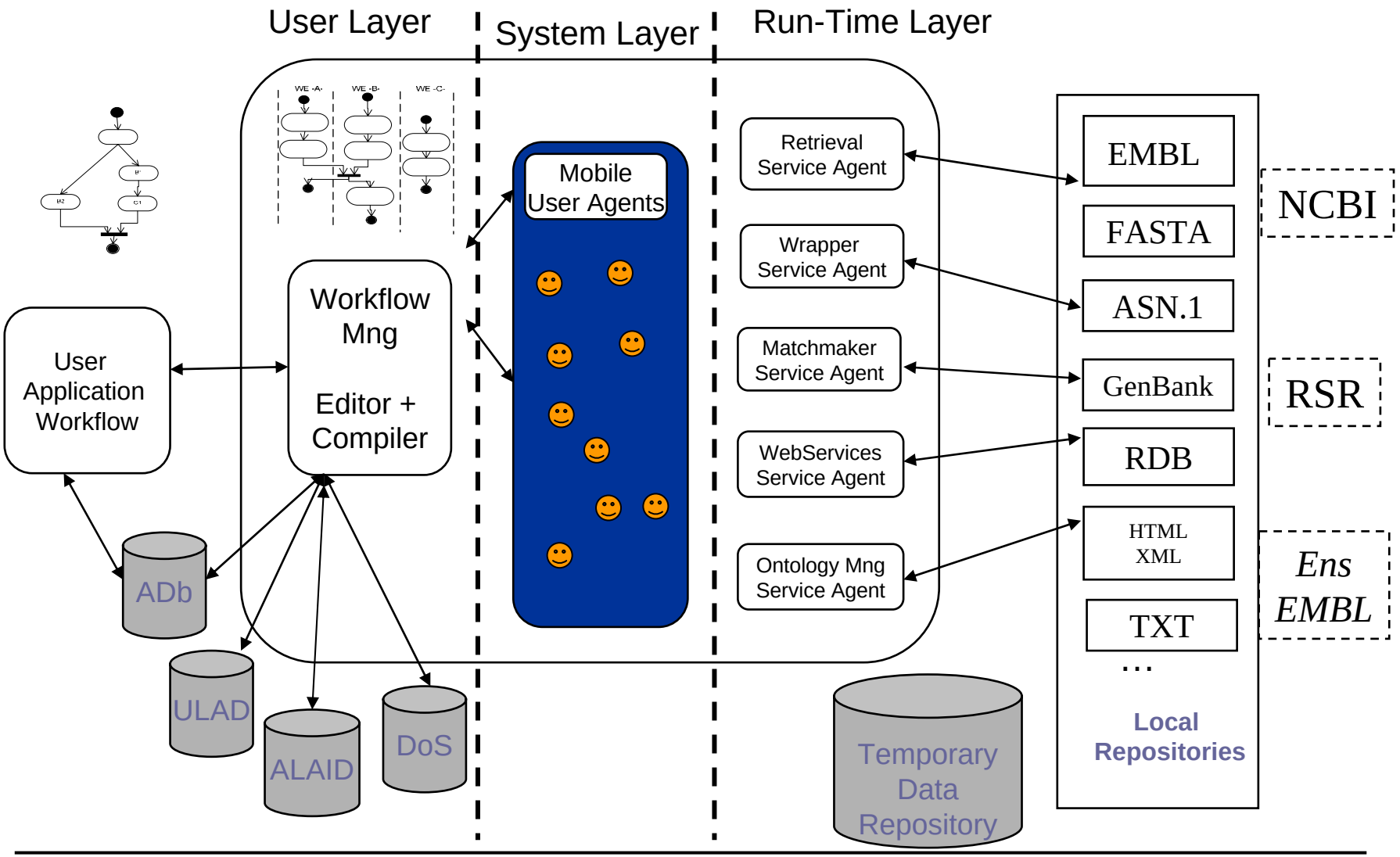
An Example of Workflow at User Level



Global Computation Environment



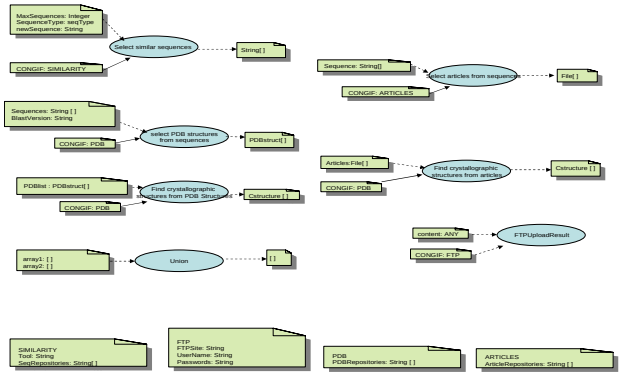
Hermes Middleware Architecture



Activity Database (ADb) to Define a Wf

Activities in Cell Line domain

A1: Find information about the cell line named x
A2: Find all cell lines derived from a specific tumour or pathology
A3: Find all Cell Lines producing a specific protein
A4: Given a specific Cell Line, find all related bibliographic references
A5: Given a specific Cell Line, find all information about produced proteins



Activities in Mutation domain

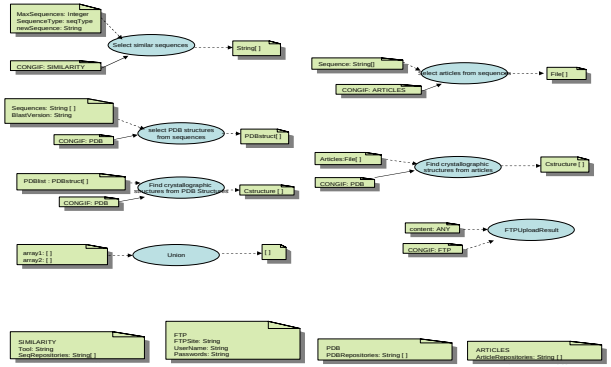
B1: Find all mutations observed in a specific intron/exon in subjects with specific sex and life habits (i.e. smokers/ drinkers)
B2: Find all mutations in subjects affected by a given pathology
B3: Find all subjects affected by a tumoural pathology and with a given protein mutation
B4: Find all mutations observed by using a given cell line
B5: Given a specific mutation, find all abstracts of the correlated bibliographic references

Activities in Bibliographic domain

C1: Select all abstracts of bibliographic references, whose text includes a given term
C2: ……..

Activities in Hermes

H1: Merge in AIXO-WA
H2: ……..



Workflow (Context-Aware) Compiler

The compiler generates a pool of cooperating agents form the workflow specification

Two steps compilation:

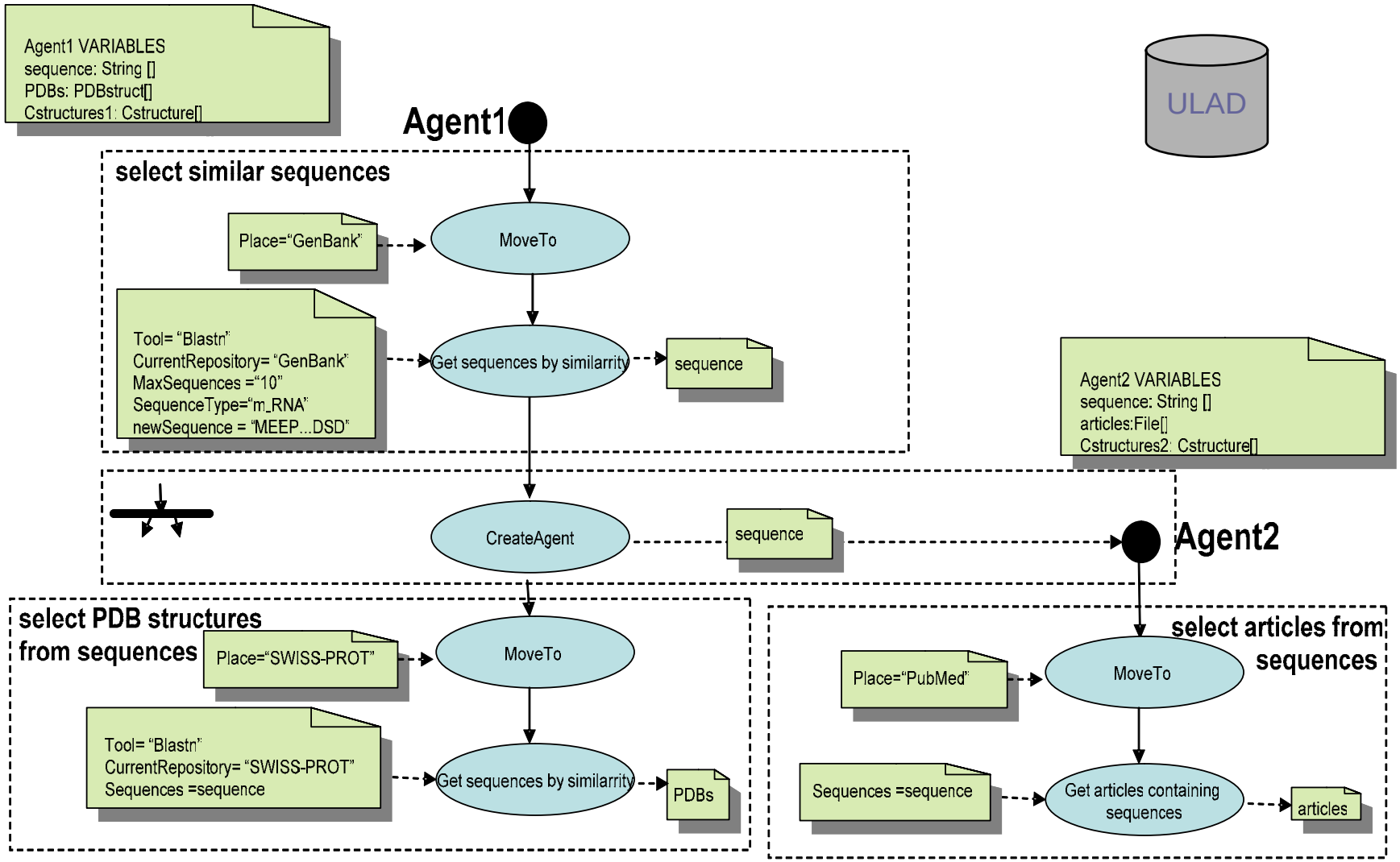
- Step1: from user level Wf to agent level Wf
  - User Level Activity Database (ULAD)
    - contains the mapping from each user activity to an agent level Wf specification

- Step 2: agent synthesis
  - Database of Skeletons (DoS)
    - contains the "empty" implementation of an agent role (skeleton)
  - Agent-Level Activity Implementation Database (ALAID)
    - contains the implementation of each agent level activity

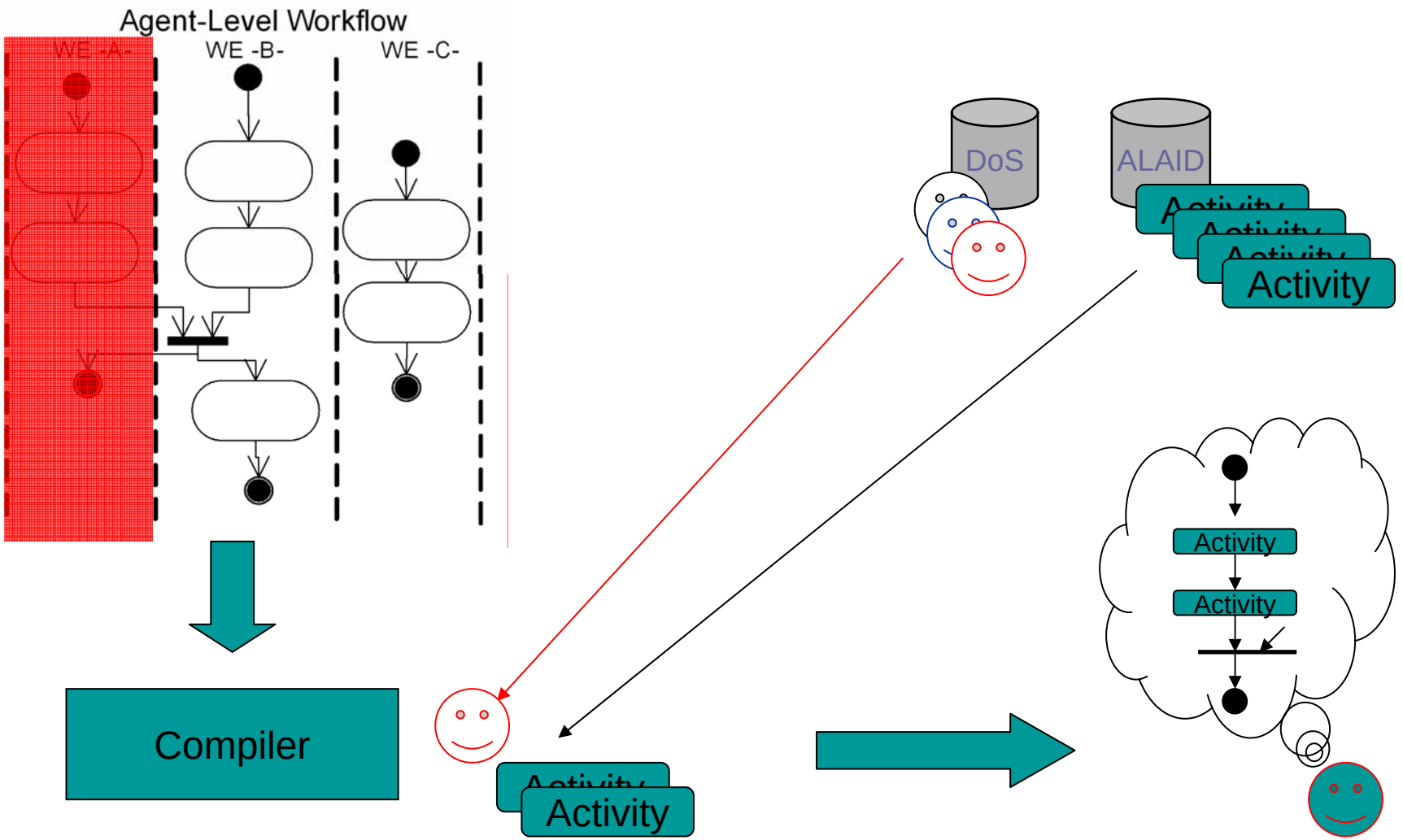
Context-awareness

The compiler takes advantage of information about the state of the global environment during the agent synthesis

Compiler Step 1: from User to Agent Level Workflow



Compiler Step 2: executable application agents



Main Publications

- F. Corradini, E. Merelli. Hermes: An agent-based middleware for Mobile Computing. FMS-Moby, Springer, LNCS 3465, pp. 234-270, 2005

- F. Corradini, L. Mariani, E. Merelli. An agent-based layered middleware as tool integration. Journal of Software Tools Technology Transfer, Vol. 6, pp. 231-244, 2004

- D. Bonura, F. Corradini, E. Merelli, G. Romiti. FARMAS: a MAS for Extended Quality Workflow. Int. Work. on Theory and Practice of Open Computational Systems, TAPOCS'04, IEEE, 2004

- F. Corradini, R. Culmone, M. R. Di Berardini. Code Mobility for Pervasive Computing. Int. Work. on Theory and Practice of Open Computational Systems, TAPOCS'04, IEEE, 2004

- F. Corradini, E. Merelli, M. Vita. A Multi-Agent System for Modelling Carbohydrate Oxidation in Cell. First International Workshop On Modelling Complex Systems, MCS'05, LNCS, 2005

Hermes Software

- Downloadable from

http://hermes.cs.unicam.it

or

http://www.bioagent.net

Thanks to:

Marco Bernardo and **Alessandro Bogliolo**

Co- Directors of FMS-Moby

Ezio Bartocci, **Diego Bonura**, **Rosario Culmone**, **Maria Rita Di Berardini**, **Leonardo Mariani**, **Leonardo Vito**

For their contribution