

sewasie



IST-2001-34825

AL3 - TFG IIAWE  
ROMA, June 30th, 2004



## The SEWASIE project agent architecture

Sonia Bergamaschi - Jonathan Gelati  
Università di Modena e Reggio Emilia



## Outline

- Sewasie overview
- Sewasie system architecture
- Description of SEWASIE agents
- Implementation of the SEWASIE multi-agent system
- Security issues



# The SEWASIE Consortium

- Università degli Studi di Modena e Reggio Emilia (I, coordinator)
- CNA Servizi Modena S.c.a.r.l. (I)
- Università degli Studi di Roma “La Sapienza” (I)
- Rheinisch-Westfaelischen Technischen Hochschule Aachen (D)
- Libera Università di Bolzano (I)
- Thinking Networks AG (D)
- IBM Italia S.p.A. (I)
- FhG/FIT (D)

## Objectives and Main Innovations (1)

The SEWASIE project aims to develop an advanced search engine enabling intelligent access to heterogeneous data sources on the web, via semantic enrichment.

- The SEWASIE system will realise a virtual network, **SEWASIE Virtual Network (SVN)** whose nodes are **SEWASIE Information Nodes (SINode)**.
  - **SINodes** are multi-database mediator-based systems, each including a Virtual Data Store, an Ontology Builder, and a Query Manager
  - The managed **Information Sources** are heterogeneous collections of structured, semi-structured, or unstructured data, e.g. relational databases, XML or HTML documents

## Main Innovations (2)

- Ontologies are multilingual
- The **Brokering Agents** maintain the knowledge related to the SEWASIE Virtual Network and the user profiles.
- **Brokering Agents** classify SINodes, are responsible for handling the acquisition of a new SINode and for consequently updating of the SEWASIE Virtual Network.
- In query solving phase, starting from a specified SINode, the **Query Agent** accesses other SINodes and collects partial answers.
- To select SINodes useful to solve a query, a **Query Agent** interacts with a Brokering Agent.

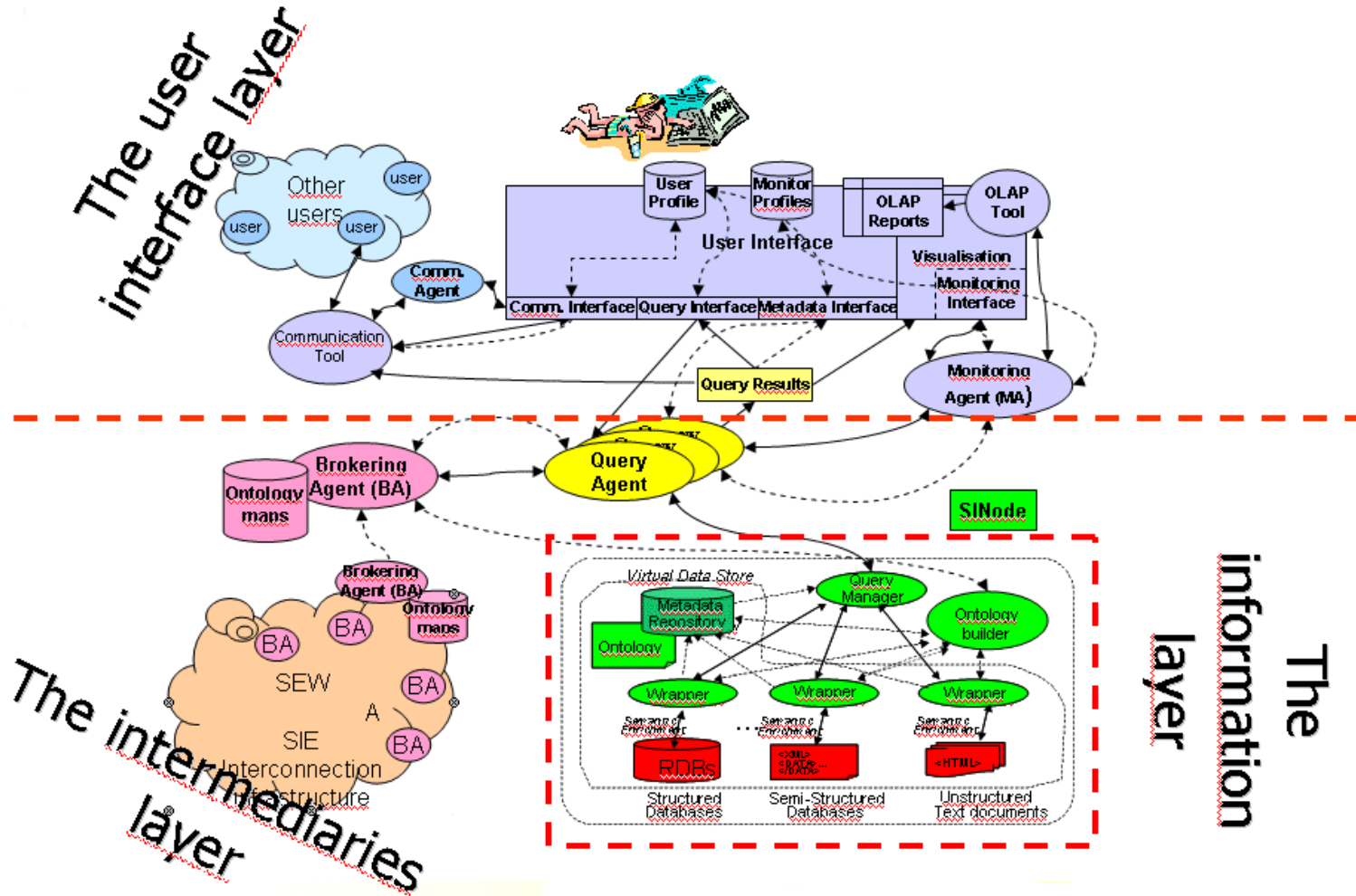


## Main Innovations (3)

- An integrated, semantically enhanced, user interface allowing
  - Support for query formulation
  - Management of query results and integration with other tools
    - OLAP engines
    - Negotiation tools
  - Continuous monitoring of query results



# The SEWASIE architecture





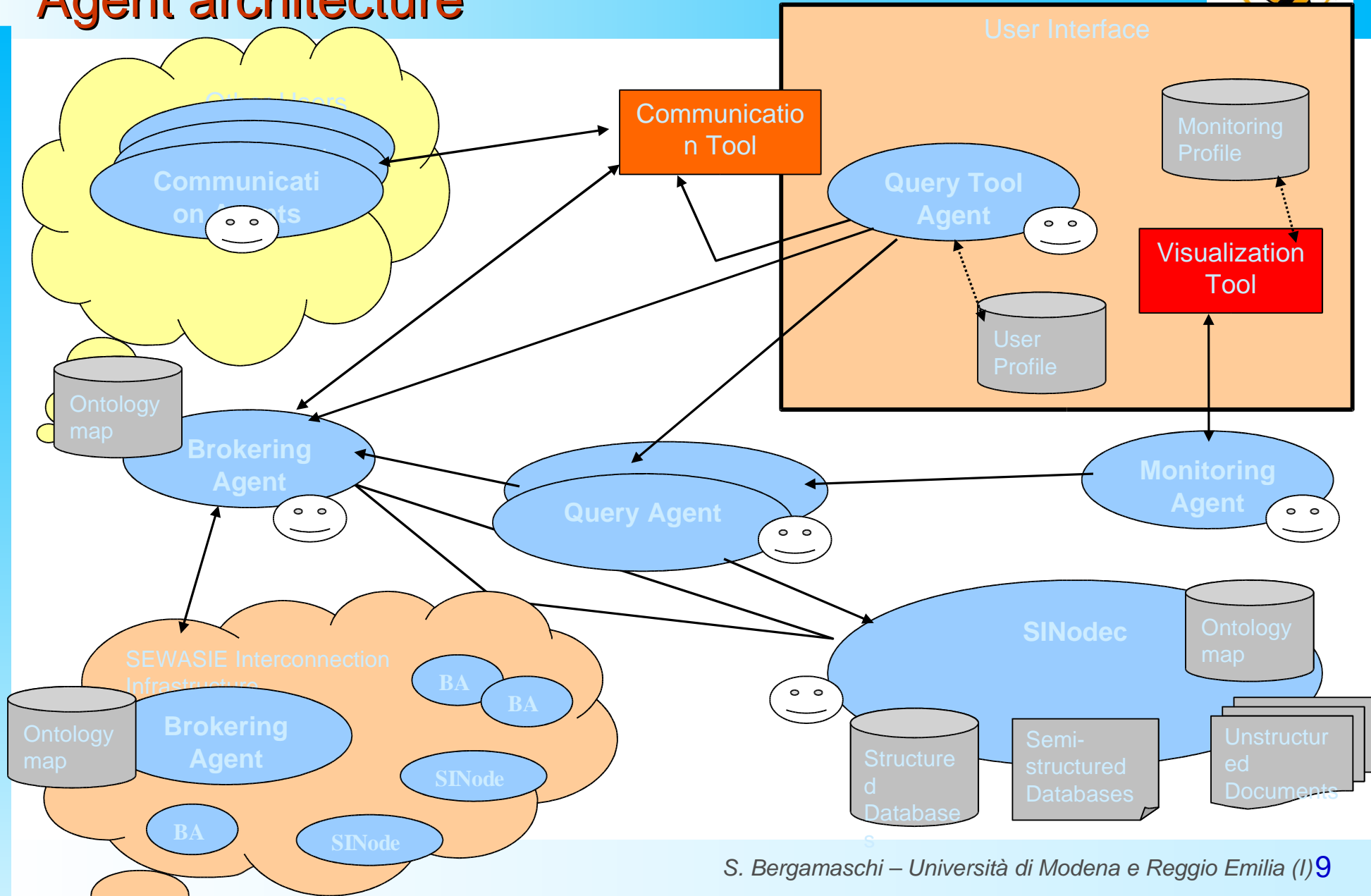
# SEWASIE agent infrastructure

1. SEWASIE agent architecture
2. System Deployment in this prototype





# Agent architecture





## Agent architecture

**Query Agents (QAs)** carry the user query from the user interface to the SINodes.

- They solve the query by interacting with BAs
- BAs return both SINodes and other BAs references relevant to the query
- Before contacting a SINode, the QA must translate the query with respect to its GVV



## Agent architecture

**Brokering Agents (BAs)** represent the higher level of knowledge integration.

- BAs Maintain a direct control of several SINodes, and are aware of other BAs.
- They are the entry points to the system, routing the queries to relevant information nodes.



## Agent architecture

The **User Interface** includes a Query Tool Agent (QTA) that is responsible to carry out communications with other agents.

- QTAs also searches the agent network to find a suitable initial BA.



## Agent architecture

The **Visualization Tool** includes **monitoring Agent (MA)** that are created for each topic of interest.

- MAs periodically create QAs to query the SEWASIE network about the assigned topic.



## Agent architecture

### Several **Communication Agents (CAs)**

participate in the **Communication Tool**: Initiation agents, Filtering and Ranking agents, Negotiation agents, and Resource Management agents.

- Initiation agents start the negotiation process; negotiation agents represent each participant.
- The other agents provide support to negotiation agents.



# SEWASIE agent infrastructure

1. SEWASIE agent architecture
2. System Deployment in this prototype



# System Deployment

JADE (Java Agent Development framework) has been chosen as the underlying agent platform.

- It is an open source platform for peer-to-peer agent based application, initially developed by TILab, Motorola and the University of Parma.
- Features:
  - FIPA compliant
  - Java based
  - free available under LGPL
  - well documented and supported





# System Deployment

