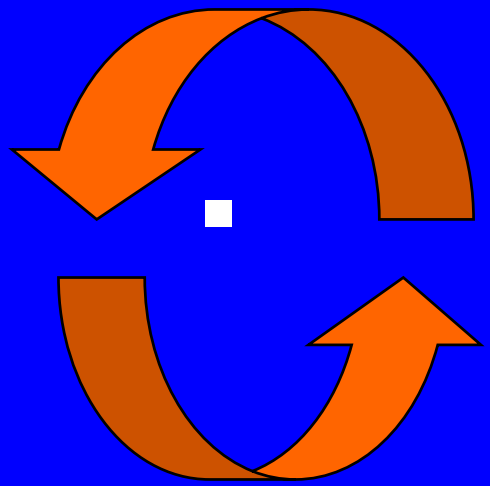
Web services and Software agents



F. Dignum
Utrecht University

Universiteit Utrecht

Contents

- What are Web Services?
- Standards
- Problems with Web Services
- Software Agents
- Conclusions

Universiteit Utrecht

Today's Web

Facts and figures

- 3 billion websites
- 450 m Internet users (33% US)
- Online B2B market volume 2000: $282 billion

Purpose

- Web designed for application to human interactions
  - Served very well its purpose:
  - Information sharing: a distributed content library
  - B2C e-commerce
  - Non-automated B2B interactions

Universiteit Utrecht

Web Services: Killer Application?

"Web services are expected to revolutionize our life in much the same way as the Internet has during the past decade or so." (Gartner)

"Just as the Web revolutionized how users talk to applications, XML transforms how applications talk to each other." (Bill Gates)

"By 2004, 40% of financial services transactions and 35% of online government services will be web service-based." (Gartner)

"Web Services will be bigger than Java or XML" (Rod Smith, VP of Emerging Technology, IBM)

© Intelligent Agents Group, Carnegie Mellon University 2002

Development on the Web

Old Web:

"The eye-ball Web"

The architecture of the Web is geared towards delivering information visually

New Web:

"The transactional Web"

The architecture of the Web is geared towards intelligently exchanging information between applications

Universiteit Utrecht

Definitions

Many Definitions Exist…

It is software designed to be used by other software via Internet protocols and formats. (Forrester)

Web Services are self-describing components that can discover and engage other web services or applications to complete complex tasks over the Internet. (Sun Microsystems, Inc.)

Web Services are loosely coupled software components delivered over the Internet via standards-based technologies like XML, and SOAP. (Gartner)

"Self-describing, self-contained, modular unit of application logic that provides some business functionality to other applications through an Internet connection…" (UDDI.org)

Web Services are Internet-based, modular applications that perform a specific business task and conform to a particular technical format. (IBM)

A web service is application logic that is programmatically available, exposed using the Internet. (Microsoft)

Universiteit Utrecht

Characteristics

- A Web Service is accessible over the Web
- Web Services communicate using platform independent and language neutral Web protocols
- A Web Service provides specific functionality that can be used by other programs
- A Web Service is registered and can be located through a Web Service Registry

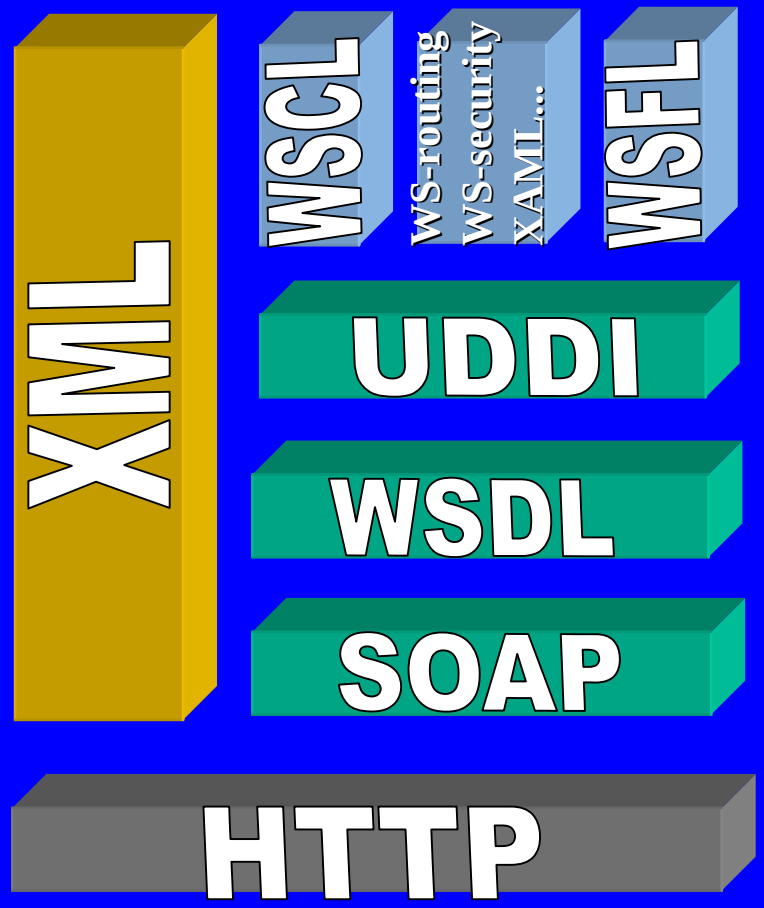
Universiteit Utrecht

Examples

- Amazon Web Services: search products and maintain shopping cart
  - http://soap.amazon.com/schemas2/AmazonWebServices.wsdl
- Convert currencies
  - http://www.xmethods.net/sd/CurrencyExchangeService.wsdl
- Find phone numbers for Swedish persons
  - http://www.marotz.se/scripts/searchperson.exe/wsdl/ISearchSwedishPerson

Universiteit Utrecht

Web services standards



- Control flow descriptions
- Security and management

- Mechanism for registering and looking up Web Services
- Programmatic description of Web Service interface
- Web Services communication protocol

Universiteit Utrecht

SOAP
Simple Object Access Protocol

- XML based web services communication protocol
  - Provides message support for many Web Services standards such as WSDL, UDDI, and Microsoft's .NET architecture
  - Uses GET/POST across http, thus providing a platform & language independent means of communicating
- SOAP documents contain:
  - Header
    - optional information about the transaction
  - Body
    - contains payload (e.g. a request or response)
    - may instead contain error/fault information if requests fail

Universiteit Utrecht

WSDL
Web Service Description Language

- Structured mechanism to describe in XML:
  - Abstract operations that a Web Service can perform
  - Format of messages it can process
  - Protocols it can support
  - Physical bindings to:
    - communication languages, e.g. SOAP or HTTP messages
    - Location of services, i.e. URI and port numbers

Universiteit Utrecht

WSDL components

- **Types:** containers for data type definitions
- **Message:** abstract definition of the data being communicated
- **Operation:** abstract message exchange protocol
- **Port Type:** abstract set of operations
- **Binding:** concrete protocol and data format for a port type
- **Port:** single, physical endpoint
- **Service:** collection of related endpoints

Universiteit Utrecht

WSDL Components: Types

```
<xsd:complexType name="KeywordRequest">
<xsd:all>
    <xsd:element name="keyword"
    type="xsd:string"/>
    <xsd:element name="page"
    type="xsd:string" />
    <xsd:element name="mode"
    type="xsd:string" />
    <xsd:element name="sort"
    type="xsd:string" minOccurs="0" />
      ...
</xsd:all>
</xsd:complexType>

< xsd:complexType name="ProductInfo">
<xsd:all>
    <xsd:element name="TotalResults"
    type="xsd:string" minOccurs="0" />
     ...
    <xsd:element name="Details"
    type="typens:DetailsArray" minOccurs="0"
    />
</xsd:all>
</xsd:complexType>
```

```
<xsd:complexType name="Details">
<xsd:all>
    <xsd:element name="ProductName"
    type="xsd:string" minOccurs="0" />
    <xsd:element name="Authors"
    type="typens:AuthorArray" minOccurs="0"
    />
    <xsd:element name="ListPrice"
    type="xsd:string" minOccurs="0" />
    <xsd:element name="OurPrice"
    type="xsd:string" minOccurs="0" />
    <xsd:element name="UsedPrice"
    type="xsd:string" minOccurs="0" />
    <xsd:element name="RefurbishedPrice"
    type="xsd:string" minOccurs="0" />
    <xsd:element name="CollectiblePrice"
    type="xsd:string" minOccurs="0" />
    <xsd:element name="ThirdPartyNewPrice"
    type="xsd:string" minOccurs="0" />
    <xsd:element name="Isbn"
    type="xsd:string" minOccurs="0" />
    <xsd:element name="Availability"
    type="xsd:string" minOccurs="0" />
</xsd:all>
</xsd:complexType>
```

WSDL components: Messages

- Protocol independent message contained within the requester's query and the services response.
- Typical transaction consists of two messages, though several messages may be defined for different transactions

```
<message name="KeywordSearchRequest">
  <part name="KeywordSearchRequest" type="typens:KeywordRequest" />
  </message>
<message name="KeywordSearchResponse">
  <part name="return" type="typens:ProductInfo" />
  </message>
```

Universiteit Utrecht

WSDL components: Operation and port type

- PortTypes contains a set of abstract operations
- There are four transaction primitives defined by WSDL operations:
  - **One-way**. The endpoint receives a message.
  - **Request-response**. The endpoint receives a message, and sends a correlated message.
  - **Solicit-response**. The endpoint sends a message, and receives a correlated message.
  - **Notification**. The endpoint sends a message.

```
<portType name="AmazonSearchPort">
<operation name="KeywordSearchRequest">
    <input message="typens:KeywordSearchRequest" />
    <output message="typens:KeywordSearchResponse" />
    </operation>
  ....
  </portType>
```

Utrecht

WSDL Components: Bindings

```
<binding name="AmazonSearchBinding" type="typens:AmazonSearchPort">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="KeywordSearchRequest">
  <soap:operation soapAction="http://soap.amazon.com" />
          <input>
                  <soap:body use="encoded“
                  encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                  namespace="http://soap.amazon.com" />
          </input>
          <output>
                  <soap:body use="encoded"
                  encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                  namespace="http://soap.amazon.com" />
          </output>
  </operation>
  …
</binding>
```

WSDL Components: Port and Service

```
<service name="AmazonSearchService">
        <port name="AmazonSearchPort"
                binding="typens:AmazonSearchBinding">
        <soap:address location="http://soap.amazon.com/onca/soap2" />
 </port>
 </service>
```

Universiteit Utrecht

UDDI: Universal Discovery, Description & Integration

- Public directory for registering and looking up services
- Uses Type Model or tModel documents
- Keyword searches based on standard taxonomies

Universiteit Utrecht

UDDI: Search

- Four methods for searching available through API:
  - *Find_business*
    - Locate information about one or more businesses.
  - *Find_binding*
    - Locate specific bindings within a registered business.
  - *Find_service*
    - Locate specific service within a registered Buisness Entity.
  - *Find_tModel*
    - Locate one or more tModel Information structures.
- Arguments to searches are keyword based
  - Uses keywords to guide search:
    - find business named IB*
  - Use tModels to find services with a feature:
    - find all services with WSDL specification

Universiteit Utrecht

Semantic interoperability?

- Locate Services
  - Different terms used for advertisements and requests
- Invoking a Service
  - Constructing valid messages based on the published signature/interface of a service
- Understanding a Service
  - Interpreting the results of invoking a service
- Composing Services
  - Constructing plans to achieve meta-goals based on available Services/Agents

Universiteit Utrecht

Locate Services

- The Web Services of Amazon are available under: “AmazonSearchService”

- What happens if I search on “BookSearchService”?

Universiteit Utrecht

Invoking a Service

- If I send the Amazon “KeywordSearchRequest” message the third parameter of the message is the “mode” of the request.

```
<xsd:complexType name="KeywordRequest">
<xsd:all>
      <xsd:element name="keyword" type="xsd:string"/>
      <xsd:element name="page" type="xsd:string" />
      <xsd:element name="mode" type="xsd:string" />
      <xsd:element name="sort" type="xsd:string" minOccurs="0" />
        ...
     </xsd:all>
     </xsd:complexType>
```

- What is the “mode” of a request?

Universiteit Utrecht

Understanding a Service

- If I receive the Amazon "KeywordSearchResponse" message one of the parameters indicates the availability of a product.

```
<xsd:complexType name="Details">
<xsd:all>
 …
 <xsd:element name="OurPrice" type="xsd:string" minOccurs="0" />
 <xsd:element name="Availability" type="xsd:string" minOccurs="0" />
</xsd:all>
</xsd:complexType>
```

- How is availability expressed?
  - Y/N, Yes/No, 1,2,3…
- What is the currency of "OurPrice"?
  - Dollars, Euros,…

Universiteit Utrecht

Combining Services

- One could combine the search service of Amazon with a currency conversion service.
- The new service would give the prices of the products in a pre-specified currency.
- How can we do this “on the fly”?

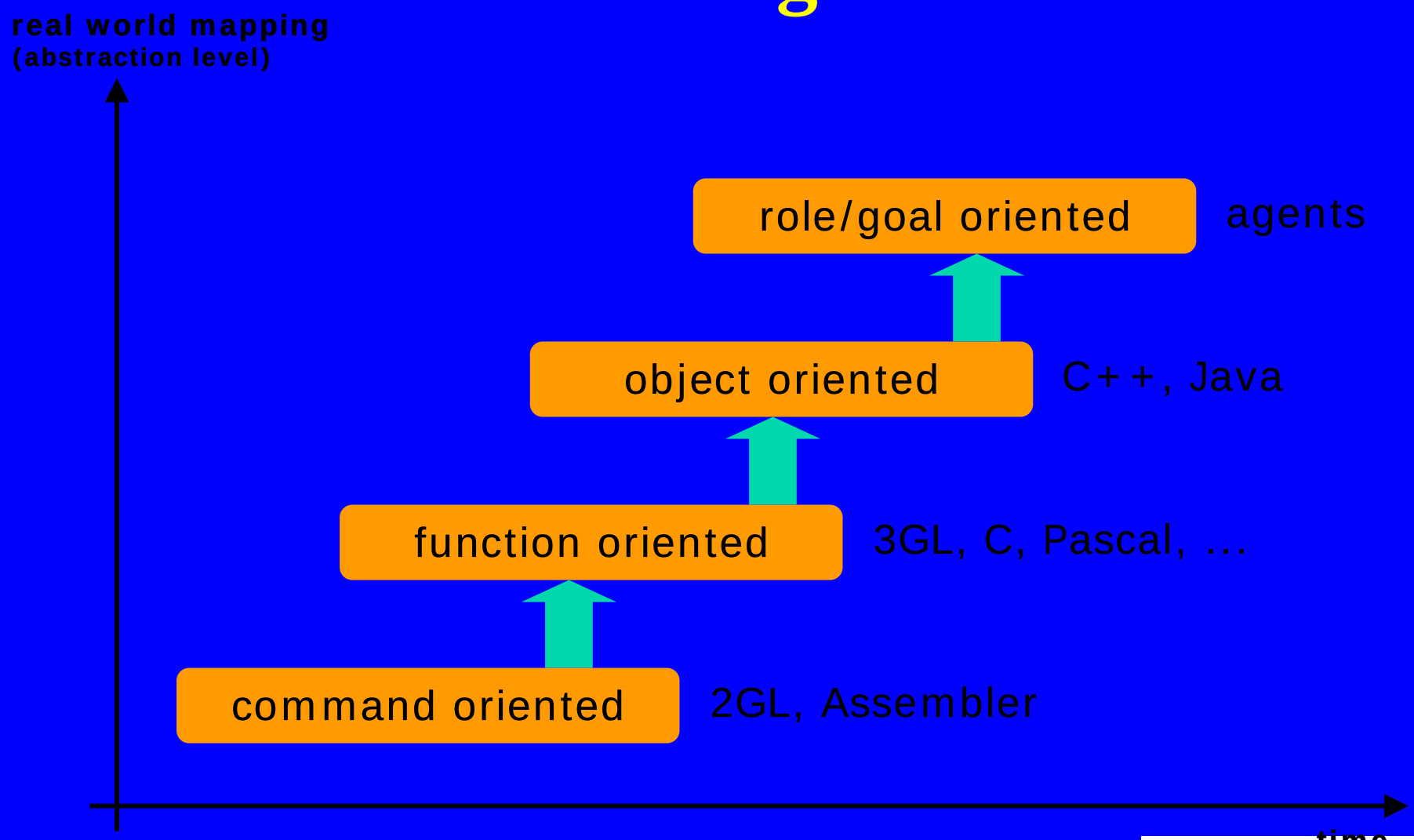
Universiteit Utrecht

Software agents?

- Development metaphor
  - Software Engineering
  - Methodology
- Technology
  - agent theory
  - agent architectures
  - agent languages

Universiteit Utrecht

Paradigm Shifts



Universiteit Utrecht

Definition of agents (weak notion)

- Autonomous
- Pro-active
- Reactive
- Social ability

Universiteit Utrecht

Definition of agents (strong notion)

- Belief
- Desire
- Intentions
- Goals
- Knowledge
- Obligations

Universiteit Utrecht

Other possible attributes

- Rationality
- Veracity
- Mobility
- Learning capacity
- Cooperativeness

Universiteit Utrecht

Theory of agents

- Represent mental attitudes
  - knowledge, belief, goals, etc.
- Reason about mental attitudes
- Plan actions
- "Observe" changes
- Update mental attitudes

Universiteit Utrecht

From Theory to Architecture

- How to use the theory in architecture of agents:
  - Only for representation of attitudes
  - As formal specification of agent implementation
  - Also use the logical inferencing of logic in deliberative agent architecture

Universiteit Utrecht

Agent architectures

- Deliberative agents
  - BDI agents (mostly theory)
  - planning agents (IRMA)
- Reactive agents
  - Brooks' subsumption architecture
- Hybrid agents
  - Interrap

Universiteit Utrecht

Building Agents

1. Agent Oriented Programming (e.g. 3APL)
   - Close to agent theory, but far from industrial use
2. Based on Java components (e.g. Jade)
   - More robust, but build intelligence yourself in Java
3. Based on robust infrastructure (e.g. Tryllian ADK)
   - Industry standard systems (robust, efficient, scalable), but no intelligence

Universiteit Utrecht

From Agents to Multi-Agent Systems

- Agent communication
- Agent societies
  - realise there are other agents
  - use other agents for your actions
- Collective plans, goals, etc.
- MAS vs. Autonomous agents

Universiteit Utrecht

Some General Application Areas:

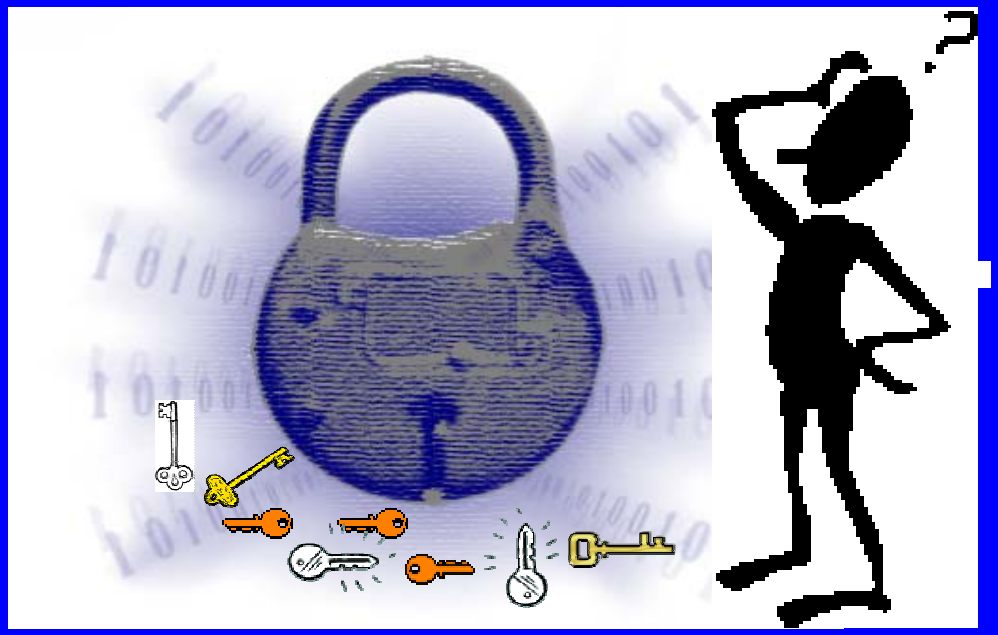
1. Industrial applications
   - manufacturing
   - process control
   - telecommunications
   - transportation systems
2. Electronic Commerce
   - electronic markets/auctions
   - buying agents (e.g. Jango, shopbot, etc.)
3. Business Process Management
4. Information Management
   - information gathering
   - information filtering

Universiteit Utrecht

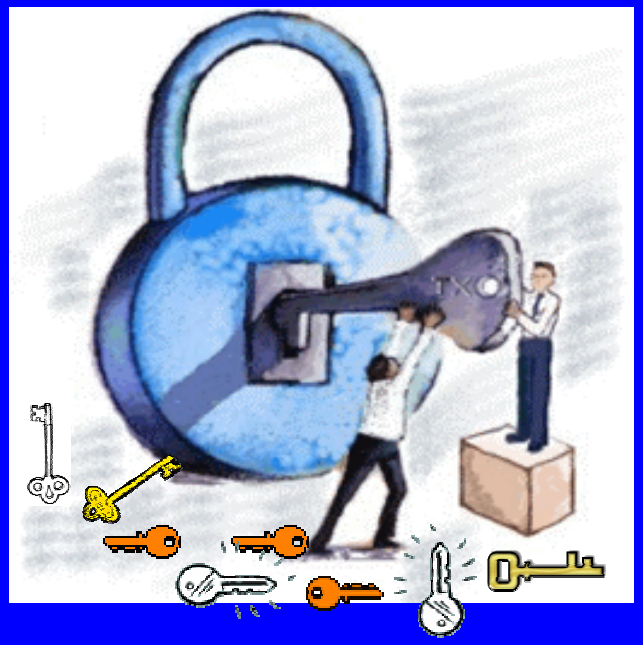
When are agents most useful?

- For large scale, complex systems
- For distributed systems
- For heterogeneous systems
- In open environment
- Adaptable systems

Agents and Web Services?



without agents



with agents

Universiteit Utrecht

Agents and Web Services?

- Web Service as capabilities of agents.
  - Agent analyses requests and can determine how to fit it with a Web Service.
  - Agent can provide explanation on results of Service.
  - Agent can answer why questions.
  - Agent can refer to other agents/Services.
- Agents can contain and reason about the semantics of the Web Services
- Agents can mediate and compose Web Services

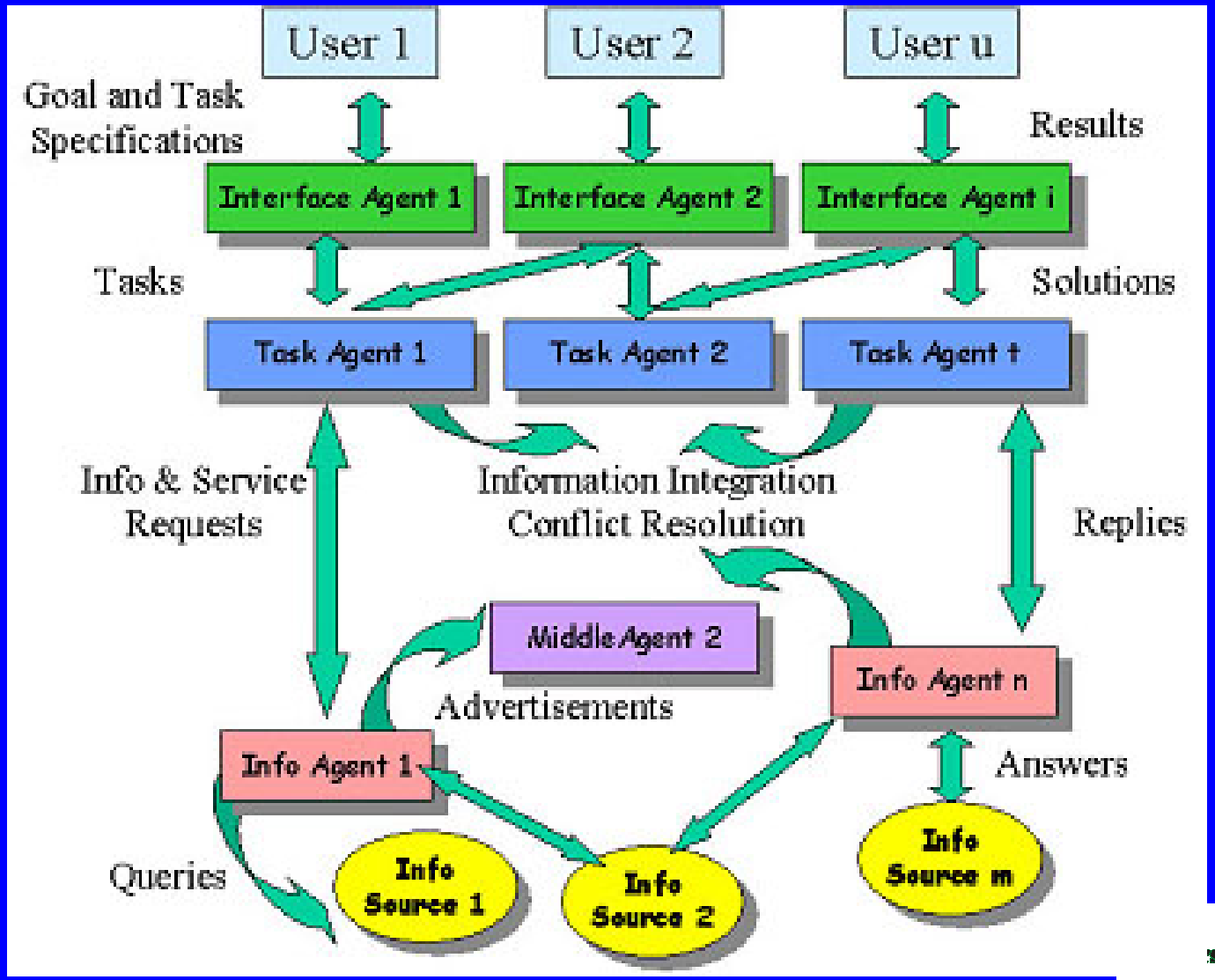
Universiteit Utrecht

Agents and standards

- Semantic descriptions of ontologies and services
  - DAML: DARPA Agent Mark up Language
  - DAML-OIL: extend DAML with logical properties
  - OWL: Ontology Web Language
  - DAML-S: DAML Service description language
- Communication standards
  - FIPA ACL (based on speech act theory)
- Mediation facilities and architectures
  - FIPA/Jade AMS; Agent System Manager
  - Mediation architectures, e.g. Retsina

Universiteit Utrecht

Agent mediation architecture



Software agents and Web Services

- Web Services profit from:
  - Dynamic creation of agents
  - Semantic level of communication
  - Mediation and reasoning capabilities
  - Flexibility and robustness
- Agents profit from:
  - Standard description techniques for procedures
  - Ontology descriptions
  - Connection to web

Universiteit Utrecht

Conclusions

Web Services

Agents

Universiteit Utrecht