Semantic Web enabled Web Services

Dieter Fensel & Christoph Bussler

Vrije University Amsterdam
Oracle, U.S.A.

Contents

- The Vision
- State of the Art
- Requirements for Making Web Services a Working Technology
- The Web Service Modeling Framework (WSMF)
- Semantic Web enabled Web Services
- Conclusions

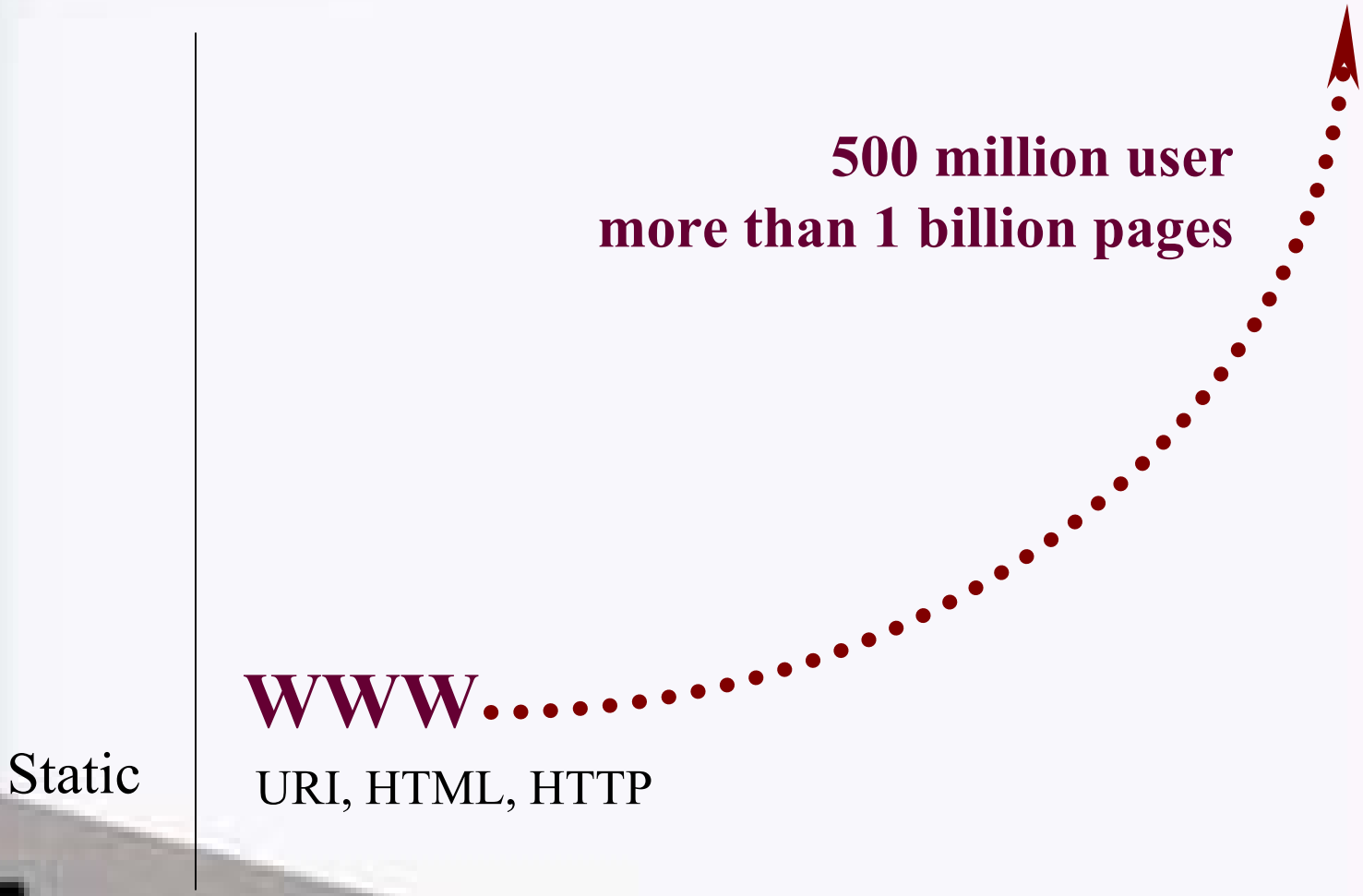
1. The Vision

- Web Services will transform the web from a collection of information into a distributed device of computation.
- In order to employ their full potential, appropriate description means for web services need to be developed.
- For this purpose we developed a full-fledged *Web Service Modeling Framework (WSMF)* that provides the appropriate conceptual model for developing and describing web services and their composition.

The Vision

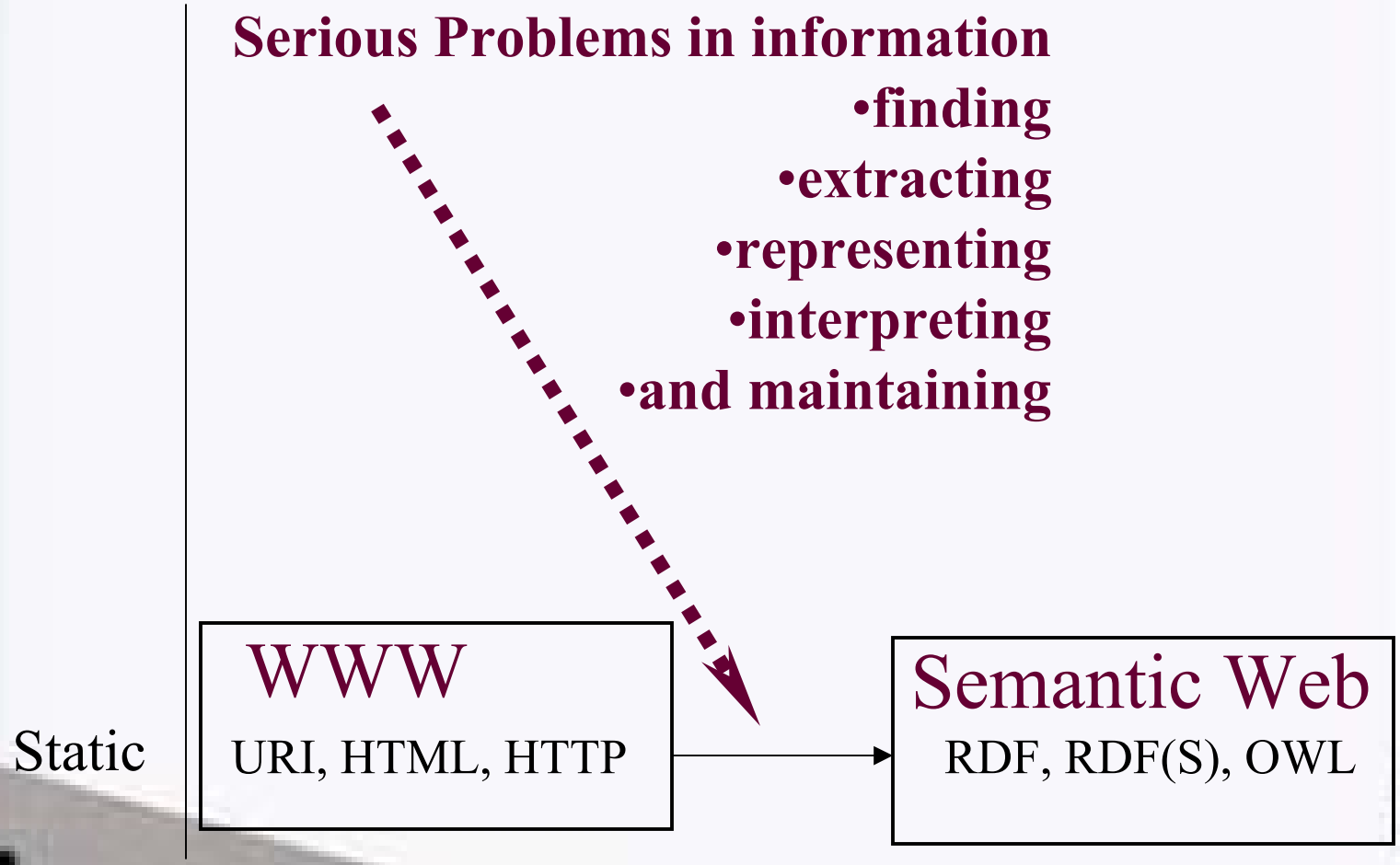
- The philosophy of WSMF is based on the following principle:
    - *maximal de-coupling* complemented by
    - *scalable mediation service*.
- This is a pre-requisite for applying *semantic web* technology for web service discovery, configuration, comparison, and combination.

The Vision

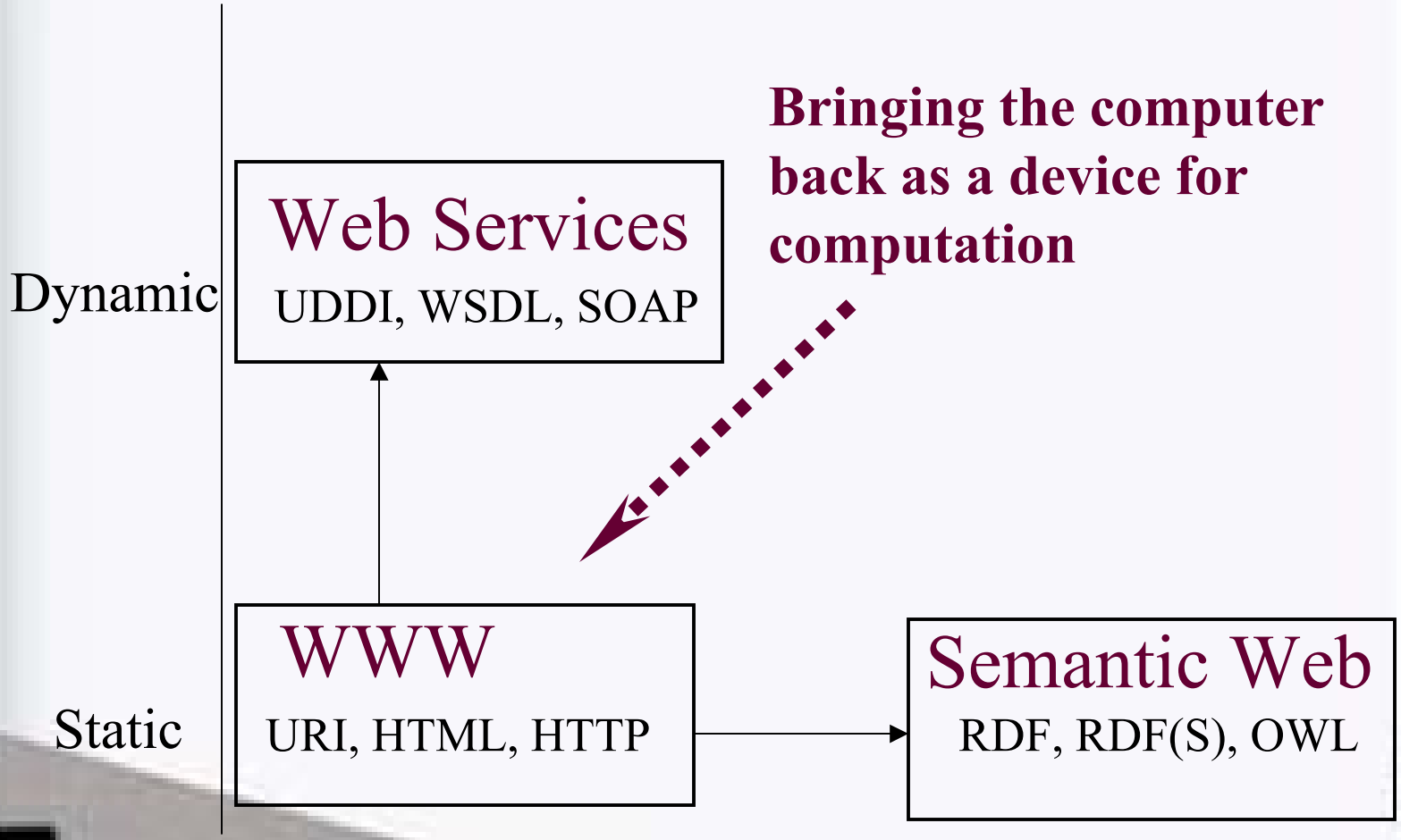


Semantic Web enabled
Web Services

The Vision

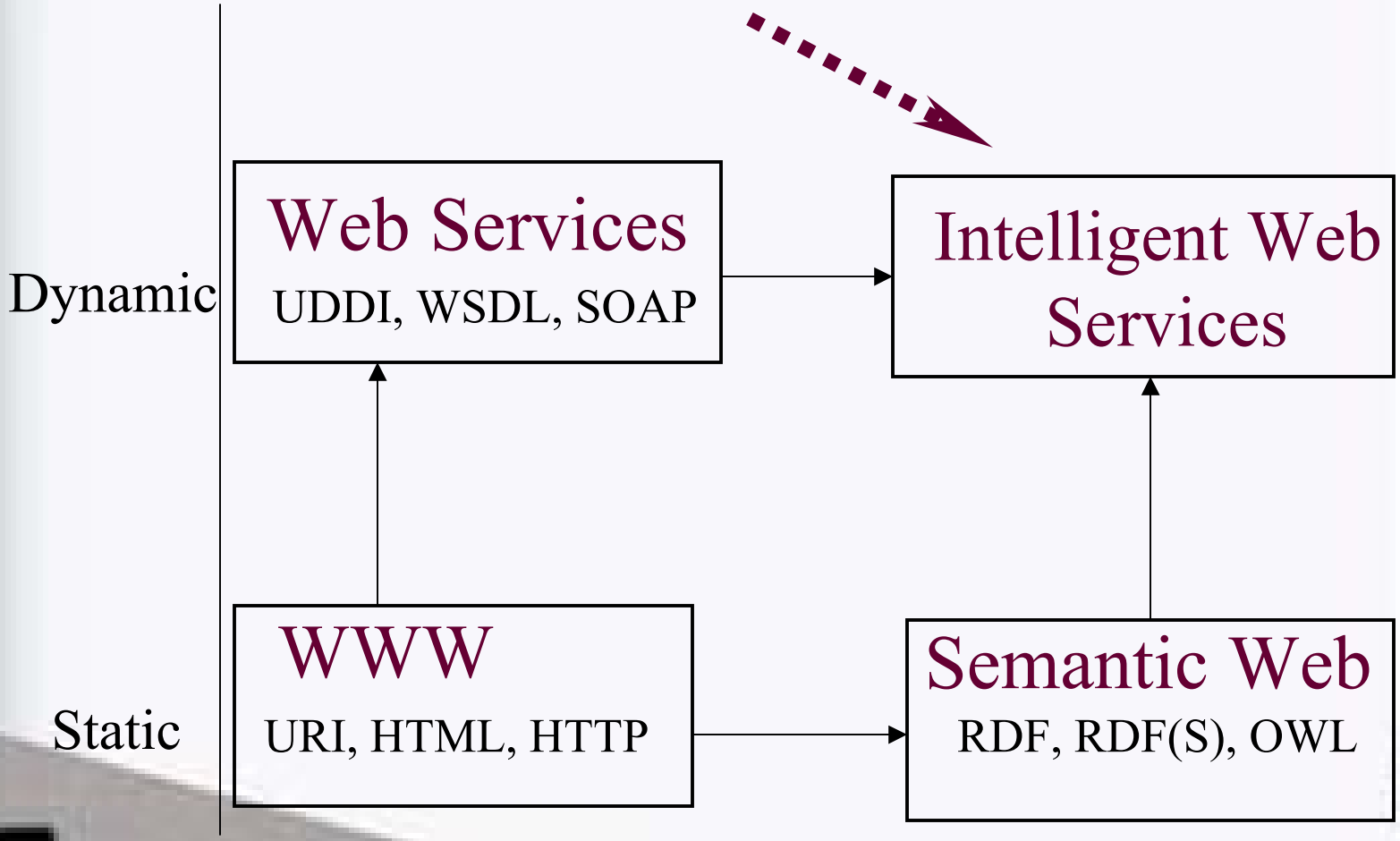

Serious Problems in information
•finding
•extracting
•representing
•interpreting
•and maintaining
Static
WWW
URI, HTML, HTTP
Semantic Web
RDF, RDF(S), OWL

The Vision


Dynamic
Static
Web Services
UDDI, WSDL, SOAP
WWW
URI, HTML, HTTP
Semantic Web
RDF, RDF(S), OWL
Bringing the computer back as a device for computation

The Vision

Bringing the web to its full potential



The Vision

"Web services are a new breed of Web application. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes. ...Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service." *IBM web service tutorial*

The Vision

==> Web Services connect computers and devices with each other using the Internet to exchange data and combine data in new ways.

==> The key to Web Services is on-the-fly software creation through the use of loosely coupled, reusable software components.

The Vision

- Software can be delivered and paid for as fluid streams of services as opposed to packaged products.
- Business services can be completely decentralized and distributed over the Internet and accessed by a wide variety of communications devices.
- The internet will become a global common platform where organizations and individuals communicate among each other to carry out various commercial activities and to provide value-added services.
- The dynamic enterprise and dynamic value chains become achievable and may be even mandatory for competitive advantage.

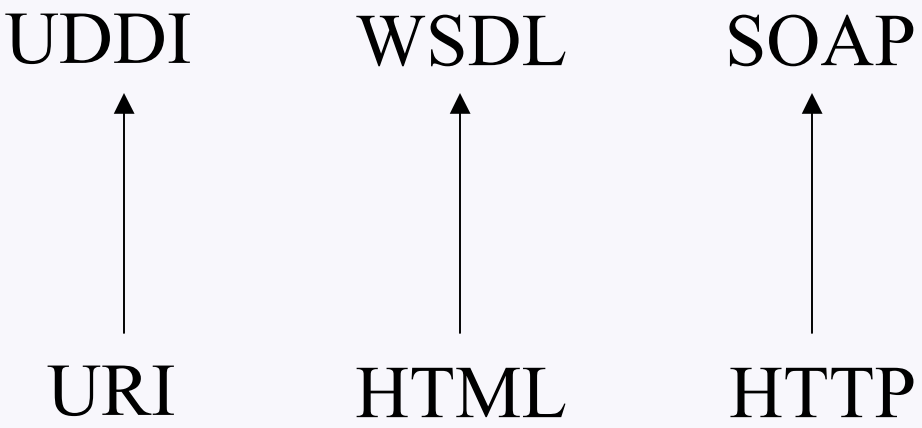
2 State of the Art

The web is organized around URIs, HTML, and HTTP.

- URIs provide defined ids to refer to elements on the web,
- HTML provides a standardized way to describe document structures (allowing browsers to render information for the human reader), and
- HTTP defines a protocol to retrieve information from the web.

==> Not surprisingly, web services require a similar infrastructure around UDDI, WSDL, and SOAP.

State of the Art



State of the Art

- **UDDI** provides a mechanism for clients to find web services. A UDDI registry is similar to a CORBA trader, or it can be thought of as a DNS service for business applications.
- **WSDL** defines services as collections of network endpoints or *ports*. A port is defined by associating a network address with a binding; a collection of ports define a service.
- **SOAP** is a message layout specification that defines a uniform way of passing XML-encoded data.In also defines a way to bind to HTTP as the underlying communication protocol for passing SOAP messages between two endpoints. SOAP is basically a technology to allow for "RPC *over the web*".

3 Requirements for Making Web Services a Working Technology

- UDDI, WSDL, and SOAP are important steps into the direction of a web populated by services.
- However, they only address part of the overall stack that needs to be available in order to achieve the above vision eventually.
- There are many layer requires to achieve automatic web service discovery, selection, mediation and composition into complex services.

Requirements for Making Web Services a Working Technology:

Document Structure

- Document types describe the content of business documents like purchase orders or invoices.
- The content is defined in terms of elements like an order number or a line item price.
- Document types are instantiated with actual business data when a service requester and a service provider exchange data.
- The payload of the messages sent back and forth are structured according to the document types defined.

Requirements for Making Web Services a Working Technology:

Semantics

- The elements of document types must be populated with correct values so that they are semantically correct and are interpreted correctly by the service requesters and providers.
- This requires that vocabulary is defined that enumerates or describes valid element values.
- For example, a list of product names or products that can be ordered from a manufacturer. Further examples are unit of measures as well as country codes.

Requirements for Making Web Services a Working Technology:

Semantics

- Ontologies provide a means for defining the concepts of the data exchanged. If ontologies are available document types refer to the ontology concepts.
- Finally, the intent of an exchanged document must be defined. For example, if a purchase order is sent, it is not clear if this means that a purchase order needs to be created, deleted or updated.

Requirements for Making Web Services a Working Technology:

Process definition

- Based on the assumption that messages can be exchanged exactly once between service requester and service provider the business logic has to be defined in terms of the business message exchange sequence.
- For example, a purchase order might have to be confirmed with a purchase order acknowledgment. Or, a request for quotation can be responded to by one or more quotes.
- These processes define the required business message logic in order to derive to a consistent business state. For example, when good are ordered by a purchase order and confirmed by a purchase order acknowledgment they have to be shipped and paid, too.

Requirements for Making Web Services a Working Technology:

Exchange sequence definition

- Communication over networks are currently inherently unreliable.
- It is therefore required that service requester and service provider make sure themselves through protocols that messages are transmitted exactly once.
- The exchange sequence definition achieves this by defining a sequence of acknowledgment messages in addition to time-outs, retry logic and upper retry limits.

Requirements for Making Web Services a Working Technology:

Transport binding

- Several transport mechanisms are available like HTTP/S, S/MIME, FTP or EDIINT.
- A service requester as well as service provider have to agree on the transport to be used when service requests are executed.
- For each available transport the layout of the message must be agreed upon and how the document sent is represented in the message sent.

Requirements for Making Web Services a Working Technology:

Security

- Fundamentally, each message exchange should be private and unmodified between the service requester and service provider as well as non-reputable.
- Encryption as well as signing ensure the unmodified privacy whereby non-repudiation services ensure that none of either service requester or service provider can claim not to have sent a message or a different one.

Requirements for Making Web Services a Working Technology:

Trading partner specific configuration

- Service requesters or service providers implement their business logic differently from each other.
- The reason is that they establish their business logic before any cooperation takes place.
- This might require adjustments once trading partners are found and the interaction should be formalized using web services.
- In case modifications are necessary trading partner specific changes have to be represented.

Requirements for Making Web Services a Working Technology

Layer / Standard	EDI	RosettaNet	ebXML	SOAP	OAGIS
Document type	X	X			X
Semantics	X	X			
Process		X	X		
Exchange Sequence		X	X		
Packaging		X	X	X	
Transport binding		X	X	X	

Requirements for Making Web Services a Working Technology

- Many organizations had the insight that message definition and exchange are not sufficient to build an expressive web services infrastructure.
- In addition to UDDI, WSDL and SOAP, standards are proposed such as WSFL, XLANG, ebXML BPSS, BPML and WSCL.

Requirements for Making Web Services a Working Technology

- Still, there are important features missing in all of the mentioned frameworks. Very important is to reflect the *loose coupling* and *scalable mediation* of web services in an appropriate modeling framework.

==> Therefore, we developed a full-fledged **Web Service Modeling Framework (WSMF)**. It provides a rich conceptual model for the development and the description of web services.

4 The Web Service Modeling Framework (WSMF)

- There are important steps to take to bring web services and fully enabled E-commerce to reality.
- Bringing E-commerce to its full potential requires a Peer-to-Peer (P2P) approach. Anybody must be able to trade and negotiate with everybody else.
- However, such an open and flexible E-commerce has to deal with many obstacles before it becomes reality!

The Web Service Modeling Framework (WSMF)

- Mechanized support is needed in finding and comparing vendors and their offers. Machine processable semantics of information allows to mechanize these tasks.
- Mechanized support is needed in dealing with numerous and heterogeneous data formats. Ontology technology is required to define such standards better and to map between them.
- Mechanized support is needed in dealing with numerous and heterogeneous business logics. Mediation is needed to compensate these differences, allowing partners to cooperate properly.

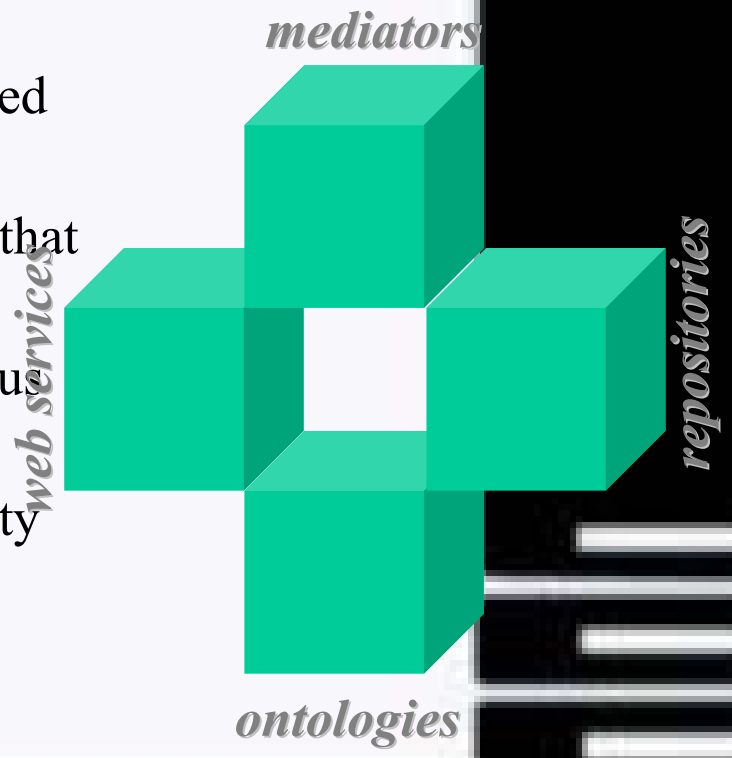
The Web Service Modeling Framework (WSMF)

- Fully enabled E-commerce based on workable web services requires a modeling framework that is centered around two complementary principles:
- Strong *de-coupling* of the various components that realize an Ecommerce application. This de-coupling includes information hiding based on the difference of internal business intelligence and public message exchange protocol interface descriptions.
- Strong *mediation* service enabling anybody to speak with everybody in a scalable manner. This mediation service includes the mediation of different terminologies as well as the mediation of different interaction styles.

The Web Service Modeling Framework (WSMF)

The WSMF consists of four main different elements:

- *ontologies* that provide the terminology used by other elements;
- *goal repositories* that define the problems that should be solved by web services;
- *web services* descriptions that define various aspects of a web service;
- and *mediators* which bypass interoperability problems.



The Web Service Modeling Framework (WSMF):

Ontologies

- Ontologies are key enabling technology for the semantic web.
- They interweave human understanding of symbols with their machine processability.
- In a nutshell, Ontologies are formal and consensual specifications of conceptualizations that provide a shared and common understanding of a domain, an understanding that can be communicated across people and application systems.

The Web Service Modeling Framework (WSMF):

Ontologies

- Thus, Ontologies glue together two essential aspects that help to bring the web to its full potential:
    - Ontologies define formal semantics for information, consequently allowing information processing by a computer.
    - Ontologies define real-world semantics, which makes it possible to link machine processable content with meaning for humans based on consensual terminologies.

The Web Service Modeling Framework (WSMF):

Ontologies

- In our framework ontologies are used to define the terminology that is used by other elements of WSMF specifications.
- Therefore, they enable reuse of terminology as well as interoperability between components referring to the same or linked terminology.

The Web Service Modeling Framework (WSMF):

Goals

- The description of a *goal* specifies objectives that a client may have in case he consults a web service. A goal specification consists of two elements:
- **Pre-conditions** describe what an web service expect for enabling it to provide its service.
- **Post-conditions** describe what a web service returns in response to its input.
- Goal specifications should be kept separate from actual web service description because there is an *n*2*m* mapping between them, i.e., the same web service can serve different goals and obviously different (competing) web services can serve the same goal.

The Web Service Modeling Framework (WSMF):

Web Services

- First, a web service has a **name**, i.e., a unique identifier to refer to him.
- Second, a web service fulfills a certain purpose, i.e., it should have a **goal reference**.
- Third, like goals, web service descriptions contain **pre conditions** and **post conditions** as introduced for goal descriptions. A web service can *strengthen a pre condition* or *weaken a post condition* of a goal.1
- Forth, a web service description describes the structure of its **input data** and **output data**.

The Web Service Modeling Framework (WSMF):

Web Services

- Fifth, **error data** can be returned from the complex service through error ports at any time to indicate problems or error states.
- Sixth, a web service in turn may invoke other web services to provide its service. For each invoked web service a proxy called **invoked web service proxy** has to be declared.
- Seventh, a web service exposes input ports and output ports. Each connection between a complex service's input port and a invoked web service proxy's input port is a **data flow**.

The Web Service Modeling Framework (WSMF):

Web Services

- Eighth, a **control flow** sequence should be introduced between the two invoked web services that defines the correct execution sequence.
- Ninth, web services may require **exception handling**. Invoked web services can fail and return an error or exception code.
- Tenth, a service need to implement a strategy of **compensation** for a failed invoked web service.

The Web Service Modeling Framework (WSMF):

Web Services

- Eleventh, web services need description related to the **message exchange protocol**. Messages from a web service requester to a web service provider and vice versa are sent over networks like the Internet. Networks can be reliable as well as unreliable.
- Twelfth, there are important **non functional properties** that characterizes a web service. Examples are the geographical reach of a service (e.g., a web-based flower shop), the price related to using a service, or the average/maximal time it may take it to produce its output.

The Web Service Modeling Framework (WSMF):

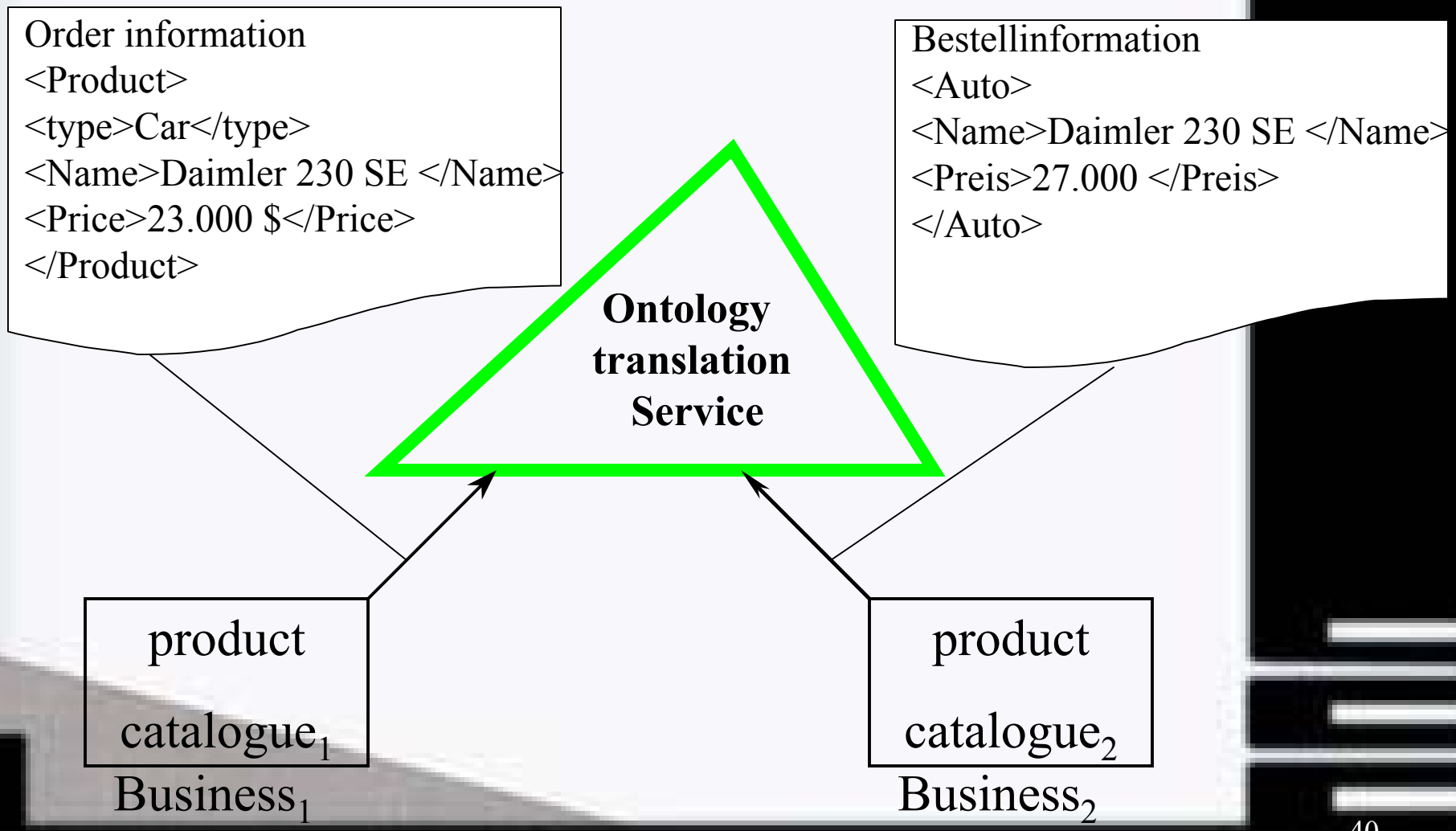
Mediators

For an open and flexible environment such as web-based computing, *adapters* are an essential means to cope with the inherit heterogeneity. This heterogeneity can wear many cloths:

- Mediation of **data structures**.
- Mediation of **business logics**.
- Mediation of **message exchange protocols**.
- Mediation of dynamic **service invocation**. A web service may invoke other web services to provide its functionality. This can be done in a hard-wired manner, however, it can also be done more flexible by just referring to certain (sub-)goals.

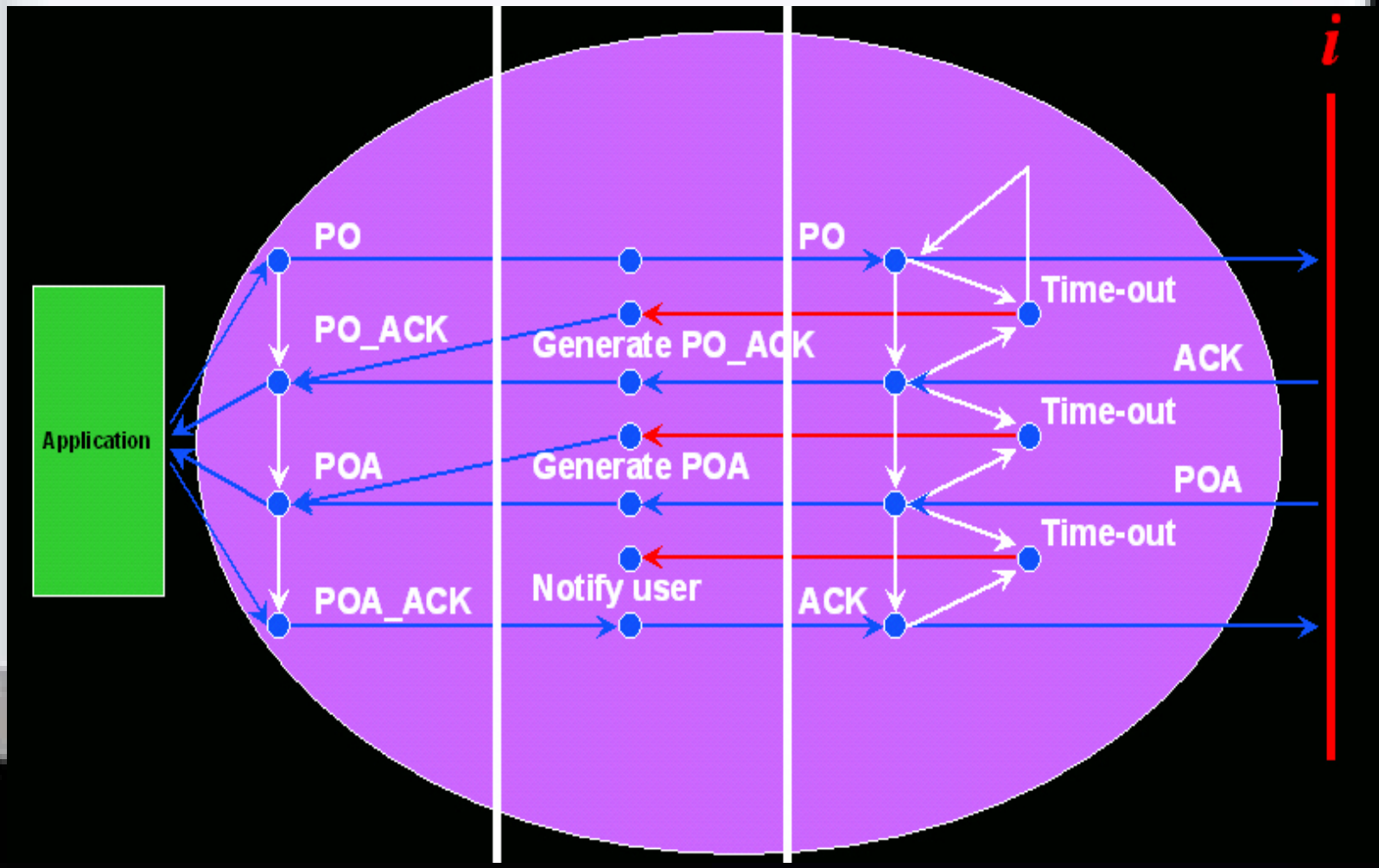
The Web Service Modeling Framework (WSMF):

Mediators



The Web Service Modeling Framework (WSMF):

Mediators



5 Semantic Web enabled Web Services

- The easy information access based on the success of the web has made it increasingly difficult to find, present, and maintain the information required by a wide variety of users.
- In response to this problem, many new research initiatives and commercial enterprises have been set up to enrich available information with machine-understandable semantics.
- This **semantic web** will provide intelligent access to heterogeneous, distributed information, enabling software products to mediate between user needs and the information sources available.

Semantic Web enabled Web Services

- **Web Services** tackle with an orthogonal limitation of the current web: Currently, the web is mainly a collection of information but does not yet provide support in processing this information, i.e., in using the computer as a computational device.
- Web services can be accessed and executed via the web.
- However, all these service descriptions are based on semi-formal natural language descriptions.
- Therefore, the human programmer need be kept in the loop and scalability as well as economy of web services are limited.

Semantic Web enabled Web Services

- Bringing web services to their full potential requires their combination with semantic web technology.
- It will provide mechanization in service identification, configuration, comparison, and combination.
- **Semantic Web enabled Web Services** have the potential to change our life in a much higher degree as the current web already did.

Semantic Web enabled Web Services

- We identify the following elements necessary to enable efficient interenterprise execution:
  - Public process description and advertisement;
  - discovery of services;
  - selection of services;
  - composition of services;
  - and delivery, monitoring and contract negotiation.

- Without mechanization of these processes, internet-based E-commerce will not be able to provide its full potential in economic extensions of trading relationships.

6 Conclusions

- In this talk:
  - we gave a vision of web service technology,
  - indicated requirements for making this technology workable,
  - and sketched our proposal, the Web Service Modeling Framework.
- Finally we want to mention some related efforts.

Conclusions

- We did not define a concrete syntax for **WSMF**. The **WSMF** language can be defined as an extensions of as **WSFL**, which is a language close in spirit to our framework.
- Also we did not define a concrete web-based syntax for **WSMF**, i.e., we did no define any web-based mark up language. Here one could take **DAML-S** as a starting point and extending it with the necessary modeling features that are missing there.
- Finally, an approach such as **PSL** could be used to define a formal semantics for the **WSMF**.

Conclusions

We are currently submitting an IST proposal
Semantic Web enabled Web Services (SWWS)
in line with the mentioned ideas.

Partners are:

- Vrije Universiteit Amsterdam, NL (coordinator)
- FZI Karlsruhe, Germany
- HP, UK
- iSOCO, Spain
- Ontotext, Bulgaria
- Oracle, U.S.A.
- Shinkatech, Germany

Lets hope it will be accepted.