Towards generic query, update, and event languages for the Semantic Web

Wolfgang May **José Júlio Alferes**
François Bry

Institut für Informatik, Universität Göttingen, Germany

CENTRIA, Universidade Nova de Lisboa, Portugal

Institut für Informatik, Ludwig-Maximilians-Universität München, Germany

PPSWR 2004, St.Malo, Sept. 8, 2004
REWERSE A3 Meeting, Hannover, Okt. 7, 2004

Motivation and Overview

This talk shows some initial steps in the REWERSE project towards languages for the Semantic Web and gives an analysis what we have to do ...

The goal of the SW is to have semantics-based query interfaces to “the Web”, or at least to fragments of the Web.

- there must obviously be some kind of a query language (although users use graphical portals, this is internally also based on the QL)

... but there is more behind “querying the Web”. Users will query “the Web”, so *it* should give an answer.

How does it do this?

There is not only *querying*, but there are *activities* going on in the Web:

- updates: in the same way as there are semantic query languages, there must be a semantic update language
- then, “the Web” must do something – it’s “the Web”, not a single page!

$\Rightarrow$ evolution of single pages (updates + reasoning)

$\Rightarrow$ evolution of of the Web

$\Rightarrow$ communication

$\Rightarrow$ reactivity: communication $\leadsto$ evolution actions

Querying "the Web"

History: 1995-2001 – Browsing

- form-based interfaces
- wrapping HTML
- following hyperlinks
- accessing multiple Web pages
- non-standardized solutions (e.g., Tsimmis, Florid, W4F, Jedi, Lixto)

Currently – Browsing + Querying

- HTML/XHTML data on the Web
- XML data accessible via Web
- standardized Query Language: XQuery/XPath
- other XML query languages: XPathLog, Xcerpt

Querying “the Web”?

- XML contents: tree data model
- combining contents from multiple autonomous sources

... this is actually “Querying XML on the Web”
(+ integration of data from the Web):

- explicit addressing of a certain Web resource that holds the information
- restricted use of inter-source references
- integration problems when combining resources
- depends on the *structure* of the provided data

$\Rightarrow$ this is not “the Web”, not semantical.

Querying "the Web"!

- independent from the actual location of information
- combination of information by "the Web" (not by the user) (e.g. portals)
- information sources must support this
- semantics-based, not syntax/data-structure-based querying

Requirements

- global data model: e.g. RDF/RDFS + OWL (+ agreements on used ontologies)
- communication for actual distributed query answering (+ mapping from local information to global format)
- global model, notions of (restricted) consistency
- global strategies for propagation of information and information *change*.

Languages: Querying

- an external query language
- powerful internal languages
  - high-level, cross-ontology mapping
  - mapping from actual data sources (XML) to higher level
- Query answering consists of
  - finding appropriate data (using metadata + communication)
  - (remotely) accessing (extensional) data
    remote queries (XML level and global level),
  - deriving information (intensional data)
  - reasoning

⇒ Modular family of (sub)languages,
Basic reactive behavior for (distributed) queries,
Rule-based languages (✓ local XML;
partly open: distributed XML, open: Semantic Web)

Languages: Updating XML in the Web

Usually, query languages are directly extended to update languages (e.g. SQL, XQuery + Updates)

- updates of local files ✓
- what does that mean on the Web – remote updates?
  - explicit statements in XQuery+Updates against a certain remote source
  - authenthication
  - transaction functionality
- remote updates by messages/method calls – mapped to local updates

⇒ simple ECA rules
update-message $\leadsto$ update
(✓; requires concepts for access control)

... behavior of “the Web” as a whole

Local Evolution of Web Nodes

- Web nodes as agents with local behavior
- react on
  - local updates
  - incoming messages
  - possibly poll/query other sources

$\Rightarrow$ describe local behavior by ECA rules
update/message + condition $\leadsto$
update (possibly including a remote query)

... so far, the (conventional) Web as a collection of separate nodes has been considered ...

- consider Web as a whole as a network of *communicating* nodes
  e.g. e-business nodes when VAT is changed:
  - a local event (update) to a certain Web source at the ministry of finance
  - an application-level event “on the Web”

  how to communicate it, detect it?
- lift the results to the Semantic Web

Global Evolution

Dependencies between different Web nodes require to propagate changes on a node of the Web:

- view-like with explicit reference to other sources
  - + always uses the current state
  - - requires permanent availability/connectivity
  - - temporal overhead
- materialize the used information
  - + fast, robust, independent
  - - potentially uses outdated information
- view maintenance strategies (web-wide, distributed)

⇒ specify and implement propagation by
ECA rules + communication/propagation strategies

Propagation of Changes

Information dependencies induce communication paths:

- direct communication: subscribe – *push*
  based on registration; requires activity by provider
- direct communication: polling – *pull*
  regularly evaluate remote query
  – yields high load on “important” sources
  – outdated information between intervals
- direct communication + view maintenance:
  regularly ask provider whether something changed (in case it maintains a log), apply view maintenance strategies
  requires some local activity by provider (logbook)

Indirect Communication

Communication via intermediate services:

- indirect communication: publish/subscribe – *push/push*
  + requires (less) activity by provider
- indirect communication: continuous queries – *pull/push*
  register query at a cq service
  + acceptable load also for “important” sources
  + shorter intervals possible

Intermediate services can add functionality:

- data integration from several services
- checking query containment
- caching

(Re)Activity & Evolution

- intended basic paradigm: reactivity
  - communication
  - specification and implementation of local behavior

$\Rightarrow$ homogeneous, modular framework

ECA Rules

- marked up in XML
- sublanguages for specifying Events, Conditions, Actions

Events

- communication events (wrapped as messages)
  - explicit queries (XML: XQuery etc.; Semantic Web: RQL etc)
  - answers (in XML/RDF)
  - other messages

    ⇒ markup language for general messages
- local events (updates on the local knowledge)

⇒ declarative extension for evolution of “the Web”:
global events “somewhere in the Web”

⇒ requires detection/communication strategies

⇒ complex events, event algebras (e.g., temporal connectives, variable bindings) , event query language, detection mechanisms

Conditions

= Queries

- local
- distributed/remote at a certain node
- distributed-global against “the Web”
- Semantic Web-level

Actions

- updates of the local state:
  - facts
  - knowledge derivation rules
  - rules describing the behavior of a node
  - all kinds of knowledge are updatable

    local evolution
- calls of procedures/services
- sending messages
- transactions

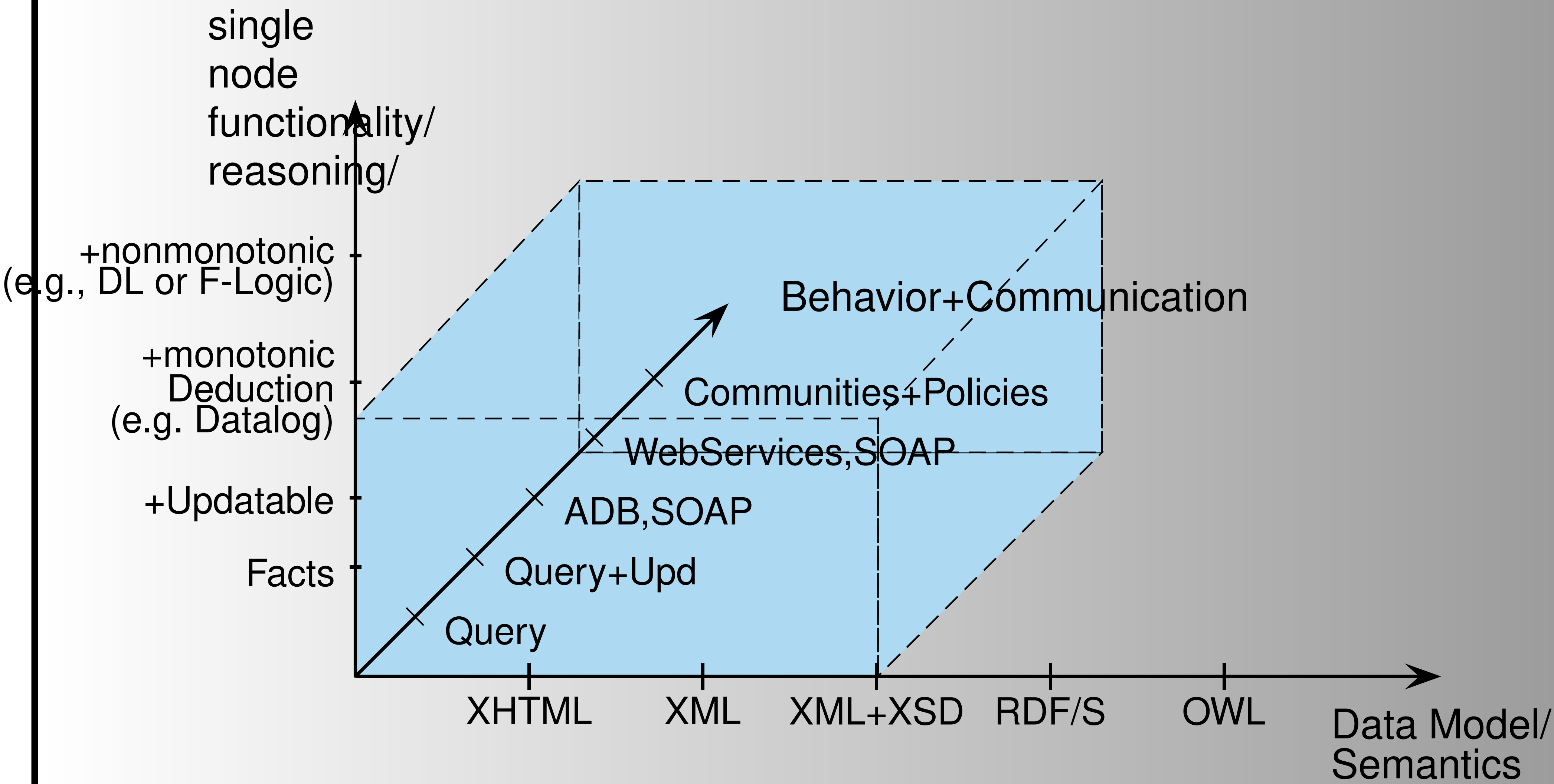
  including queries against other sources
- reasoning (about global effects, state etc.)

Global State, Reasoning and Evolution

... must be located in specific nodes, or provided by communication

- information integration + reasoning
  - data (e.g. RDF)
  - signatures (e.g. RDFS)
  - ontologies (e.g. OWL)
  - potentially each of these distributed
- global model, notions of (restricted) consistency
- consistency maintenance (by communication)
- querying: handling uncertainty and incomplete knowledge
- quality of information

Dimensions and State of the Art



Comments to the previous figure

The "dimensions" are not completely independent, but describe different axes of enhancing technology from simple Web pages to comprehensive Web nodes.

1. (horizontal axis) "built-in" data model of the nodes: from simple HTML pages up to OWL/XML data sources.
2. (vertical axis) functionality and reasoning "around" the data model: from nothing up to complex reasoning.
   Everything from (1) that is at least "XML" can be equipped with reasoning; some reasoning capabilities comes already with the horizontal axis (RDFS/OWL).
3. communicative functionality of nodes – from single nodes over Web Services to communities.

Current Web:

- mainly: HTML/XML, facts, Browsing/Querying
- less frequent: XML+XSD, some simple behavior (deduction/views), SOAP/Web Services

Perspectives

BLA^{TM} (Be Lazy Approach):

- use concepts and components that are available for prototyping
- stepwise extension in each of the dimensions
  - Semantic Web query languages
  - reasoning mechanisms for the Web
  - communication and complex events
  - uniform, modular ECA-based environment
- if necessary, develop better concepts/components