# Ten Theses on Logic Languages for the Semantic Web

François Bry[1] and Massimo Marchiori[2,3]

[1] Institut für Informatik, Ludwig-Maximilians-Universität München
[2] Dipartimento di Informatica, Università ca' Foscari, Venezia
[3] World Wide Web Consortium

29th May 2005

> *"For the semantic web to function, computers must have access to [...] sets of inference rules that they can use to conduct automated reasoning."* — Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web, Scientific American, May 2001.

Theses on Logic Languages for the Semantic Web, referring to:

- Languages
- Data and Data Processing
- Language Semantics
- Engineering and Rendering

# Contents

# Thesis 1

## Diversity

### Diversity

The Semantic Web requires Logic Languages of different kinds:

1. three kinds of reasoning, or deductive, languages, viz.
    1. constructive rules (or "views")
    2. normative rules (or "integrity constraints")
    3. descriptive specifications (or "ontologies")
2. reactive rules

# Thesis 2

## Negation

### Negation

- Non-monotonic negation is the negation of choice for constructive rules (views), normative rules (integrity constraints), and reactive rules.
- Monotonic negation may, but needs not, be offered in constructive, normative, and reactive rules.
- Monotonic negation is the negation of choice for descriptive specifications (ontologies).

# Thesis 3

## Coherency and Inter-Operability

### Coherency and Inter-Operability

Inter-operable logic languages of the various kinds should be striven for. Inter-operability is sustained by the following form of coherency: syntax coherency, rendering coherency, reasoning coherency, and explanation coherency.

# Thesis 4

## Data Distribution and Versality, Meta-Level Reasoning

### Data Distribution and Versality, Meta-Level Reasoning

A logic language for the Semantic Web must

- access data everywhere on the Web;
- be capable of accessing data and meta-data in any common Web and Semantic Web format – especially XML, RDF, Topic Maps, and OWL, as well as the formats of Semantic Web logic languages, and
- be capable of some form of meta-level reasoning

# Thesis 5

Reasoning Paradigms

## Reasoning Paradigms (a)

Constructive and normative rules (views and integrity constraints) should be evaluable by both forward and backward chaining, backward chaining being the reasoning of choice.

## Reasoning Paradigms (b)

Descriptive specifications (ontologies) call for (non-constructive) reasoning, including *excluded middle*, *non-contradiction*, and *refutation*.

## Reasoning Paradigms (c)

The reasoning paradigms of Semantic Web logic languages should support grouping, aggregation, theory reasoning, and non-monotonic negation.

# Thesis 5
Reasoning Paradigms

## Reasoning Paradigms (a)

Constructive and normative rules (views and integrity constraints) should be evaluable by both forward and backward chaining, backward chaining being the reasoning of choice.

## Reasoning Paradigms (b)

Descriptive specifications (ontologies) call for (non-constructive) reasoning, including *excluded middle*, *non-contradiction*, and *refutation*.

## Reasoning Paradigms (c)

The reasoning paradigms of Semantic Web logic languages should support grouping, aggregation, theory reasoning, and non-monotonic negation.

REWERSE reasoning on the web

# Thesis 5
Reasoning Paradigms

## Reasoning Paradigms (a)

Constructive and normative rules (views and integrity constraints) should be evaluable by both forward and backward chaining, backward chaining being the reasoning of choice.

## Reasoning Paradigms (b)

Descriptive specifications (ontologies) call for (non-constructive) reasoning, including *excluded middle*, *non-contradiction*, and *refutation*.

## Reasoning Paradigms (c)

The reasoning paradigms of Semantic Web logic languages should support grouping, aggregation, theory reasoning, and non-monotonic negation.

# Thesis 6

Event Processing

## Event Processing

Event broadcasting is undesirable on the Web. Events can be exchanged between Web sites using a push or a pull model. Pushed events can be sent as data streams, calling for streamed query evaluation methods. Evaluating event queries, e.g. the event parts of ECA rules, calls for event driven query evaluation methods.

# Thesis 7

## Declarative Semantics

### Declarative Semantics

Logic languages for the Semantic Web, except reactive rule languages, should have a declarative semantics defined as 'Tarski-style model theories'.

# Thesis 8

## Operational Semantics

### Operational Semantics

The operational semantics of a logic language is conveniently expressed with constructive and normative rules. Backtracking is useful for a fine tuning of proof construction in implementing logic languages.

# Thesis 9

## Language Engineering

### Language Engineering

Logic languages for the Semantic Web should be referentially transparent, strongly closed, have Web formats, and modern type systems. The specification of abstract machines should be striven for.

# Thesis 10

## Visual and Verbal Rendering

### Visual and Verbal Rendering

Logic languages for the Semantic Web should have visual and verbal renderings.

# REWERSE's approach

- Xcerpt: deductive query language for Web data with constructive rules
- XChange: reactive language for event handling and processing
- both languages share a common core for both syntax and semantics
- type systems with static type checking developed for both languages:
  - $R^2G^2$ for specifying graph grammars
  - CaTTs for specifying and reasoning with calendric types
- policy specification and reasoning
- verbalisation and visualisation of Semantic Web Data, Xcerpt, and XChange

# Thank You!