Is Semantic Technology making things worse?
–
Preliminary answer: yes!

Christoph Bussler
(speaking for myself)

International Workshop on Service Composition & SWS Challenge
(SerComp & SWS Challenge '07)

November 5, 2007, Silicon Valley, USA

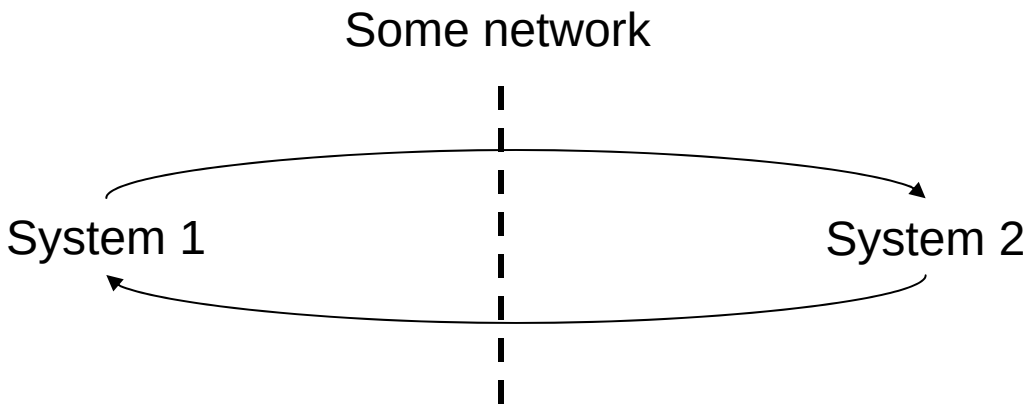
Goals and Promises of the Semantic Web (Technology)

- Seamless interacting agents (people, services)
  - Based on reliable communication
  - Uniform data and process semantics
- Overcoming heterogeneity and interoperability problems in data and processes
  - Dynamic and automatic discovery / composition combined with data / process mediation
- Achieving semantic correctness and dependability
  - Including trust and explanation of reasoning results

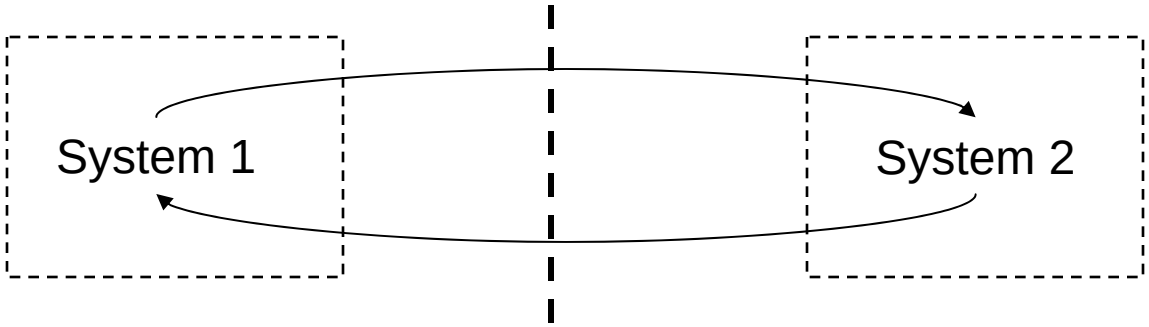
Goals and Promises of the Semantic Web (Technology)

- Main goal is semantically correct interoperability
  - Not just on the user interface, but end to end (!)
  - Machine-understanding is important, too
    - In the future, but not essential today
    - Automatic (service) discovery is not the most important problem for quite a few quarters ahead

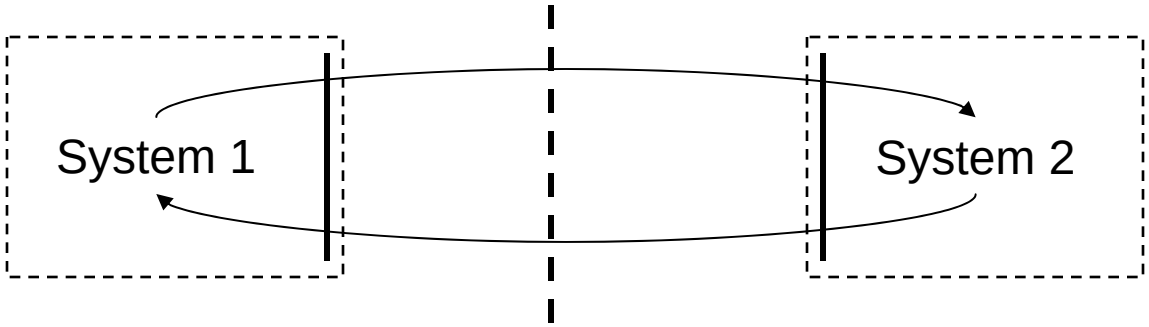
Meaning of “End-to-End” – EAI and B2B



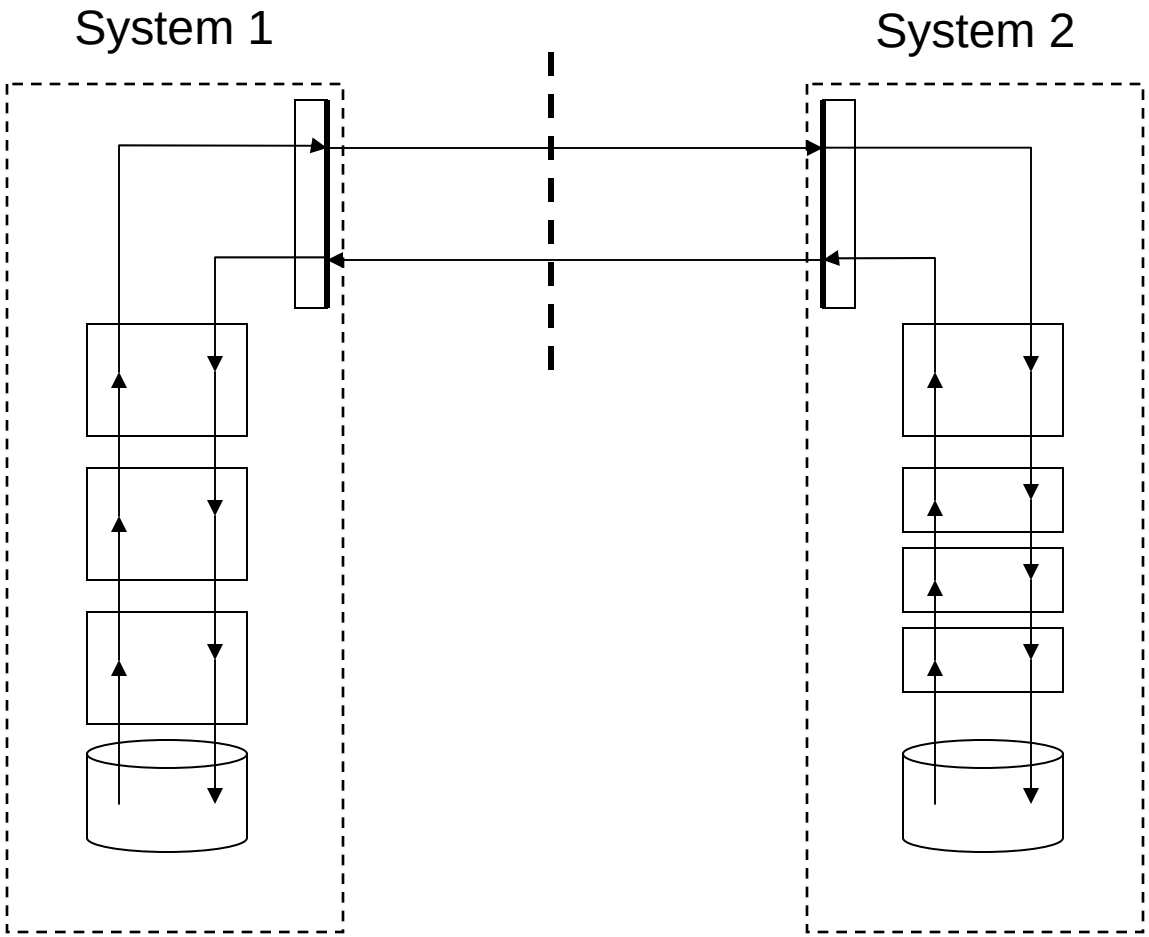
Meaning of “End-to-End” – EAI and B2B



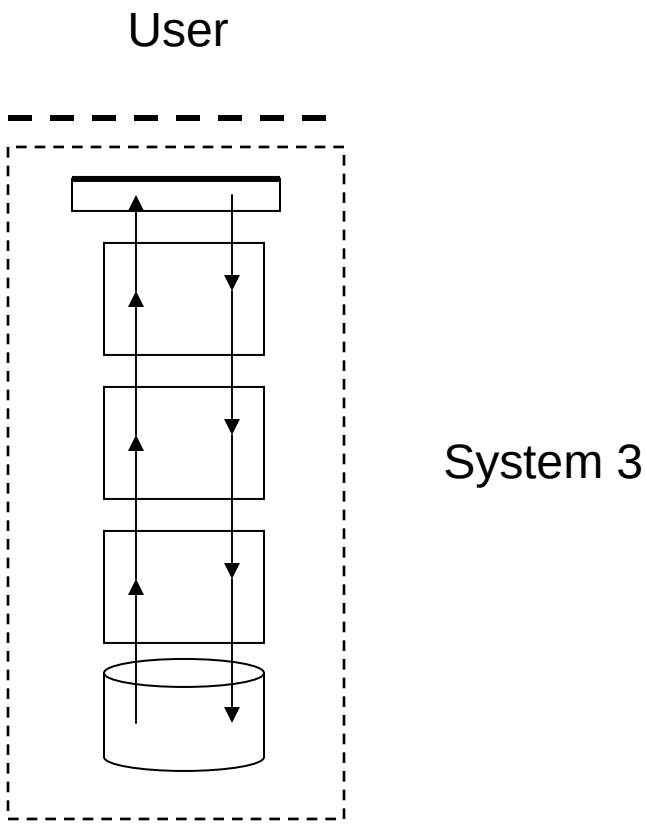
Meaning of “End-to-End” – EAI and B2B



Meaning of “End-to-End” – EAI and B2B



Meaning of “End-to-End” – UI



Examples (I)

- Address
  - If country == “Ireland”
  - Then address must not have zip code

Examples (I)

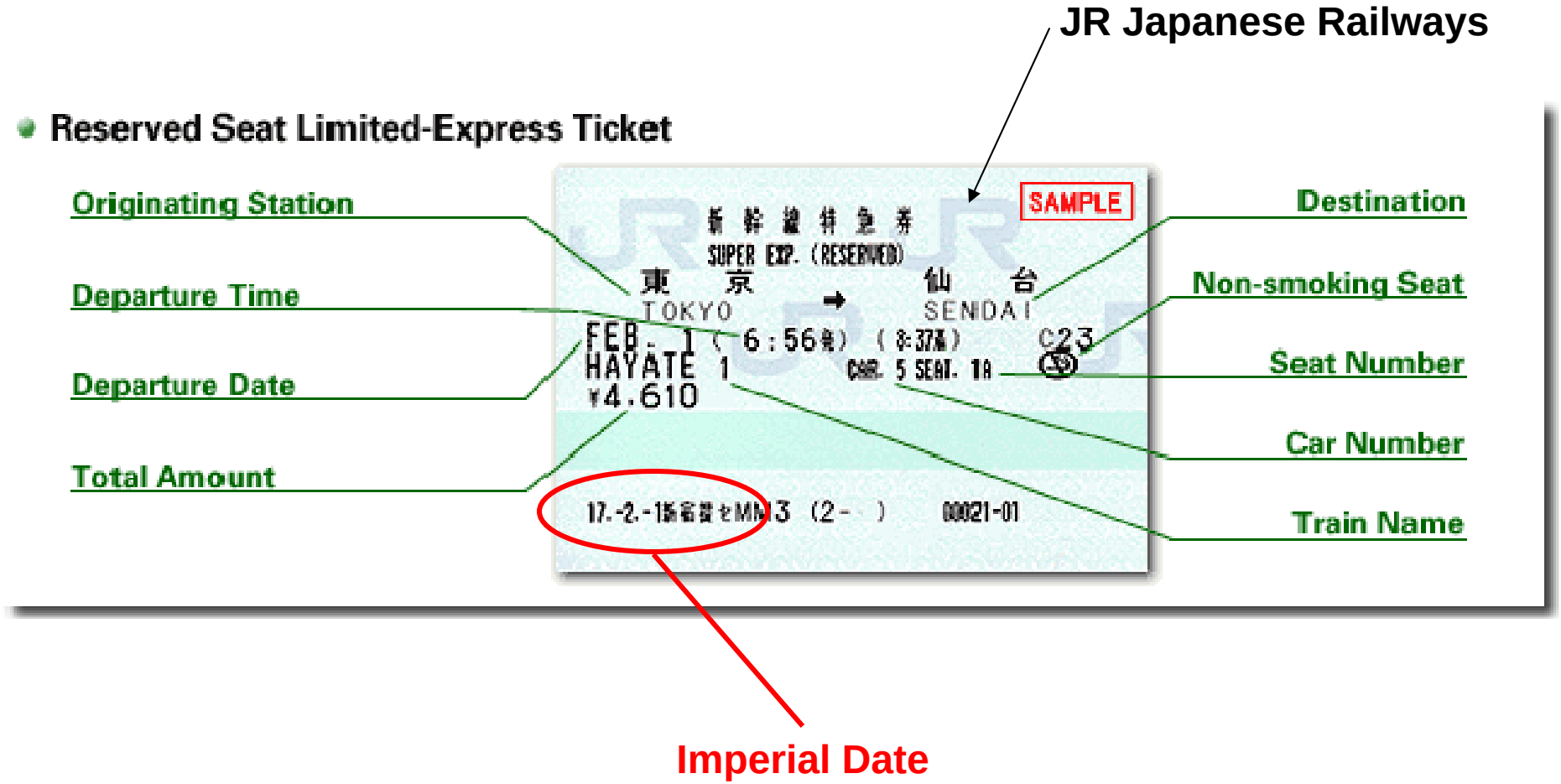
- Address
  - If country == “Ireland”
  - Then address *****must not***** have zip code
- “Must not”
  - Don’t ask for it on user interface
  - Don’t ask for it in any B2B or EAI connection
  - Don’t store any zip code value except maybe “n/a”
  - If values are provided then cause an error or an exception or (consistently) ignore the values

Examples (II)

- Date
  - If date is in Japanese context
  - Then year can be Gregorian or Imperial
- Gregorian or Imperial means
  - Value for this year either 2007 or 19
  - Or 1 if change in Era
    - For example, January 7th, 1989, area changed from Showa (64) to Heisei (1)
  - Additionally, the complete Imperial date has an identifier for the Era

Examples (II)

- Reserved Seat Limited-Express Ticket



Examples (III)

- amazon.de and amazon.com
  - For adding addresses in DE and US different drop down lists for countries are provided. Entries corresponding to locale (“Irland” vs. “Ir**e**land”)
  - Values of entries (!) are stored in address book, not enumeration index in drop down list
  - If you select ‘Irland’ in amazon.de, then ‘Irland’ shows up in the address in amazon.com, without the ‘e’

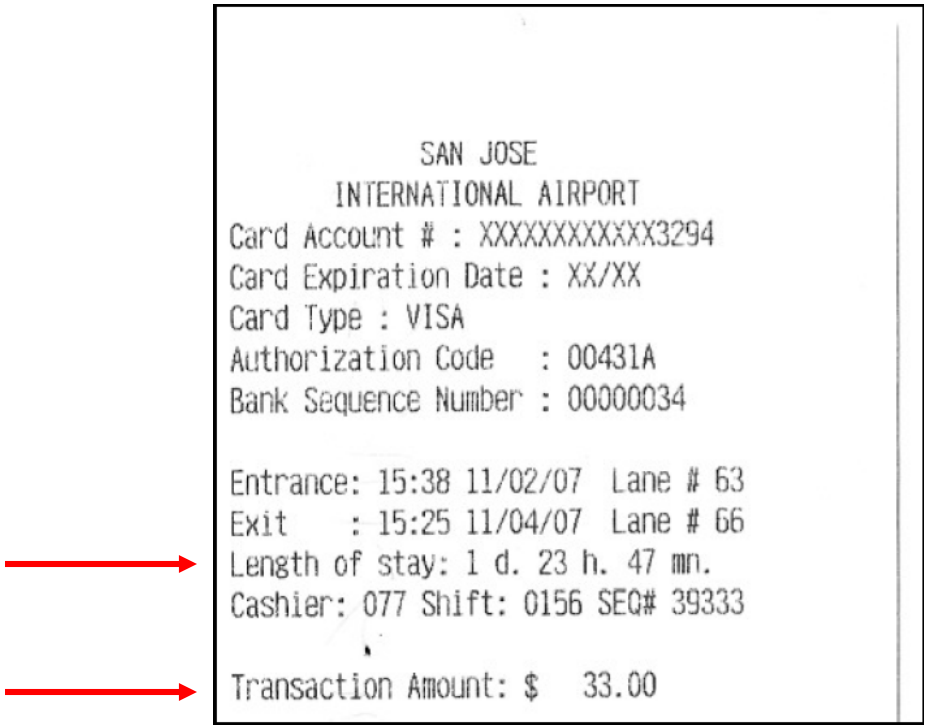
Examples (IV)

- Ordering process
  - “Free” parts
- Shipping process
  - Shipper tracking number

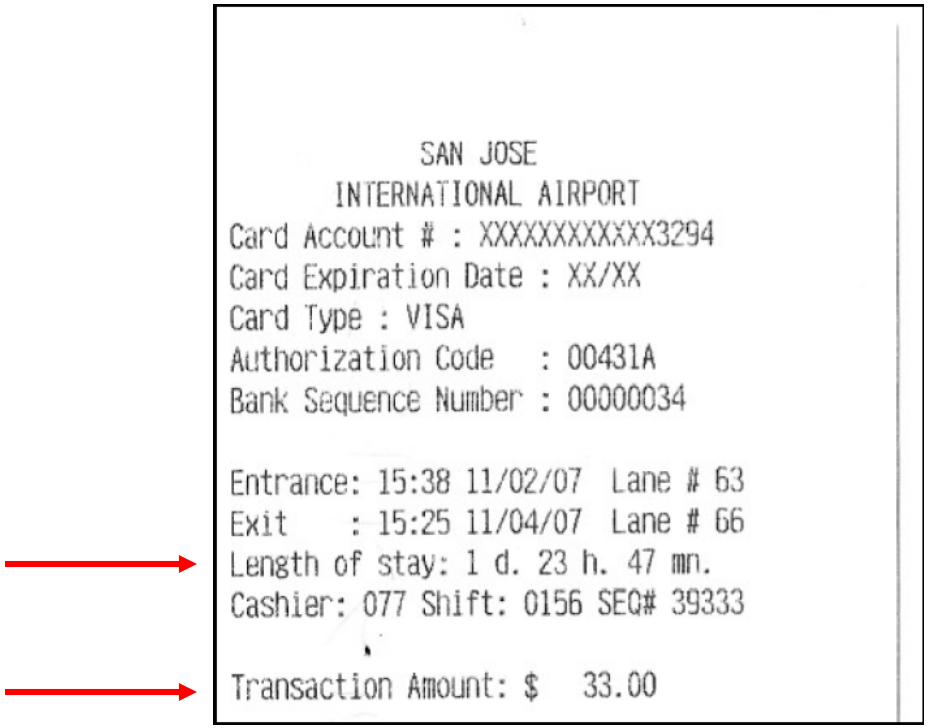
Examples (V - highlight)

- Parking
  - 1 USD per 20 Minutes
  - 15 USD max per 24-hour period

Examples (V - highlight)



Examples (V - highlight)



SAN JOSE
INTERNATIONAL AIRPORT
Card Account # : XXXXXXXXXXXX3294
Card Expiration Date : XX/XX
Card Type : VISA
Authorization Code : 00431A
Bank Sequence Number : 00000034

Entrance: 15:38 11/02/07 Lane # 63
Exit : 15:25 11/04/07 Lane # 66
Length of stay: 1 d. 23 h. 47 mn.
Cashier: 077 Shift: 0156 SEQ# 39333

Transaction Amount: $ 33.00

1 USD per 20 minutes
15 USD max per 24-hour period

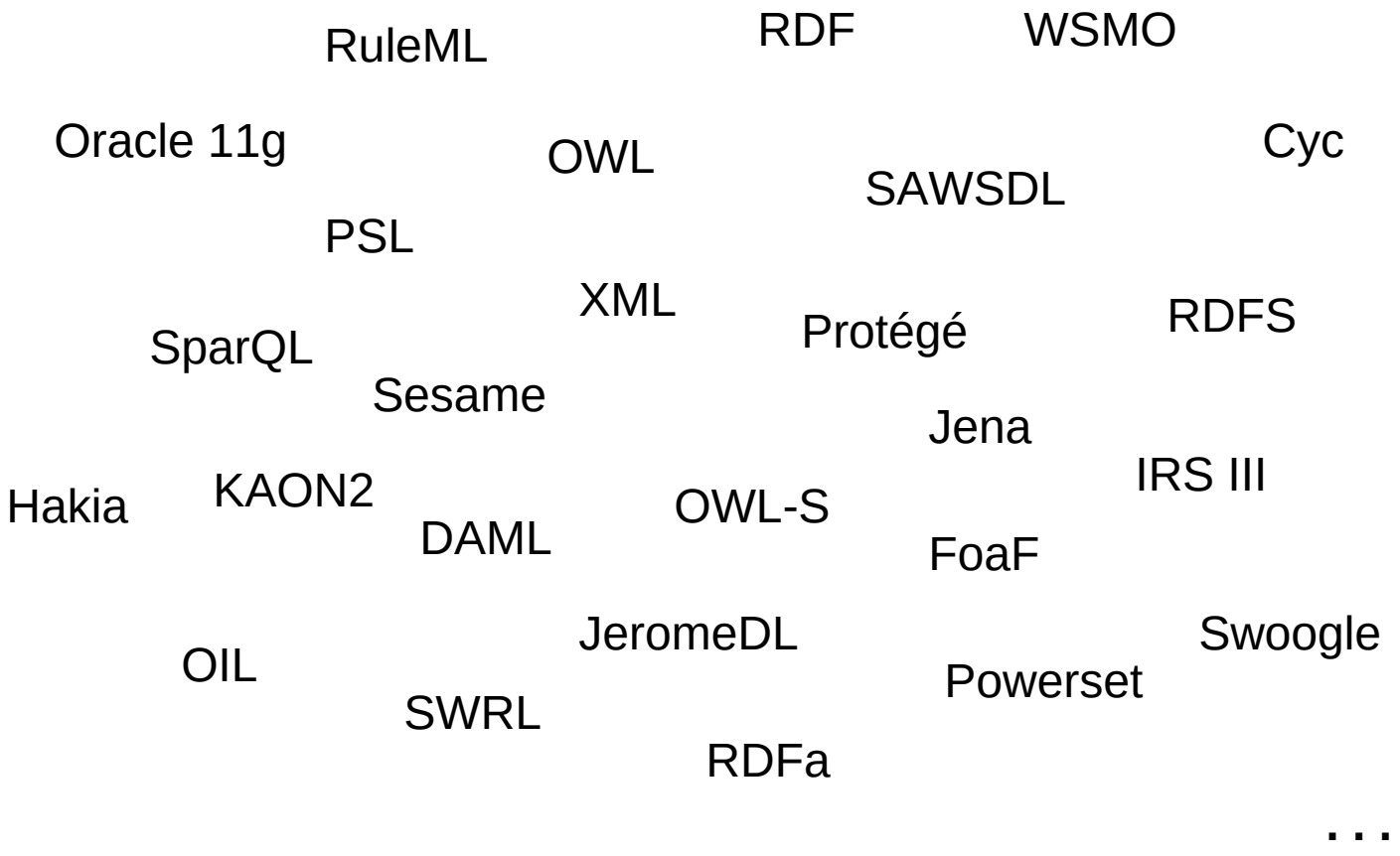
Interoperability

- If two systems, components, etc. “exchange” data, then both should have the same understanding
  - Date or address, line item or shipping number
  - End-to-end
- Major problem
  - Pass-by-value does not pass on the ‘constraints’, but only the structure / values of data

Interoperability

- Pass-by-value
  - How should a ‘downstream’ component realize that Imperial dates are valid data values?
  - How should a ‘downstream’ system realize that the first digit is missing from the tracking number?
- All mapping approaches are based on data structure and pass-by-value today
- Happy to hear that Semantic Web Technology solves this problem as its goal is semantic interoperability
  - How?

Semantic Web Technology Cloud



Industrial Application System Architecture

- Layers
  - Functionality abstraction
  - Functionality separation
- Technology specialization
  - Domain specific languages and technologies are developed for the specific layers leading to conceptual distance
- 7 layers generally accepted
  - Can be found in real systems
  - Are supported by dedicated technology
    - Popular non-enterprise stack is “LAMP”

Industrial Application System Architecture

- 7 Layers in general software architecture
  - UI (User interface)
  - UID (User interface driver)
  - BL (Business logic)
  - BR (Business rules)
  - BP (Business process)
  - BO (Business object)
  - PS (Persistence system)

Industrial Application System Architecture

- 7 Layers in general software architecture
  - UI (User interface)
  - UID (User interface driver)
  - BL (Business logic)
  - BR (Business rules)
  - BP (Business process)
  - BO (Business object)
  - PS (Persistence system)
- Example implementation technologies
  - Layout: Jsf
  - Dialog: Jsf configuration
  - Manipulation: Java
  - Rules: Jess
  - Workflow: BPEL
  - Persistent object: JPA
  - Database: Postgres

Industrial Application System Architecture

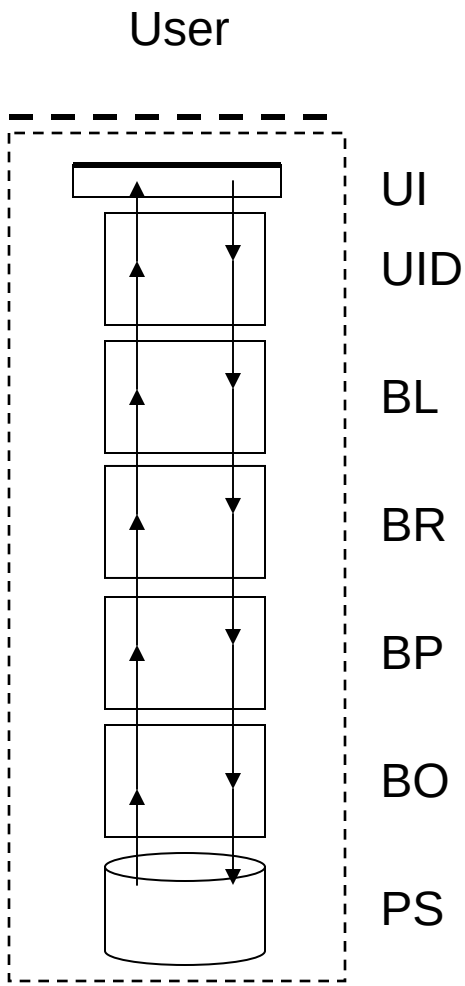
- 7 Layers in general software architecture
  - UI (User interface)
  - UID (User interface driver)
  - BL (Business logic)
  - BR (Business rules)
  - BP (Business process)
  - BO (Business object)
  - PS (Persistence system)
- Integration
  - UI and UID replaced by B2B interface and protocol
- Example implementation technologies
  - Layout: Jsf
  - Dialog: Jsf configuration
  - Manipulation: Java
  - Rules: Jess
  - Workflow: BPEL
  - Persistent object: JPA
  - Database: Postgres
- Integration technology
  - RosettaNet

Industrial Application System Architecture

- Data type system
  - Each implementation technology in each layer has its own data type system
- Data model
  - Some components have their own data model defining ‘some’ semantics like ‘date’
- Execution model
  - Each technology in each layer has its own execution model

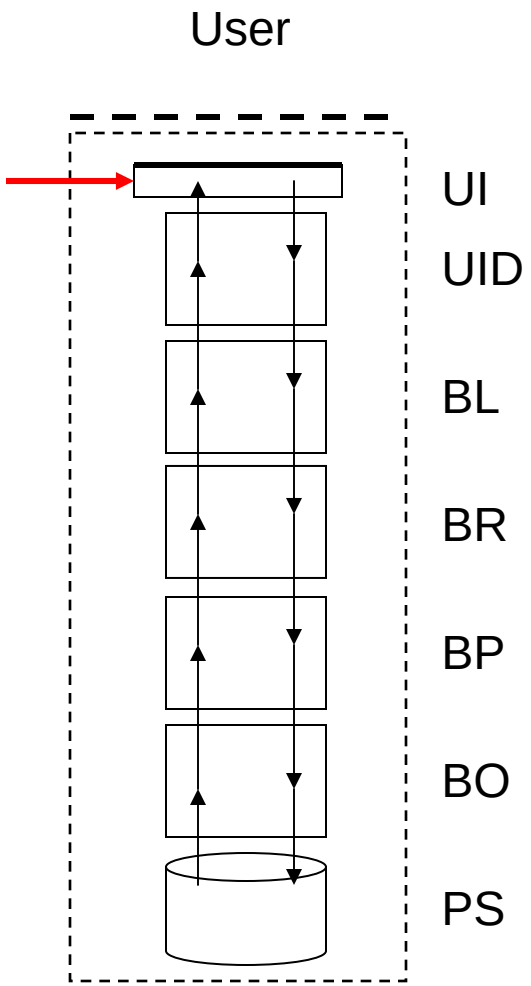
Industrial Application System Architecture

- Constraints
  - E.g. Irish addressed must not have zip codes
  - In BL? BR? PS? BO?



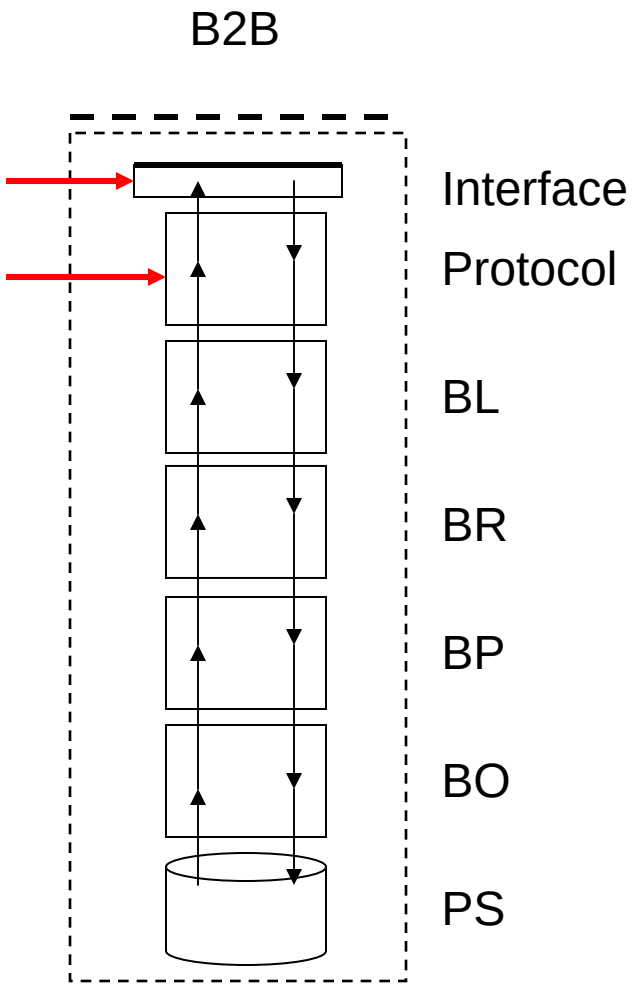
Industrial Application System Architecture

- Constraints
  - In BL? BR? PS? BO?
- UI-based system
  - Constraints implemented in UI



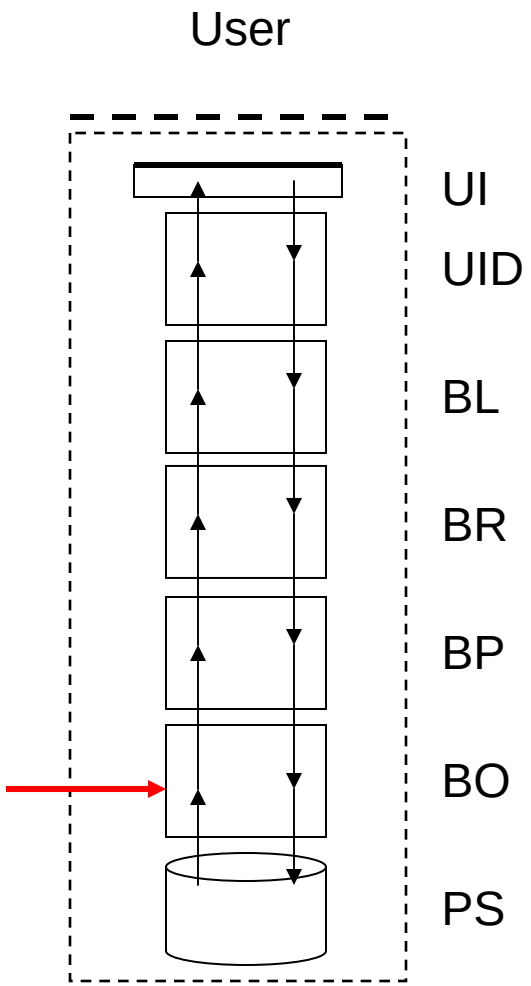
Industrial Application System Architecture

- Constraints
  - In BL? BR? PS? BO?
- Integration
  - Constraints implemented in layers equivalent to UI and UID



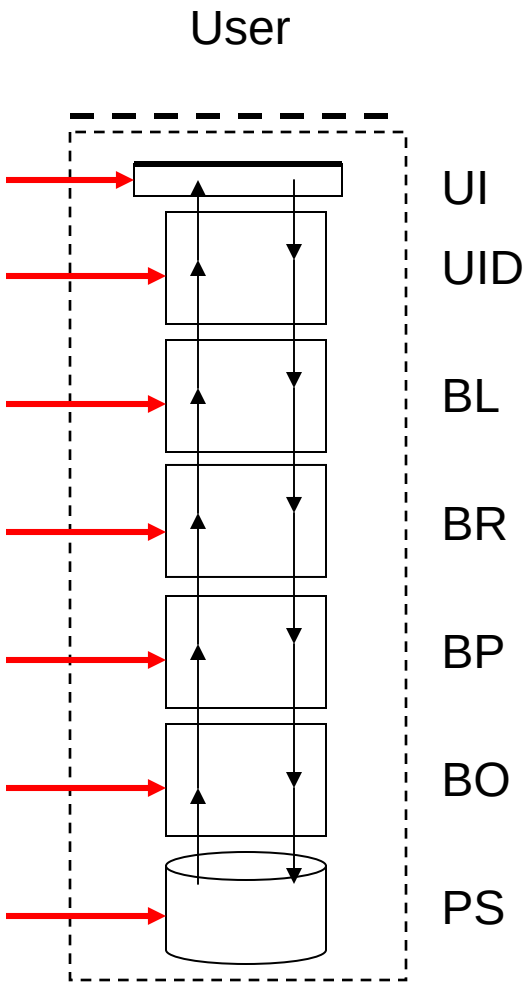
Industrial Application System Architecture

- Constraints
  - In BL? BR? PS? BO?
- Reuse
  - If BO component is shared the constraints must be in this component



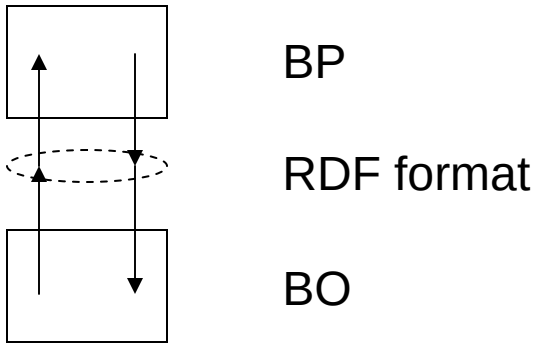
Industrial Application System Architecture

- Constraints
  - In BL? BR? PS? BO?
- Semantic interoperability across layers and systems
  - Constraints must be implemented in all layers, UI, UID, BL, BR, BP, BO AND PS (!)



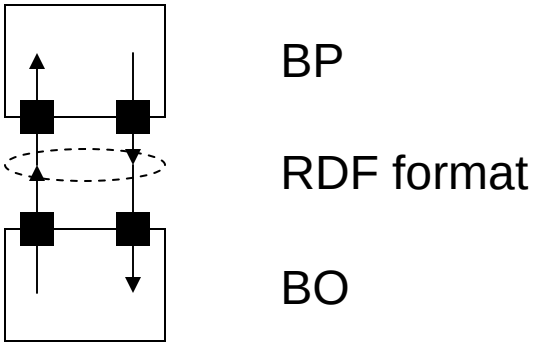
Analysis

- Semantics to be implemented in all layers
  - Consistently
- Pass-by-value and transformation
  - Deals with data structure and values only ☹
- Suggestion by SW community
  - Use RDF for communication



Analysis

- Problem 1: still only data structure and values
- Problem 2: number of mediations required



- 4 mediations per interface = 24 total for one round-trip from UI to PS
  - Significant number of mediations (!)

Analysis

- RDF as communication language between two components
  - Causes 4 transformations between up to 3 languages
- Still requires constraints to be implemented consistently in all layers
- Wait, what about SWT?
  - No difference ☹, as SWT follows the layer and component architecture

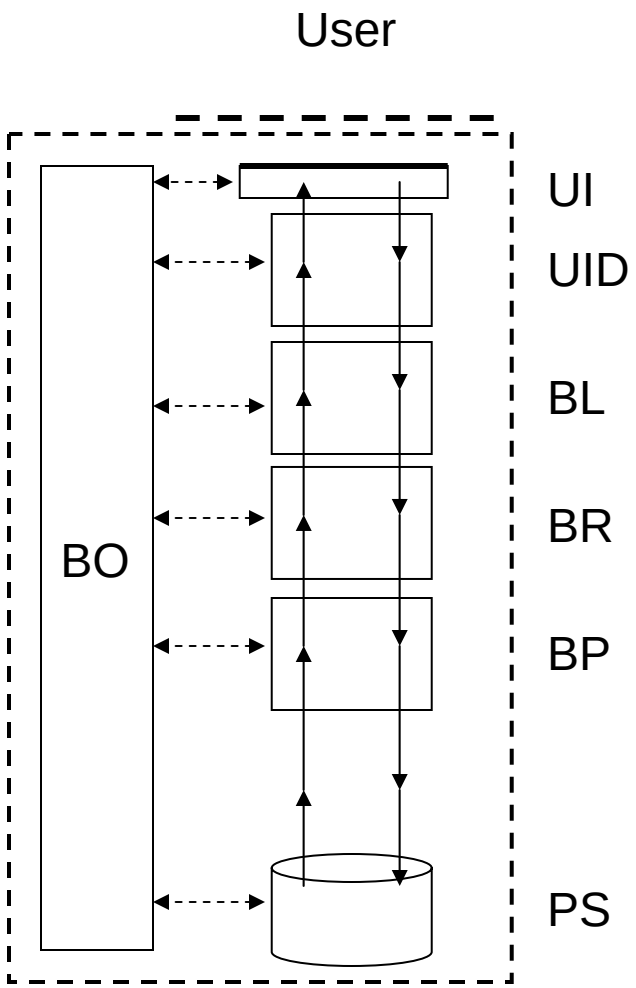
Naïve Solution

- Do not pass-by-value, but pass-by-reference
  - Avoids data structure and value mediation
- Lot's of problems remain to be solved if RDF (or any other language) is used
  - Common representation is a plus
  - Constraints still not captured
  - Concurrency?
  - All layers must be able to operate on RDF
- BTW, original Scientific American article tried to convince that all layers understand RDF
  - And follow-up article tried to convince that data standards (with their constraints) are important, too

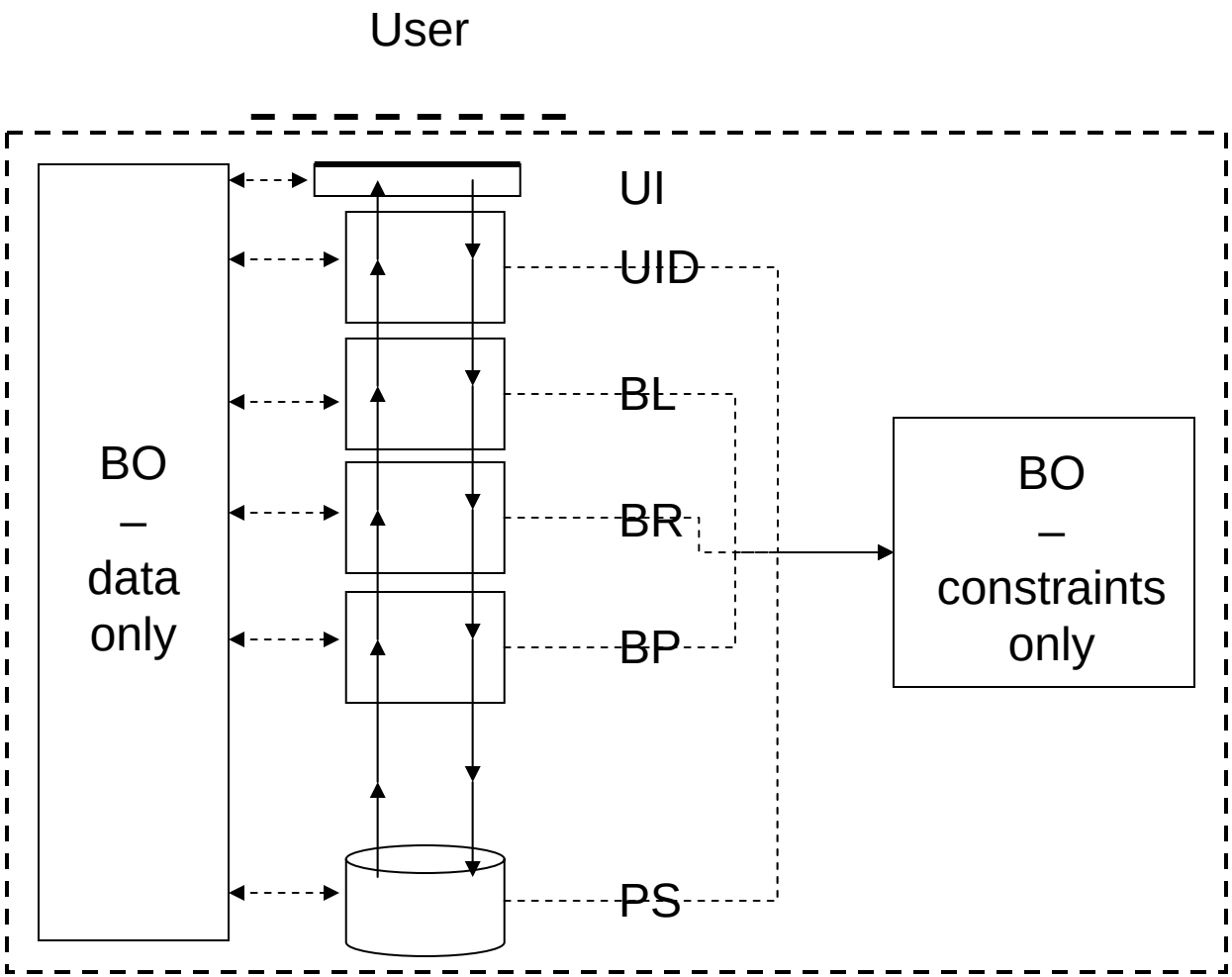
Back-to-Basics Approach

- Parameter passing
  - Is parameter passing the adequate approach to achieve semantic interoperability?
- Constraints location
  - Constraints are part of the data definition (not separate)
  - Constraints are passed within parameters automatically
- Alternative
  - Separation of data and control flow between layers
  - No passing-by-value, but passing-by-reference

Back-to-Basics Approach (I)



Back-to-Basics Approach (II)



Back-to-Basics Approach

- Address
  - If country == Ireland selected, then BO does not accept zip code values from any layer
- Date
  - If ’17’ is provided for the year field then system can ask back in every layer if 2005 or 1942 (or any other valid era is meant)
- Selector List
  - User selection returns index into list of countries (i18N data is maintained separately)
- Ordering process
  - Manufacturing order is a separate BO than sales order (clear separation)
- Shipping process
  - Passing tracking number by reference itself solves the problem

Summary

- Current SWT efforts ignore industrial application system architecture and programming completely
  - And surrounding eco-system as well
- Programming model is necessary
  - Rethink programming and architecture approach?!
- If SWT is the proposed solution
  - Must fit into existing frameworks
  - Or has to establish own complete (!) programming framework

Thank You