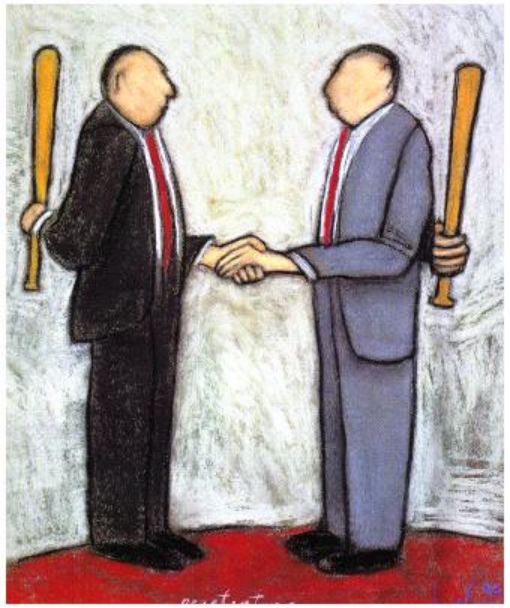
Group 4 – Topic 5

Service composition

Christoph Paur, Ondrej Balún, Dominik Kertys

QoS based service selection in WS Composition

- Optimization-based – local/global
- Negotiation-based – automated process by negotiator agent
- Hybrid approaches
  - Matchmaking
  - Provider selection
  - Agreement configuration



Main goal

- build a service composition that integrates different existing or yet-to-built Web services

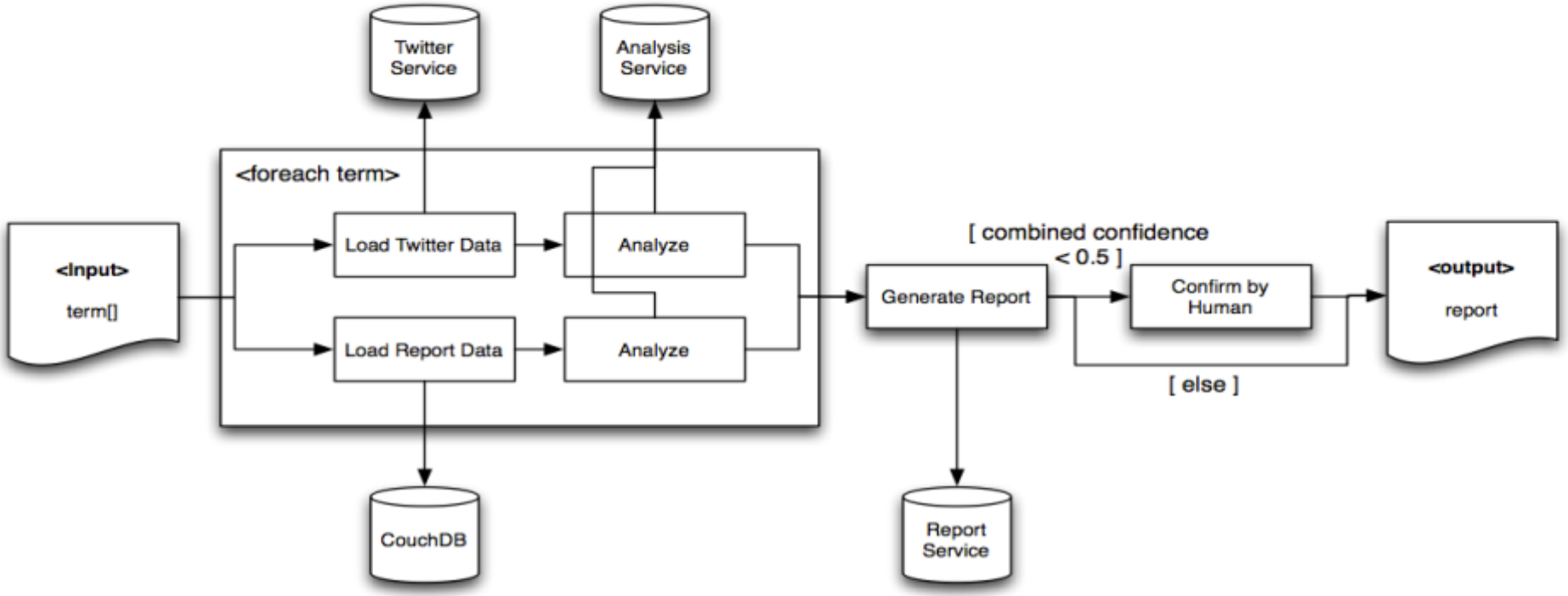


Figure 1: Sentiment Analysis Process Sketch

Composition IDE Tools

- NetBeans 6.5.1 with SOA(service-oriented architecture)
  - GlassFish ESB v2.1 – free standard application server
  - easy drag&drop editor
  - composite app – includes multiple BPEL Modules and other types of Java Business Integration (JBI) modules
  - easy test integration
  - easy to implement own Java services

Our services

- Twitter service
  - Custom Web Service (handles OAuth)
  - troubles with SSL certificate
- CouchDB service
  - Custom Web Service (handles Search function generation)
- HumanActivity service
  - Receives low confidence reports from BPEL Web Service
  - Returns confirmed or altered report depend. on Human choice

GUI

- JAXB Binding – Java Architecture for XML binding
- JSP
- deploying of WSs to Glassfish
- Analyse servlet – executes the composition
- ViewPdf servlet – enables PDF functionality
- Confirm by Human – accept,reject,upload PDF

BPEL - Advantages

- Really simple cases work
- Automatic Generation of Process Flow
  - Impress your CEO
- Lots of presentations with corporate fuzz
  - May help with decision makers
- Language Focus *Process Orchestration*
  - Compensation-Handlers
    - Run on *finished* scopes if a fault happens *later*
    - Necessary when no locks can be held for rollback

BPEL - Disadvantages

- Even simple tasks take forever
- More complex tasks quickly turn unwieldy
  - Complex in the BPEL sense
    - E.g. Merge two Lists or Arrays
    - Check if a variable is unassigned (null)
- Graphical designer cannot hide the complexities of BPEL
  - Complex IF conditions
  - Data Conversions (XML Namespaces!)

BPEL - Disadvantages

- Need to switch between Graphical Designer and Source Code
  - No Auto-Complete for Variables and Properties
  - Namespaces unintuitive
- Multiple Technologies involved
  - Obviously: BPEL
  - WSDL, XSD, XSLT, …
  - WS-Adressing, WS-Security, …
- Debugging is extremely cumbersome
  - Deploying to BPEL Engine takes a while
  - Data stored in XML

BPEL - Problems

- Not really platform-agnostic
  - Every vendor has their own set of extensions
    - E.g. <bpelx:append>, :remove>, :copyList>
    - Address most common Shortcomings
  - Leads to vendor-specific „code “
  - BPEL-Specification not fully supported

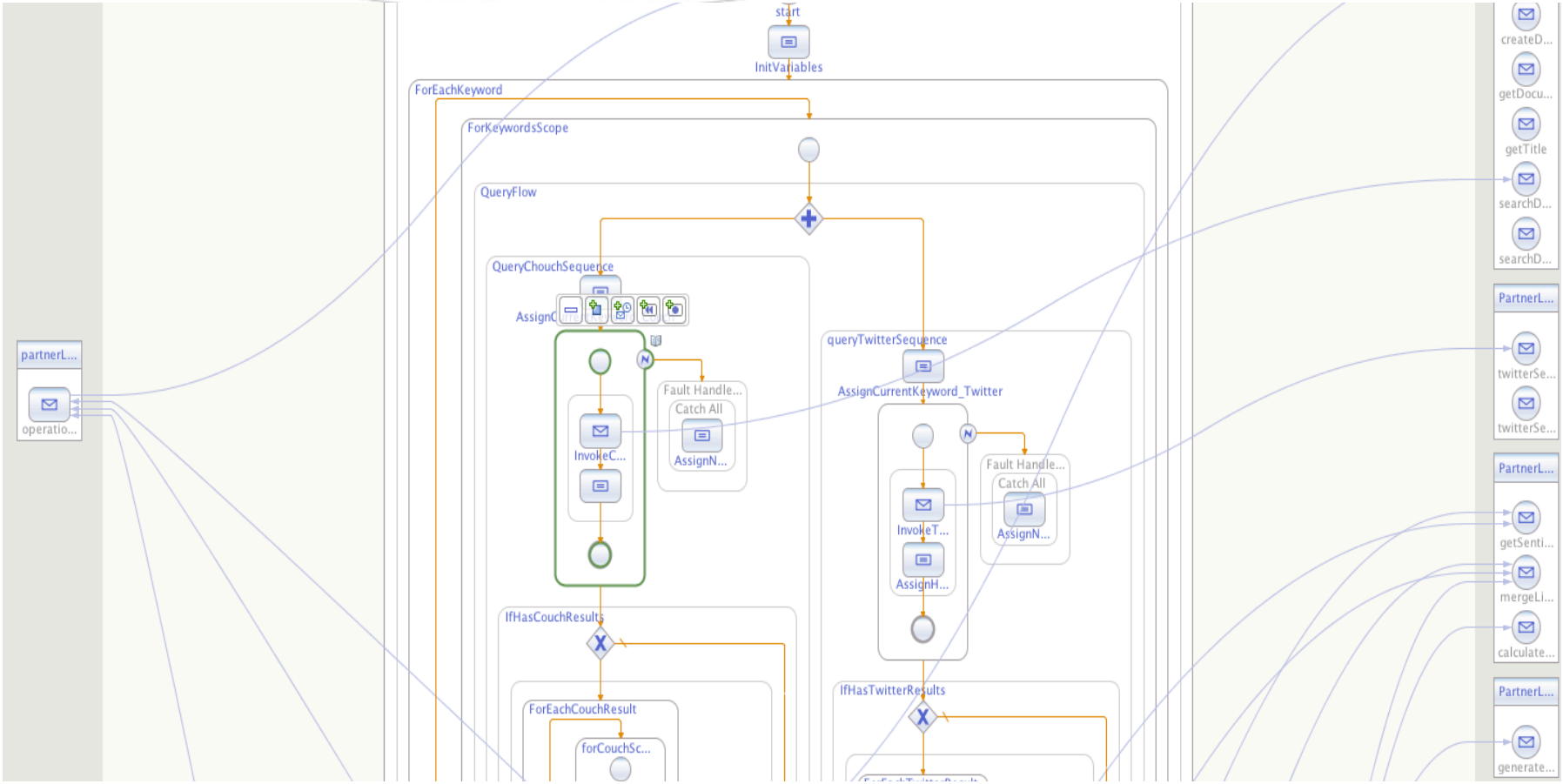
BPEL - Problems

- Kind of dead, though some might argue
  - Oracle 11g released before 2009
  - Apache ODE still alive (Oct. 2013)
    - Just BPEL Engine, not an IDE (see Eclipse BPEL)
    - Only Bugfixes and Performance improvements?
  - Last Open Source Updates:
    Netbeans SOA 2008, (jOpera 2011),
    Eclipse BPEL Designer 2011

BPEL - Problems

- Poor Support for Parallelism
  - No Synchronization Constructs (Locks, etc)
  - <For>-Loop: parallel=true not supported by Netbeans/SUN BPEL Engine
  - Even Oracle seems to have (had?) problems
    - No true parallelism, just preemptive multitasking
    - Web Service calls still block the Process
    - Even Wait blocks
    - http://technology.amis.nl/2008/10/23/

Our composition



Demo tool

Q&A

?