

## Semantics of interoperable and outsourced information systems

H. Balsters G.B. Huitema

University of Groningen

### *Objectives*

*Establishing correctness criteria for service delegation: How do we validate that the service has eventually been outsourced correctly?*

**(Semantics of service delegation)**

### *Context*

- Businesses can change their business structure by unbundling drastically into separate companies or, on the other end of the spectrum, by smoothly delegating (or outsourcing) some of their business processes to other more specialized parties.
- We concentrate on conceptual modeling and semantics of outsourcing in information systems.
- Outsourcing in the context of information systems will be defined as delegating a part of the functionality of the original system to an existing outside party (the supplier).
- Such functionality typically involves one or more operations (or services), where each operation satisfies certain input- and output requirements.

### *Methods (1)*

- Outsourcing requirements will be defined in terms of the ruling service level agreements (*SLAs*).
- We provide a formal means to enabling a *check* that the outsourcing relationship between outsourcing party and supplier, determined by a SLA, satisfies specific correctness criteria.
- These correctness criteria are defined in terms of *consistency* and *completeness* between the operation to be outsourced and the operation offered by the supplier.
- Our correctness criterion will concern mappings between an outsourcer schema and a supplier schema, and will address both semantical and ontological aspects pertaining to outsourcing.

### *Methods(2)*

- Our analysis is performed within the modeling framework based on the UML/OCL formalism
- In particular we offer detailed specifications of so-called *exact views* on the information systems of both parties engaged in the outsourcing process.
- These exact views will capture the correctness criteria for outsourcing of operations.

### *Limitations*

*Virtual* outsourcing:outsourcer usually cannot give up ownership of data, due to existence of other services than just the delegated one using same data

### *Results*

A method for modelling requirements for IS-outsourcing. Our approach offers formal specifications that can prove their value in the setup and evaluation of outsourcing contracts.

**Conclusions**

- We have to conceive mappings between outsourcer and supplier: get common format for data and operations (*data and operation extraction/exchange*)
- We have to align constraints on outsourced service between outsourcer and supplier (match pre- & post-conditions)
- Successful alignment constitutes basis for successful validation of outsourcing

**Implications**

- Matching initiates negotiation process between outsourcer and supplier (alignment by weakening/strengthening of constraints)
- Alignment process can guide setup and evaluation of outsourcing contracts

**METHODS** (delegating IS-functionality)

**Outsourcer:**

- Isolate operation *O* in source company to be delegated
- Determine relevant attributes, relations, constraints and auxiliary operations in definition of operation *O*
- This material represents *outsourcer view* of *O*

**Supplier:**

- Offer *supplier view* with an operation *O'* having a pre-condition that always rejects objects that are rejected by *O*
- Post-condition of *O'* should never offer results contradicting post-condition of *O*

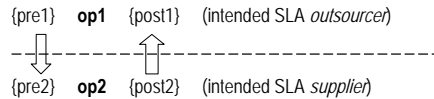
**Alignment:**

Typically, an outsourcer (*Source*) is roaming for a supplier (*Target*) with compliant pre/post-conditions

**Abstract implementation principle :**

constructing an  $\omega$ -schema

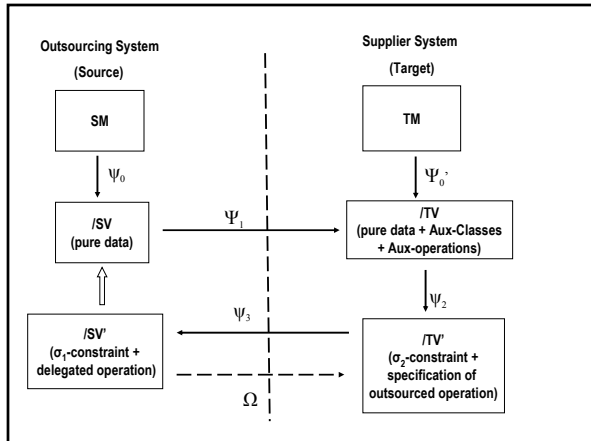
(op2 implements op1):



Negotiation means negotiation on pre- and post-conditions: i.e. on *strengthening/weakening*

*Views are used as virtual representatives of parts of existing systems*

- We construct a view on top of the corresponding IS.
- This view will be **exact**: this means that it always has a *unique inverse*.
- This way we cannot only query, but also update through this view (Abiteboul/Vianu-95, Balsters-04). Updates can be implemented using so-called IN STEAD OF- triggers, in the case of relational databases.
- Views in UML/OCL can be depicted by derived classes (Warmer/Kleppe-03, Balsters-03).
- Exact views are constructed through  $\Psi$ -maps ( $\Psi$  stands for *preservation of system integrity*; Balsters-04).

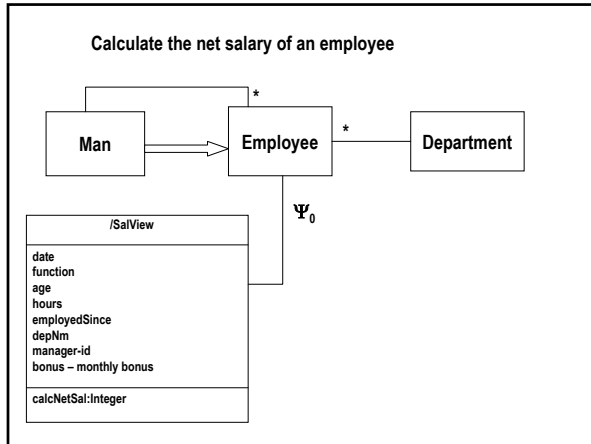


**The assignment**

Calculate the net salary of a given set of employees on the basis of

- The present date
- The employee's function
- The employee's age
- The number of hours an employee works
- The initial date of employment
- The employee's department
- The employee's monthly bonus

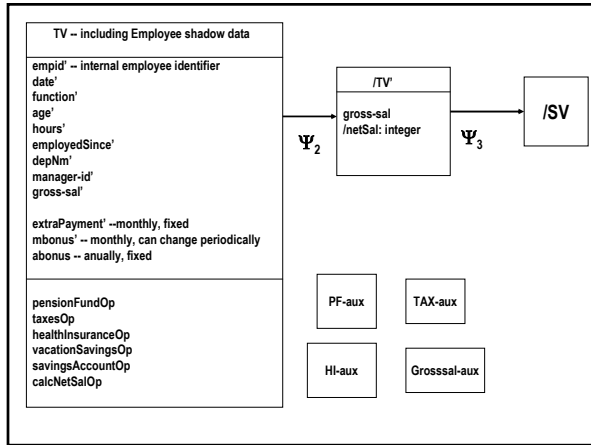
Also give a report describing all amounts deducted from the employee's gross salary



**Contract specifications (minimal SLA):**

```

context /SalView::CalcNetSal(date:Date): Integer
pre: --none
post: age>40 implies result>3000 and
      result < (self.manager).date.calcNetSal and
      depNm='toy' implies result<4000
  
```



**Existence of a  $\psi$ -map like  $\psi_3$  is necessary for 3 reasons:**

- $\psi_3$  maps object data from TV' to SV' solving *data extraction problems* (homonyms, synonyms, conversions etc.) in the transition from the target system to the source model.
- $\psi_3$  maps object data from TV' to SV' solving *data reconciliation problems* (constraint resolution) in the transition from the target system to the source model.
- We have to ensure that each object in the set of instances of the target system corresponds to exactly one (combination of) object(s) in the source system, and vice versa.

(The same holds for each of the  $\psi$ -maps involved. Only in this way can we freely, and unambiguously move between the source and the target systems.)

**How does negotiation work in this process of constructing  $\omega$ -schemas?**

Let's look again at the Abstract Implementation Principle (constructing an  $\omega$ -schema)

**(op2 implements op1):**

```

{pre1} op1 {post1} (intended SLA outsourcer)
-----
{pre2} op2 {post2} (intended SLA supplier)
  
```

**Negotiation** is directed at *strengthening* and/or *weakening* of pre- and post-conditions

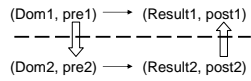
- "Okay, we would like to cater to all of your employees, but we can only deal with your administrative personnel"
- "We can't check all of your constraints pertaining to salary specification, but only some of them"

**We now have a choice:**

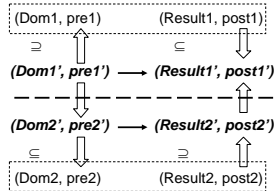
- The **outsourcer** can consider *weakening* his demands:
  - Pre1' will state that we only deal with administrative personnel
  - Post1' will state that only a subset of the constraints are guaranteed
- The **supplier** can consider *strengthening* his facilities:
  - Pre2' will state that a larger category can be dealt with than only administrative personnel
  - Post2' will include a larger category of constraint checks than those found in Post2

*Consider the case that the supplier fully wins the negotiation, forcing the outsourcer to weaken his demands ...*

**Before negotiation:**



**After negotiation (strengthening the pre- and weakening the post condition):**



Typically, domains will shrink and result constraints get weaker

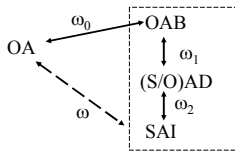
**Why does formalizing the negotiation process help?**

Formalization:

- Helps to understand
- Facilitates analysis (consistency, completeness)
- Paves the way to tool support
- Beneficial for checking  $\omega$ -schemas (since this is non-trivial)

**There is a whole suite of outsourcing agents!**

1. **Business SLA:** outsourcing agent (OA), supplier agent (SA)
  - technical (our example)
  - non-technical: costs, delivery time, penalties
2. **Design SLA:** OA-business (OAB = SA), SA-design (SAD)
3. **Implementation SLA:** OA-design (OAD), SA-implementation (SAI)



Each  $\omega$  abides to the abstract implementation principle!

*Synergy between the agents translates to composition of the  $\omega$ -schemas on the three different agent levels!*

**Research issues**

1. The relation between design and implementation in IS-context is already well-documented and can be considered as more or less traditional
2. Our research up till now has its focus on the relation between (external) business and (external) supplier, and also between (internal) business and (internal) design – albeit that we have confined ourselves to the *technical* issues pertaining to a SLA). New research could be directed at non-technical issues, and especially in the  $\omega$ -relation between (external) business and (external) supplier
3. Possibly UML/OCL is sufficiently equipped to perform the job, but maybe we need something more expressive (like ORM+) to specify business needs (specified as business rules), and also to validate correctness of these business rules (did the information analyst correctly understand the rules he obtained from the domain expert?)