

A Framework for Integrating Business Processes and Business Requirements

Raman Kazhamiakin, Marco Pistore, Marco Roveri

{raman,pistore}@dit.unitn.it roveri@itc.it

DIT, University of Trento, Italy ITC-irst, Trento, Italy



• motivation

- business requirements modelling
- from business requirements to business processes
- analysis support
- conclusions and future work



- New challenges
 - IT growth and internet development remove bounds on the enterprises and customers collaborations
 - Organization operates in heterogeneous, competing and changing environment
 - Autonomy and flexibility of partners participating in cross-enterprise business processes



- New challenges
 - IT growth and internet development remove bounds on the enterprises and customers collaborations
 - Organization operates in heterogeneous, competing and changing environment
 - Autonomy and flexibility of partners participating in cross-enterprise business processes
- Business Process Management in a broader sense
 - universal interoperability between applications
 - resolution of conflicts and changes in business strategies
 - reduce costs of integration and adaptation
 - (CSC) Success in understanding and managing business processes can mean the difference between keeping and loosing your company

- SOA pros:
 - enables definition of coarse-grained loosely-coupled services
 - supplies interoperable solution to application integration
 - facilitates the integration of applications across enterprise boundaries
 - enables high level of automation to the solution delivery process

- SOA pros:
 - enables definition of coarse-grained loosely-coupled services
 - supplies interoperable solution to application integration
 - facilitates the integration of applications across enterprise boundaries
 - enables high level of automation to the solution delivery process
- SOA cons:

- SOA pros:
 - enables definition of coarse-grained loosely-coupled services
 - supplies interoperable solution to application integration
 - facilitates the integration of applications across enterprise boundaries
 - enables high level of automation to the solution delivery process
- SOA cons:
 - decentralized society of autonomous and changing actors
 - no control over partners services and processes
 - changes are autonomous, frequent, unpredictable

- SOA pros:
 - enables definition of coarse-grained loosely-coupled services
 - supplies interoperable solution to application integration
 - facilitates the integration of applications across enterprise boundaries
 - enables high level of automation to the solution delivery process
- SOA cons:
 - decentralized society of autonomous and changing actors
 - lack of support for "strategic" descriptions of business models
 - different participants act on behalf of their own strategies and requirements
 - their requirements and expectations are often in conflict
 - changes in strategies should be aligned with the business process models

- SOA pros:
 - enables definition of coarse-grained loosely-coupled services
 - supplies interoperable solution to application integration
 - facilitates the integration of applications across enterprise boundaries
 - enables high level of automation to the solution delivery process
- SOA cons:
 - decentralized society of autonomous and changing actors
 - lack of support for "strategic" descriptions of business models
 - lack of analysis techniques supporting negotiation in collaborations and their modifications
 - correctness of the process composition
 - analysis of processes with respect to specific behavioral properties

astro Proposed Framework

- - Requirements modelling language
 - to incarnate motivations and intentions behind a business process models
 - to represent "negotiation" aspects of collaboration
 - Integration of business requirements and business processes
 - to visualise the implication of business strategies changes in the underlying processes and their compositions
 - Formal analysis techniques
 - to increase the reliability of the models
 - to support the resolution of conflicts during the negotiation
 - to verify the conformance of the business processes with respect to the strategic descriptions

Basing on Tropos language (from Greek trope: easily adaptable).

- Tropos is requirements-driven:
 - focus on early phases of requirements analysis, aiming to the understanding of the operational environment of the software system.

Basing on Tropos language (from Greek *trope*: easily adaptable).

- Tropos is requirements-driven:
 - focus on early phases of requirements analysis, aiming to the understanding of the operational environment of the software system.
- Tropos is agent-oriented:
 - agents and related notions, such as goals and plans, are used in all phases of software development.

Basing on Tropos language (from Greek *trope*: easily adaptable).

- Tropos is requirements-driven:
 - focus on early phases of requirements analysis, aiming to the understanding of the operational environment of the software system.
- Tropos is agent-oriented:
 - agents and related notions, such as goals and plans, are used in all phases of software development.
- Tropos has been applied in several case studies on **information systems** and **agent-based software systems**.

Basing on Tropos language (from Greek *trope*: easily adaptable).

- Tropos is requirements-driven:
 - focus on early phases of requirements analysis, aiming to the understanding of the operational environment of the software system.
- Tropos is agent-oriented:
 - agents and related notions, such as goals and plans, are used in all phases of software development.
- Tropos has been applied in several case studies on **information systems** and **agent-based software systems**.
- Tropos offers a set of **graphical notations** and of analysis techniques to support the designer in the development of the software system.

Basing on Tropos language (from Greek *trope*: easily adaptable).

- Tropos is requirements-driven:
 - focus on early phases of requirements analysis, aiming to the understanding of the operational environment of the software system.
- Tropos is agent-oriented:
 - agents and related notions, such as goals and plans, are used in all phases of software development.
- Tropos has been applied in several case studies on **information systems** and **agent-based software systems**.
- Tropos offers a set of **graphical notations** and of analysis techniques to support the designer in the development of the software system.
- Formal Tropos extends Tropos with a formal specification language and with verification based on Model Checking.

astro Case Study

- General domain: Public Administration
- Specific domain: Environmental Protection Agency
 - Authorization for the establishment and operation of a waste disposal or recycling plant.
 - A citizen (factory) submits an application to obtain the license for its waste disposal or recycling plant (incinerator, recycling facility, private landfill,...).
 - The **local government**, involving various agencies and experts, evaluates the proposal and authorizes the plant if it complies with high standards of environmental protection (norms and laws).
- Involves many heterogeneous, distributed and autonomous actors
- Takes into account global requirements for the composition and (probably conflicting) local requirements of different actors

astro Business Requirements: Case Study



EDOC'04 - 22.09.2004 - p. 8

a structure of Distributed Rusiness Processes Business Requirements: Case Study



.

a stro Business Requirements: Case Study

Protocol Provide Office Registration Register Information Incoming Applications Public Conference Activate Manage Application Applications Waste Management Management Office Submit Application Provide Efficiency Progress Information Track Provide Application Documents Progress Get Waste Citizen License Responsible Transparent Participation Application Management

EDOC'04 – 22.09.2004 – p. 8

.

..........

a strong Business Requirements: Case Study



EDOC'04 - 22.09.2004 - p. 8

a strong Business Requirements: Case Study



EDOC'04 - 22.09.2004 - p. 8

astro Business Requirements: Case Study

Representation of requirements in collaboration



EDOC'04 - 22.09.2004 - p. 8

astro Business Requirements: Refinement

Waste Management Office Manage Applications

astro Business Requirements: Refinement

Waste Management Office Manage Applications Complete Valid Manage Efficiency Transparency Applications Application Documents

EDOC'04 - 22.09.2004 - p. 9

astro Business Requirements: Refinement



astro Business Requirements: Refinement

Waste Management Office Manage Applications Complete Manage Valid Efficiency Transparency Applications Application Documents / + Provide Satisfy Time Valid Minimize Public Correct Valid Progress Management Conference App. Docs Termination Reports Limits Information Time Process Application Initialize Analyze Application Manage Docs Organize Conference Validate Validate Technical Publish Provide Obtain Call Activate Documents $\neg \supset$ Reports -⊳ Conference \rightarrow Recommended Registration H Partners Application Obtain Protocol Decision Information Obtain Technical Documents Reports

astro Business Requirements: Refinement

Waste Management Office Manage Applications Complete Manage Valid Efficiency Fransparency Applications Application Documents + Provide Valid Satisfy Time Minimize Public Correct Valid Progress Management Conference App. Docs Reports Termination Limits Information Time Process Application Initialize Analyze Application Manage Docs Organize Conference Validate Validate Technical Provide Publish Obtain Call Activate Documents Reports Conference - Recommended ∽ ₩Þ Registration Partners Application Obtain Protocol Decision Information Obtain Technical Documents Reports Integration Registration Additional Technical Conference Appointment Recommendations Request Reports Information Documents Call Technical Technical Province Protocol Citizen Citizen Commission Commission Board Office

EDOC'04 – 22.09.2004 – p. 9

astro Business Requirements: Refinement

Representation of local requirements



EDOC'04 – 22.09.2004 – p. 9

astro Business Requirements: Formal Properties



EDOC'04 - 22.09.2004 - p. 10

astro Formal Tropos

- classes representing actors, goals, activities and dependencies
- first-order linear-time temporal constraints on the evolutions of the model
- focus on **creation** and **fulfillment** of activities



Task Initialize mode achieve Attribute docs: Documents

- Fulfillment trigger
 - The initialization task completes with the
 - application activation
 - \exists aa: ActivateApp (aa.**super = self** \land **Fulfilled**(aa))

Fulfillment condition

- when the initialization task completes,
- the documentation should be valid

docs.valid

as tro Integrating Requirements and Processes



EDOC'04 – 22.09.2004 – p. 12

S t r o Formal Verification

- Verification of business requirements
 - consistency checks: "the specification admits valid scenarios";
 - possibility checks: "there is *some* scenario that respects possibility property":
 in: Initialize (Fulfilled(in))
 - assertion validation: "*all* scenarios respect assertion property":
 - \forall ri: RegInfo (\forall wmo: WMO (ri.receiver = wmo \wedge ri.docs.valid \rightarrow
 - $F \exists in: Initialize (in.actor = wmo \land in.docs = ri.docs \land Fulfilled(in))))$

S t r o Formal Verification

- Verification of business requirements
 - consistency checks: "the specification admits valid scenarios";
 - possibility checks: "there is *some* scenario that respects possibility property":
 ∃ in: Initialize (Fulfilled(in))
 - assertion validation: "*all* scenarios respect assertion property":

 \forall ri: RegInfo (\forall wmo: WMO (ri.receiver = wmo \land ri.docs.valid \rightarrow

 $F \exists in: Initialize (in.actor = wmo \land in.docs = ri.docs \land Fulfilled(in))))$

• Process verifications

• Deadlocks and livelocks freedom verifications

astro Formal Verification

- Verification of business requirements
 - consistency checks: "the specification admits valid scenarios";
 - possibility checks: "there is *some* scenario that respects possibility property":
 ∃ in: Initialize (Fulfilled(in))
 - assertion validation: "*all* scenarios respect assertion property":

 \forall ri: RegInfo (\forall wmo: WMO (ri.receiver = wmo \land ri.docs.valid \rightarrow

 $\mathsf{F} \exists in: Initialize (in.actor = wmo \land in.docs = ri.docs \land \textbf{Fulfilled}(in))))$

• Process verifications

- Deadlocks and livelocks freedom verifications
- Verification of process against requirements models
 - verify on the refined model all possibilities and assertions of the formal requirements model;
 - verify whether the refined model satisfies the requirements specified in the **Creation**, **Invariant** and **Fulfillment** constraints;
 - verify whether the composition of processes satisfies above properties.

astro Conclusions

Conclusions...

- A methodology for business requirements modelling
 - based on (extension of) Tropos modelling language
 - starting from strategic goals and constraints
 - refining business requirements into business processes

astro Conclusions

Conclusions...

- A methodology for business requirements modelling
 - based on (extension of) Tropos modelling language
 - starting from strategic goals and constraints
 - refining business requirements into business processes
- Integration with Web service process definitions (e.g. in BPEL4WS)
 - extraction of definitions for ports, messages, partners and process skeletons
 - explicit relations of tasks with the Web service process definitions
 - analysis of specifications on more detailed level

astro Extracting Web Service Processes



```
supporting Composition of Distributed Business Processes
```

astro Extracting Web Service Processes

<sequence name="Initialize"> RECEIVE <receive name="receive reg info" registration info operation="manageApp" variable="vAppRequest"/> <assign> <copy> ASSIGN <from variable="vAppRequest " query="/docs/complete"/> valid = docs.complete <to variable="valid"/> </copy> </assign> WHILE < while condition= ! valid "getVariableData('valid') == false() "> <invoke name="request documents" operation="docRequest" inputVariable="vDocRequest"/> **INVOKE** request docs <receive name="receive new docs" operation="docResponse" variable="vDocResponse"/> <assign> RECEIVE <copy> new docs <from variable="vDocResponse" query="/docs/complete"/> <to variable="valid"/> </copy> **ASSIGN** </assign> update valid </while> <empty name="activate application"/> **EMPTY** </sequence> activate app \mathbf{V}

astro Conclusions

Conclusions...

- A methodology for business requirements modelling
 - based on (extension of) Tropos modelling language
 - starting from strategic goals and constraints
 - refining business requirements into business processes
- Integration with Web service process definitions (e.g. in BPEL4WS)
 - extraction of definitions for ports, messages, partners and process skeletons
 - explicit relations of tasks with the Web service process definitions
 - analysis of specifications on more detailed level
- Support for analysis techniques
 - consistency of requirements
 - correctness of processes
 - correspondence between processes and strategic goals and constraints

astro Formal analysis: T-Tool

Tool Counter **SPIN** $\underset{\scriptscriptstyle Q}{\text{example}}_{\scriptscriptstyle Q}$ ₽ Citizen ntermediate **IL2SPIN** New Docu Doc Request **BPEL Process Verification Engine NuSMV** _anguage **IL2SMV FT Model** Being Assisted Citizen Quality Service Receive Service **Verification Engine** Pay

astro Formal Verification: Examples

Deadlock: WMO re-requests documents but the Citizen does not respond

Livelock: WMO re-requests documents repeatedly and the Citizen sends incomplete docs infinitely

Processes against requirements:

when initialisation completes the documents should be valid \forall in: Initialize (**Fulfilled**(in) \rightarrow in.docs.valid) *missed assignment*



astro Future Works

Future works...

- Complete intermediate language for better capturing the needs of the business domain
 - better focus on activity level description
 - better integration of processes with requirements models
- Experiment with alternative verification techniques and tools
- Improve BPEL code extraction and generation
- Integration with the planning techniques for the process synthesis to enable adaptation of processes to changes in requirements



Thank You!