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
Outline

- 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
Service-oriented Architecture

SOA and Web services infrastructure create an environment for interconnecting organisations and applications

- **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
Service-oriented Architecture

SOA and Web services infrastructure create an environment for interconnecting organisations and applications

- **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:**
Service-oriented Architecture

SOA and Web services infrastructure create an environment for interconnecting organisations and applications

- **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
Service-oriented Architecture

SOA and Web services infrastructure create an environment for interconnecting organisations and applications

- **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
Service-oriented Architecture

SOA and Web services infrastructure create an environment for interconnecting organisations and applications

- **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
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
Language for Business Requirements

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.
Language for Business Requirements

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.
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
Business Requirements: Case Study

Protocol Office
- Register Incoming Applications
- Activate Application Management
- Submit Application
- Track Application Progress
- Get Waste License
- Responsible Participation
  - Transparent Application Management

Citizen
Business Requirements: Case Study

Protocol Office
- Register Incoming Applications
- Activate Application Management
- Submit Application
- Track Application Progress
- Get Waste License
- Responsible Participation

Citizen
- Transparent Application Management
- Provide Registration Information
- Provide Progress Information
- Provide Documents

Waste Management Office
- Manage Applications
- Public Conference
- Provide Progress Information
- Provide Documents

Efficiency
Business Requirements: Case Study

- Register Incoming Applications
- Activate Application Management
- Submit Application
- Track Application Progress
- Get Waste License
- Responsible Participation
- Transparent Application Management
- Provide Registration Information
- Manage Applications
- Provide Progress Information
- Provide Documents
- Efficiency
- Public Conference
- Waste Management Office
- Conference Announcement
- Analyze Application
- Expert Analysis
- Technical Commission
- Protocol Office
- Citizen
Business Requirements: Case Study

- Register Incoming Applications
- Activate Application Management
- Submit Application
- Track Application Progress
- Get Waste License
- Responsible Participation
- Transparent Application Management
- Provide Registration Information
- Provide Progress Information
- Provide Documents
- Provide Final Decision
- Provide Recommended Decision
- Provide Decision
- Fair Management
- Efficiency
- Management Office
- Waste Management Office
- Conference Announcement
- Analyze Application
- Expert Analysis
- Technical Commission
- Office
- Protocol Office
- Office
- Citizen
- Province Board
- EDOC’04 – 22.09.2004 – p. 8
Business Requirements: Case Study

Representation of requirements in collaboration

Diagram:
- **Citizen**
  - Get Waste License
  - Responsible Participation
  - Transparent Application Management
  - Track Application Progress
- **Protocol Office**
  - Register Incoming Applications
  - Activate Application Management
- **Provide Registration Information**
- **Provide Application Management**
- **Submit Application**
- **Provide Progress Information**
  - Efficiency
  - Fair Management
- **Manage Applications**
  - Conference Announcement
- **Public Conference**
  - Analysis Application
- **Provide Documents**
- **Provide Final Decision**
- **Technical Commission**
  - Expert Analysis
  - Province Board
  - Decision Ratification
  - Efficiency
  - Management
  - Provide Recommended Decision
Business Requirements: Refinement

Manage Applications

Waste Management Office
Business Requirements: Refinement
Business Requirements: Refinement

- Manage Applications
  - Manage Application
  - Valid Documents
    - Valid App. Docs
    - Valid Reports
  - Complete Applications
    - Correct Termination
    - Satisfy Time Limits
  - Efficiency
    - Minimize Management Time
  - Transparency
    - Public Conference
    - Provide Progress Information

- Waste Management Office

EDOC’04 – 22.09.2004 – p. 9
Business Requirements: Refinement

- Manage Application
  - Valid Documents
  - Complete Applications
- Organize Conference
  - Obtain Registration Information
  - Manage Docs
  - Activate Application
  - Validate Documents
  - Obtain Documents
  - Analyze Application
    - Validate Technical Reports
    - Obtain Technical Reports
  - Call Partners
  - Publish Conference Protocol
  - Provide Recommended Decision

Waste Management Office

Manage Applications

Valid App. Docs
Valid Reports
Correct Termination
Satisfy Time Limits
Minimize Management Time
Public Conference
Provide Progress Information

Efficiency
Transparency

EDOC’04 – 22.09.2004 – p. 9
Business Requirements: Refinement
Business Requirements: Refinement

Representation of local requirements
Business Requirements: Formal Properties

Initialize task is successful if application procedure is started.

Manage Docs task is completed if the documents are valid.

Obtain Registration Information task completes with the reception of message.

When the initialisation is completed the documents should be valid.
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
- \( \exists \text{aa}: \text{ActivateApp} (\text{aa.\texttt{super}} = \text{self} \land \text{Fulfilled}(\text{aa})) \)

**Fulfillment condition**
- when the initialization task completes,
- the documentation should be valid
  docs.valid
Integrating Requirements and Processes

Local view

Collaboration view

Business Requirements

Business Processes

Formal Verification

- Verification of business requirements
  - consistency checks: “the specification admits valid scenarios”;
  - possibility checks: “there is some scenario that respects possibility property”:
    \[ \exists \text{ in: } \text{Initialize} \left( \text{Fulfilled}(\text{in}) \right) \]
  - assertion validation: “all scenarios respect assertion property”:
    \[ \forall \text{ ri: } \text{RegInfo} \left( \forall \text{ wmo: } \text{WMO} \left( \text{ri.receiver = wmo} \land \text{ri.docs.valid} \rightarrow \right) \right) \]
    \[ \text{F } \exists \text{ in: } \text{Initialize} \left( \text{in.actor = wmo} \land \text{in.docs = ri.docs} \land \text{Fulfilled}(\text{in})) \right) \]
Formal Verification

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

- **Process verifications**
  - Deadlocks and livelocks freedom verifications
Formal Verification

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

- 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.
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
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
Extracting Web Service Processes

Initialization

Obtain Registration Information

Manage Applications

Valid Documents

Obtain Documents

Validate Documents

Complete Applications

Correct Termination

Satisfy Time Limits

Validate Application

Receive registration info

Assign valid = docs.complete

While ! valid

Invoke request docs

Receive new docs

Assign update valid

Empty activate app

Manage Application

Valid Documents

Complete Applications

Valid App. Docs

Valid Reports

Receive new docs

Assign update valid

Empty activate app

Manage Application

Valid Documents

Complete Applications

Valid App. Docs

Valid Reports

Receive registration info

Assign valid = docs.complete

While ! valid

Invoke request docs

Receive new docs

Assign update valid

Empty activate app
<sequence name="Initialize">
  <receive name="receive reg info" operation="manageApp" variable="vAppRequest"/>
  <assign>
    <copy>
      <from variable="vAppRequest" query="/docs/complete"/>  
      <to variable="valid"/>
    </copy>
  </assign>
  <while condition="getVariableData('valid')==false()">
    <invoke name="request documents" operation="docRequest" inputVariable="vDocRequest"/>
    <receive name="receive new docs" operation="docResponse" variable="vDocResponse"/>
    <assign>
      <copy>
        <from variable="vDocResponse" query="/docs/complete"/>  
        <to variable="valid"/>
      </copy>
    </assign>
  </while>
  <empty name="activate application"/>
</sequence>
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
Formal analysis: T-Tool

TTool

Intermediate Language

Counter example

BPEL Process

FT Model

SPIN
Verification Engine

IL2SPIN

NuSMV
Verification Engine

IL2SMV

Counter example

Assign waitResponse = false

Invoke InitialRequest

While

Pick WaitMessage

OnMessage Response

OnMessage InfoRequest

Assign waitResponse = false

Assign result = message.result

Reply Info

WaitResponse !

WaitResponse

BPEL
Process

FT Model

Citizen Being Assisted

Receive Service

Pay

Quality Service

PO

Registration Info

Doc Request

New Documents

Intermediate Language

SPIN
Verification Engine

IL2SPIN

NuSMV
Verification Engine

IL2SMV
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*

∀ in: Initialize ($\text{Fulfilled}(\text{in}) \rightarrow \text{in.docs.valid}$)

*missed assignment*
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!