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.

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.

- Tropos has been applied in several case studies on **information systems** and **agent-based software systems**.

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.
- 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.

Language for Business Requirements

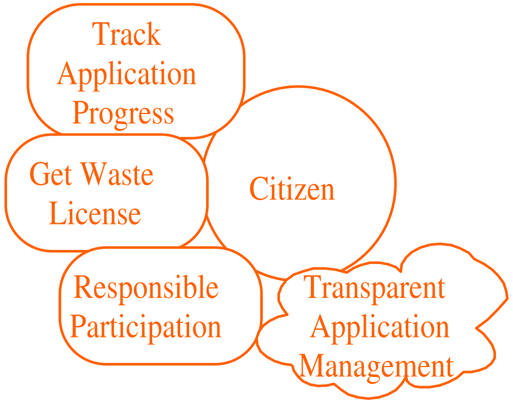
Basing on Tropos language (from Greek *trope*: easily adaptable).

- 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.
- 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.

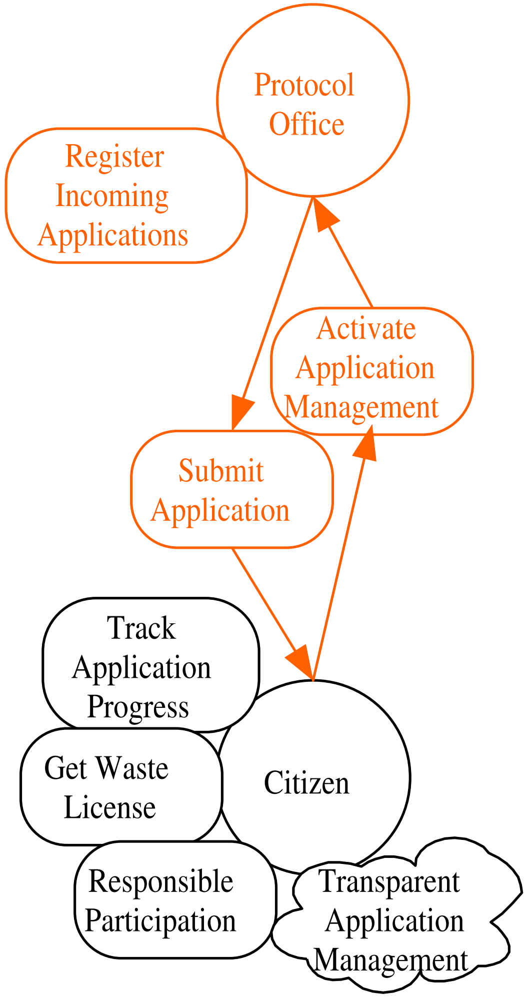
Case Study

- 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

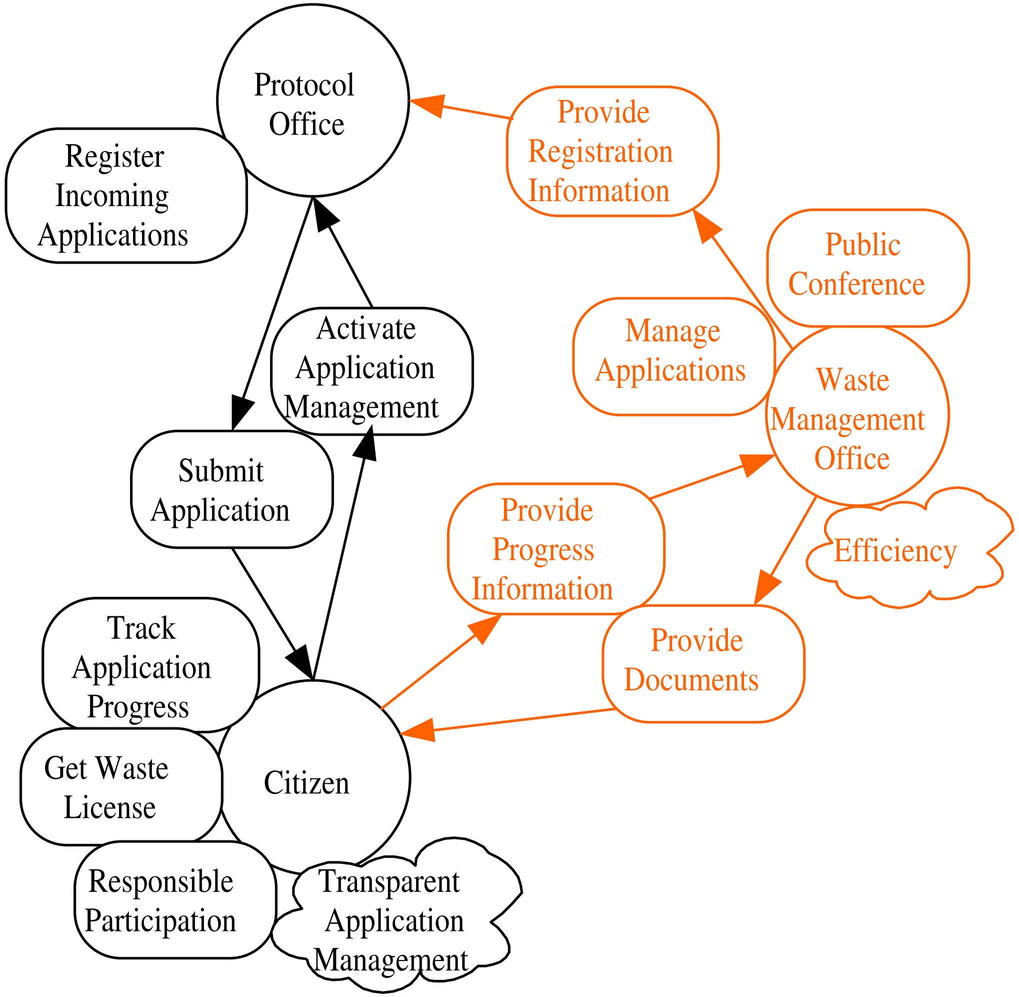
Business Requirements: Case Study


Track Application Progress
Get Waste License
Citizen
Responsible Participation
Transparent Application Management

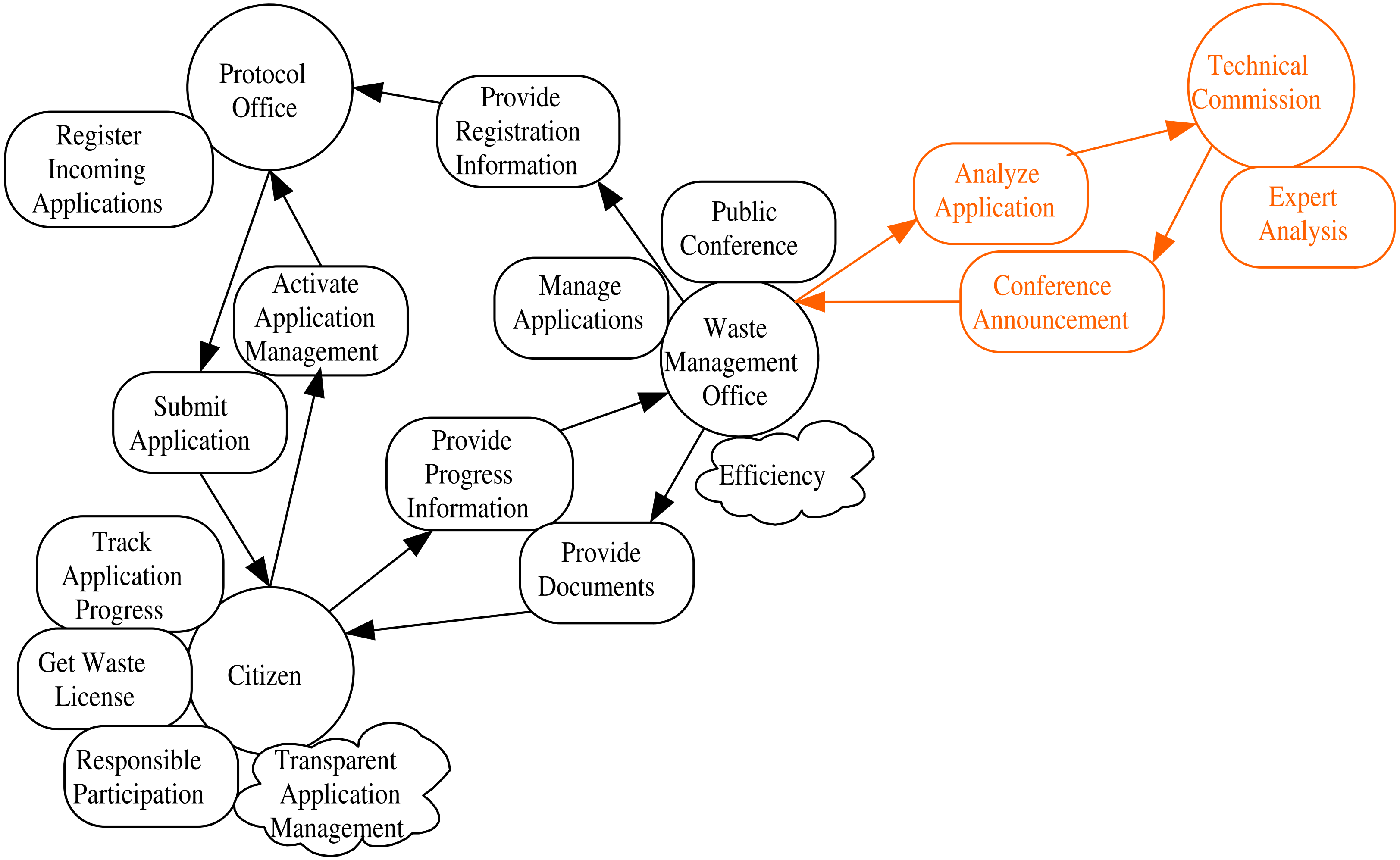
Business Requirements: Case Study


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

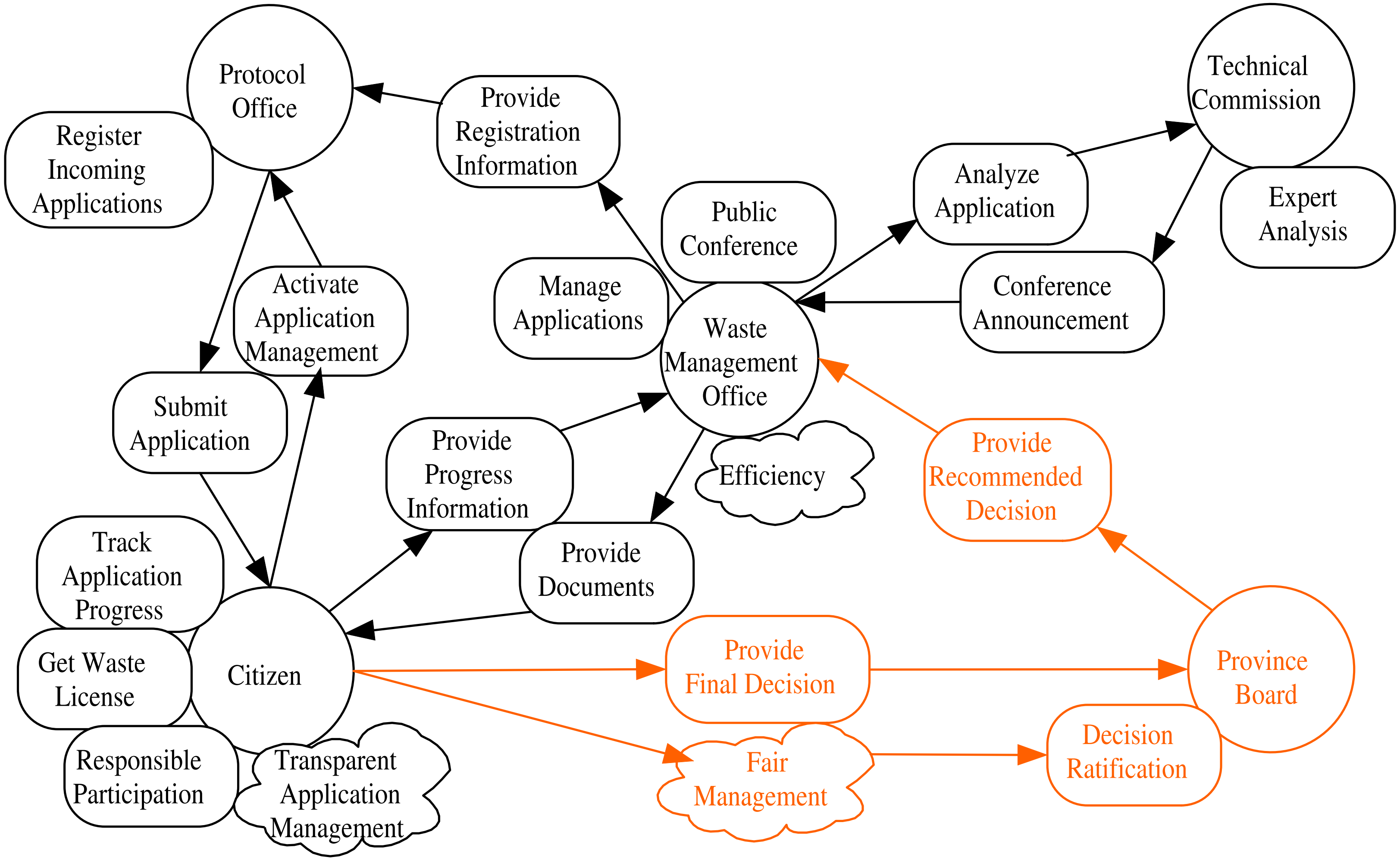
Business Requirements: Case Study


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

Business Requirements: Case Study

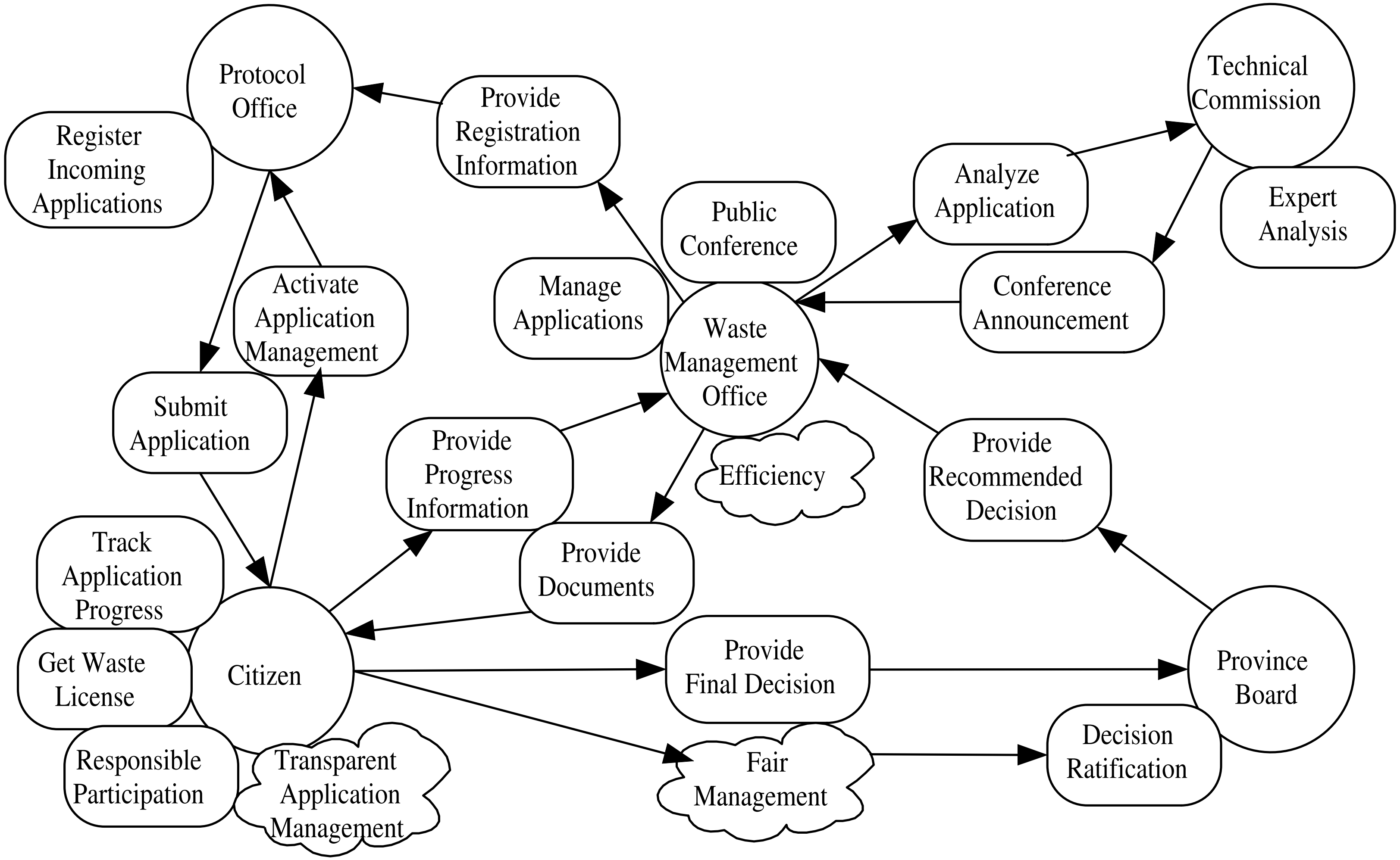


Business Requirements: Case Study

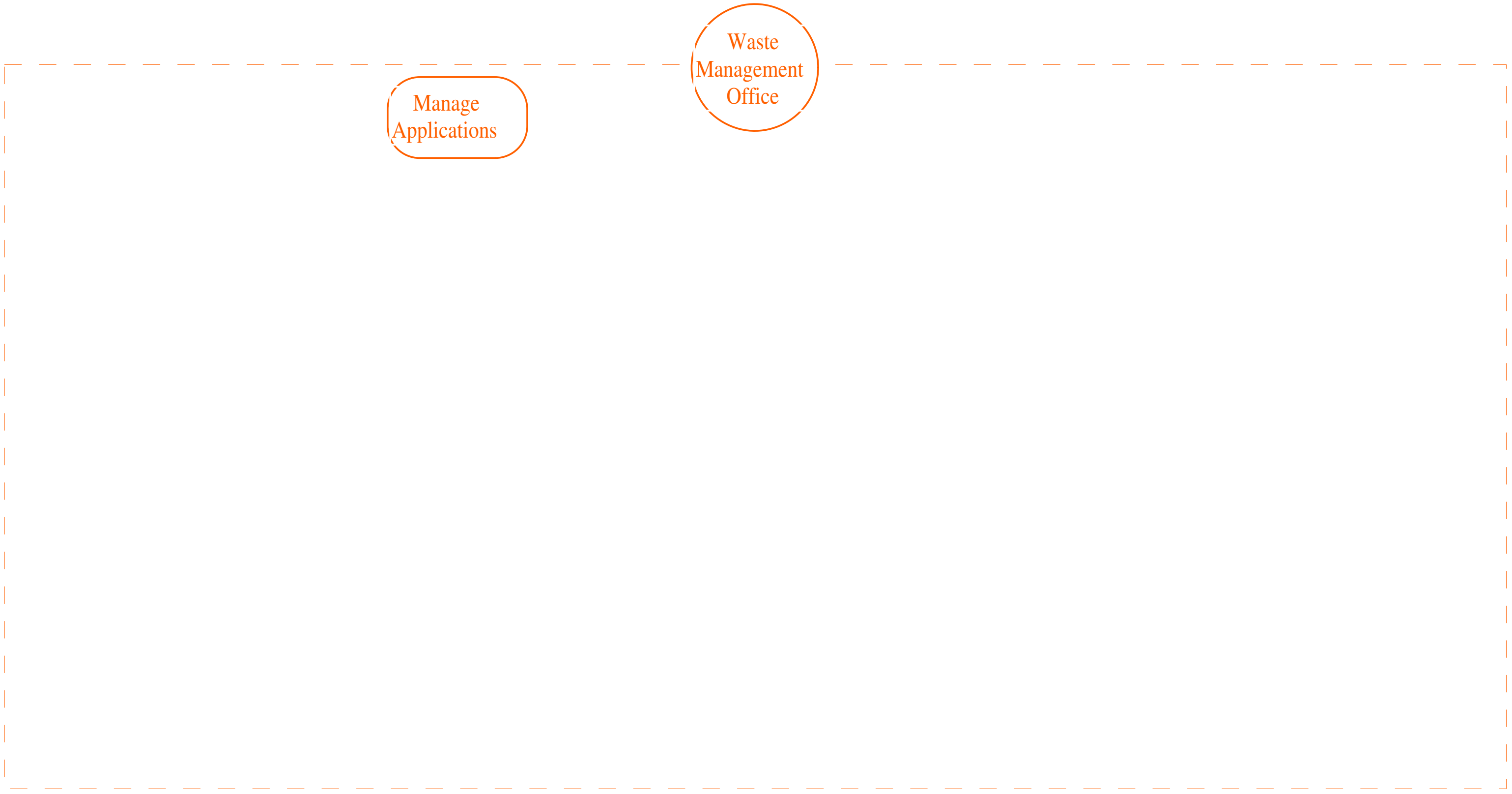


Business Requirements: Case Study

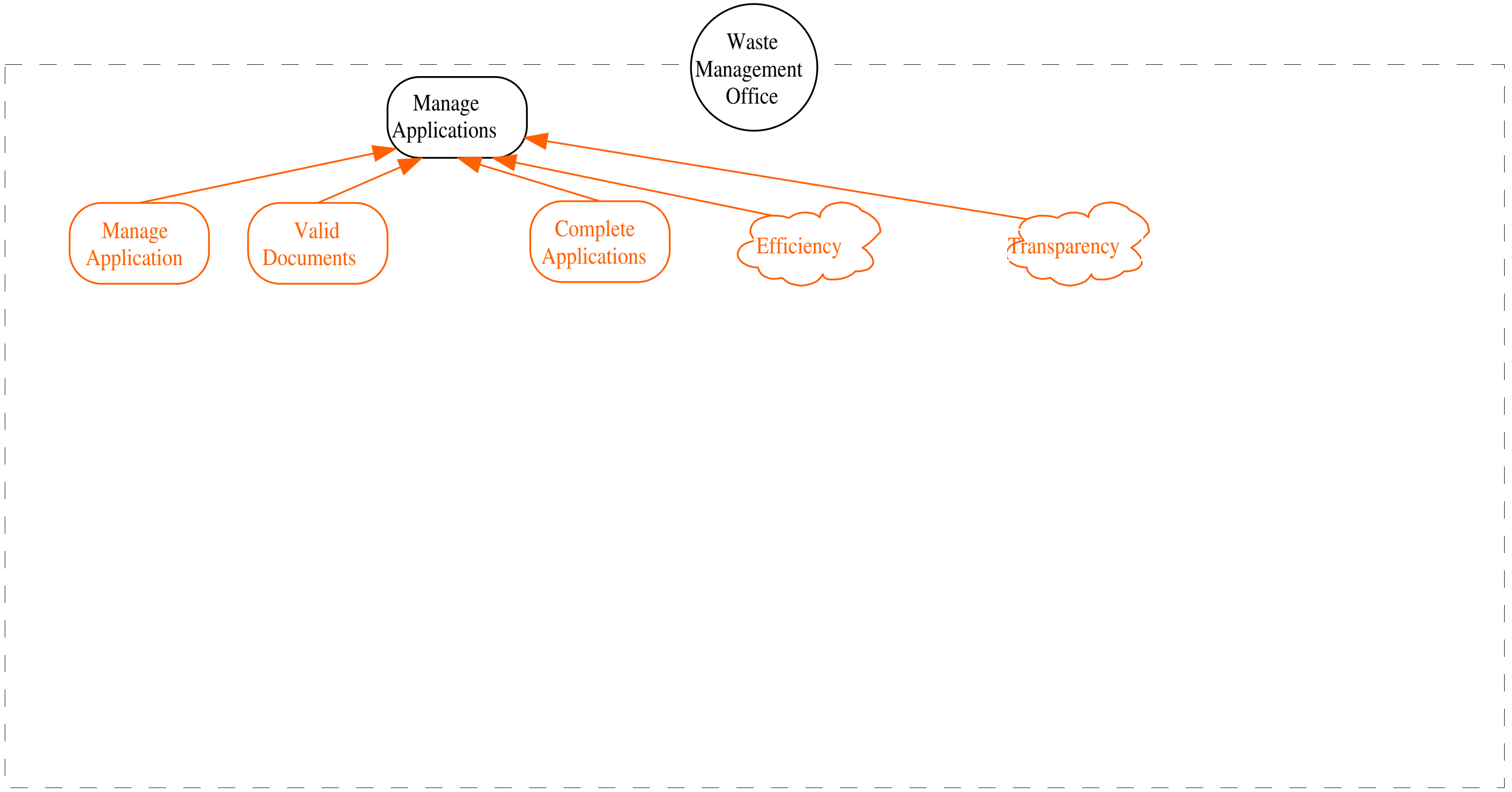
Representation of requirements in collaboration



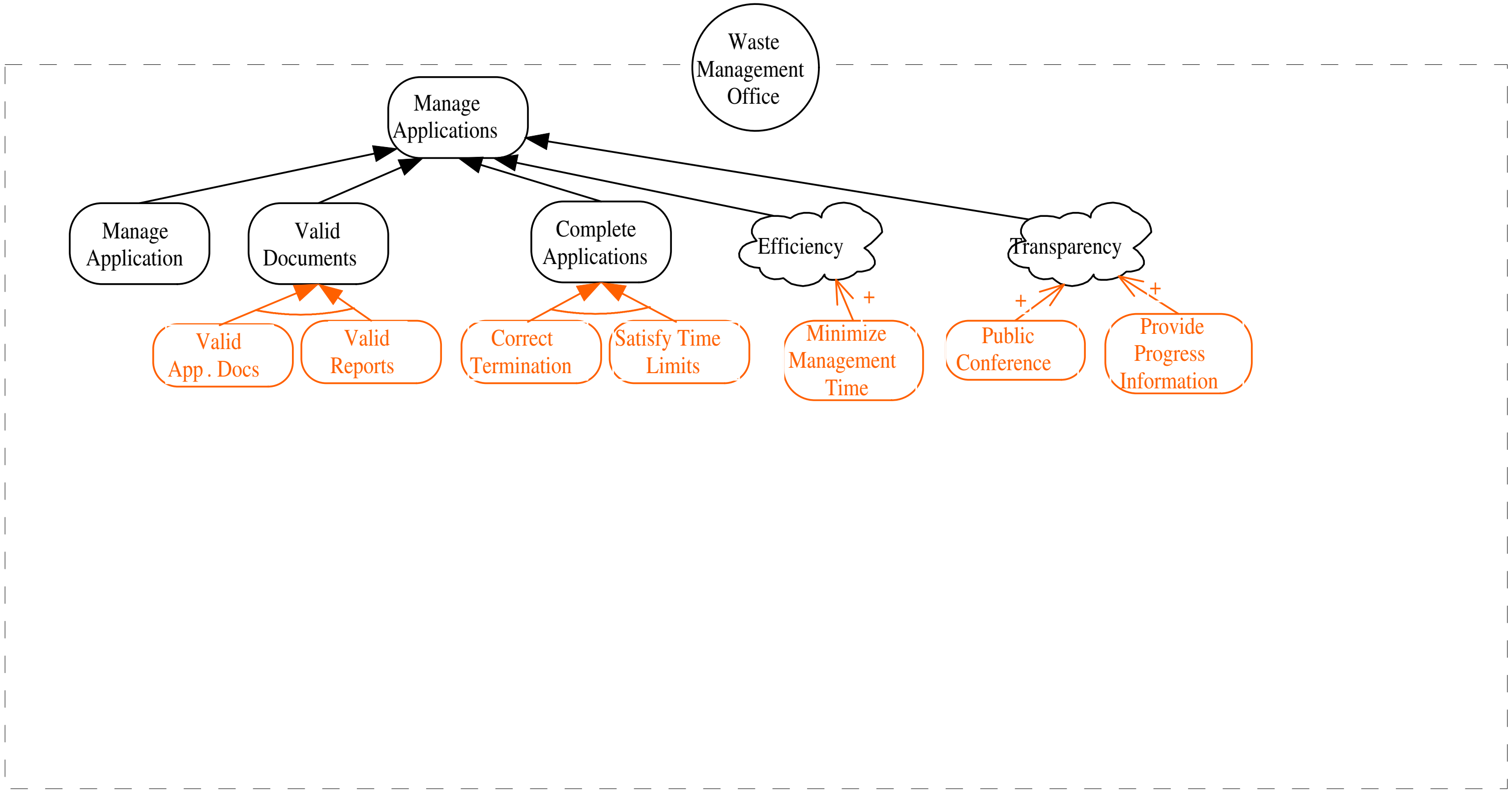
Business Requirements: Refinement



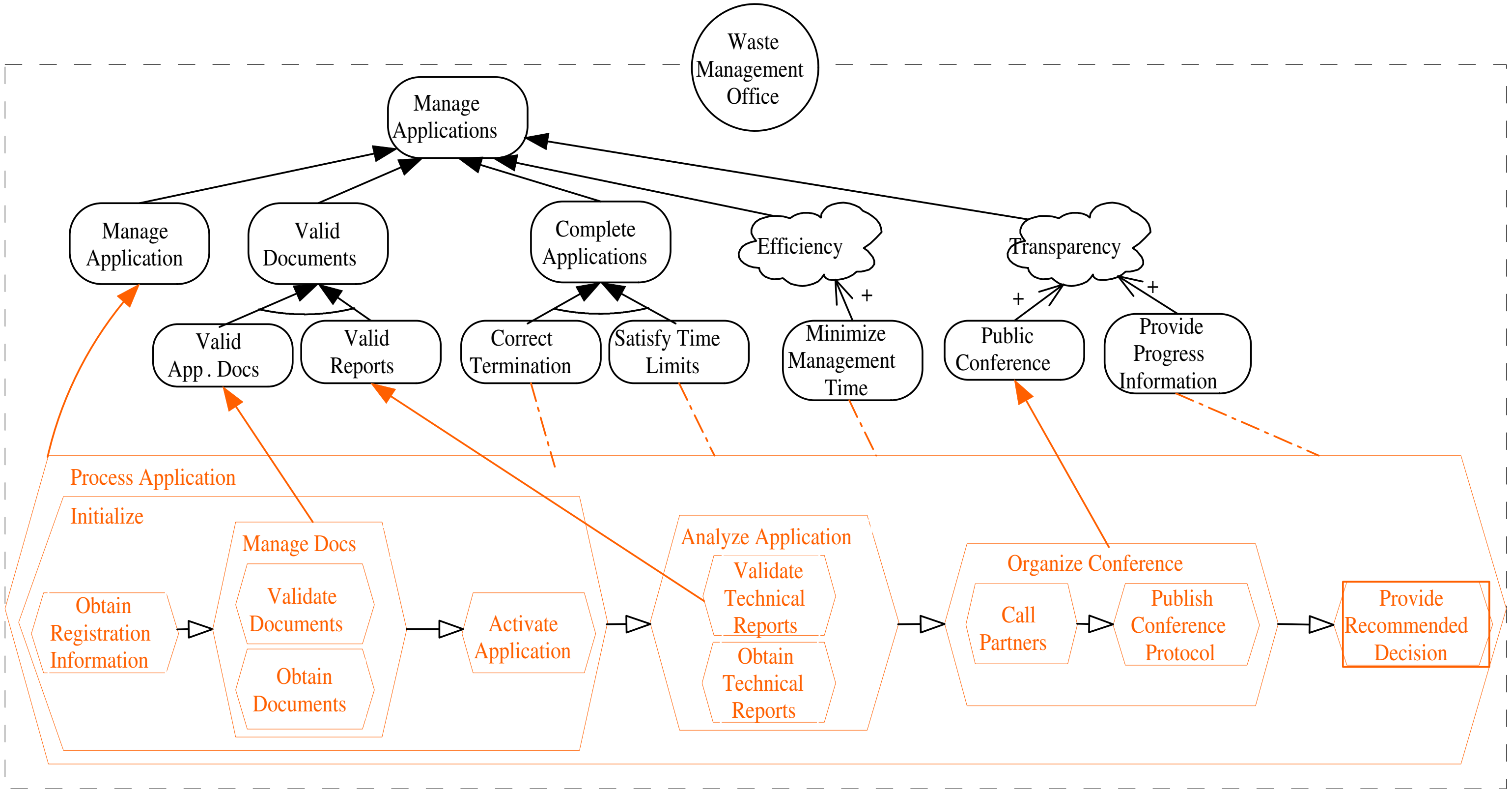
Business Requirements: Refinement



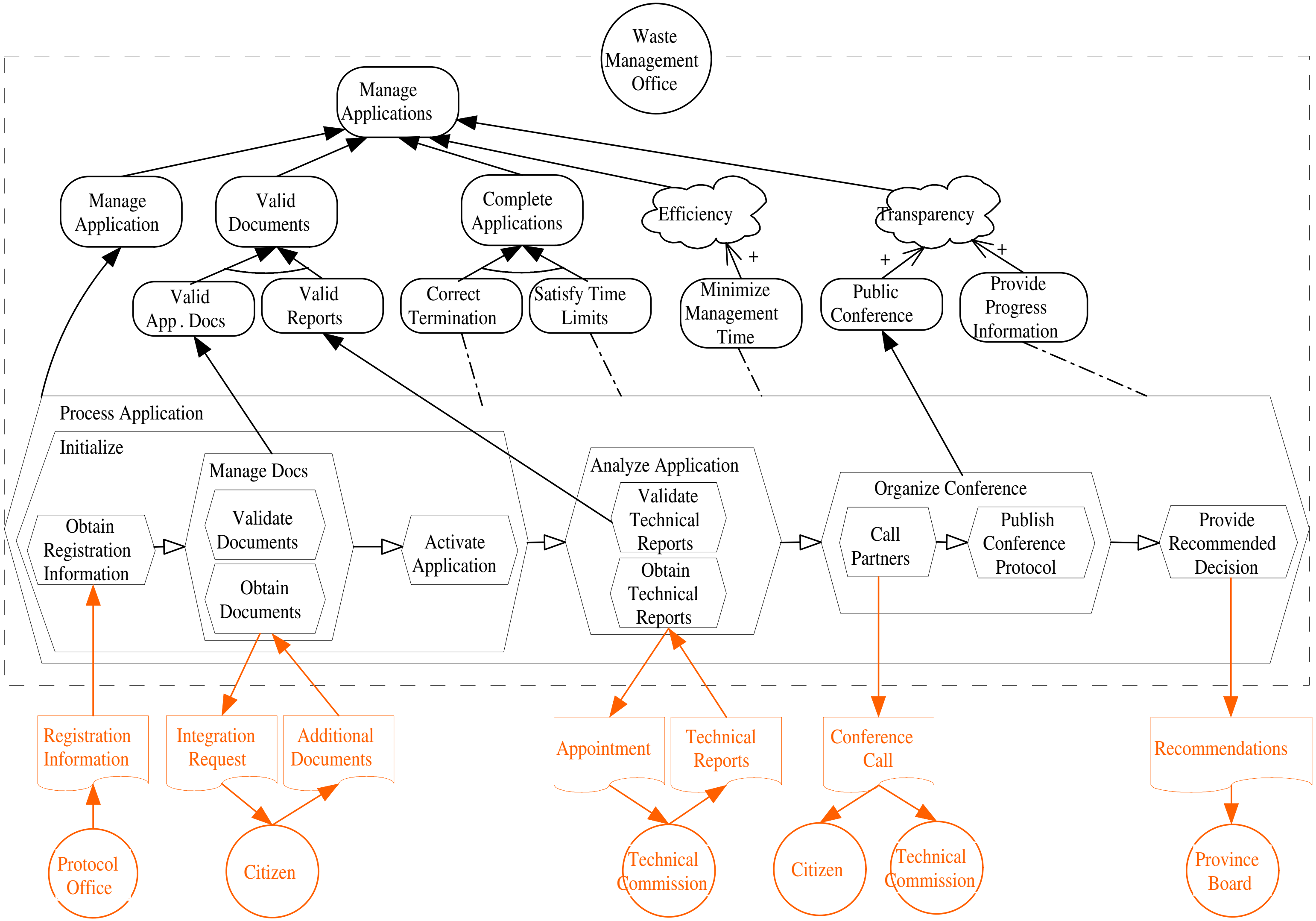
Business Requirements: Refinement



Business Requirements: Refinement

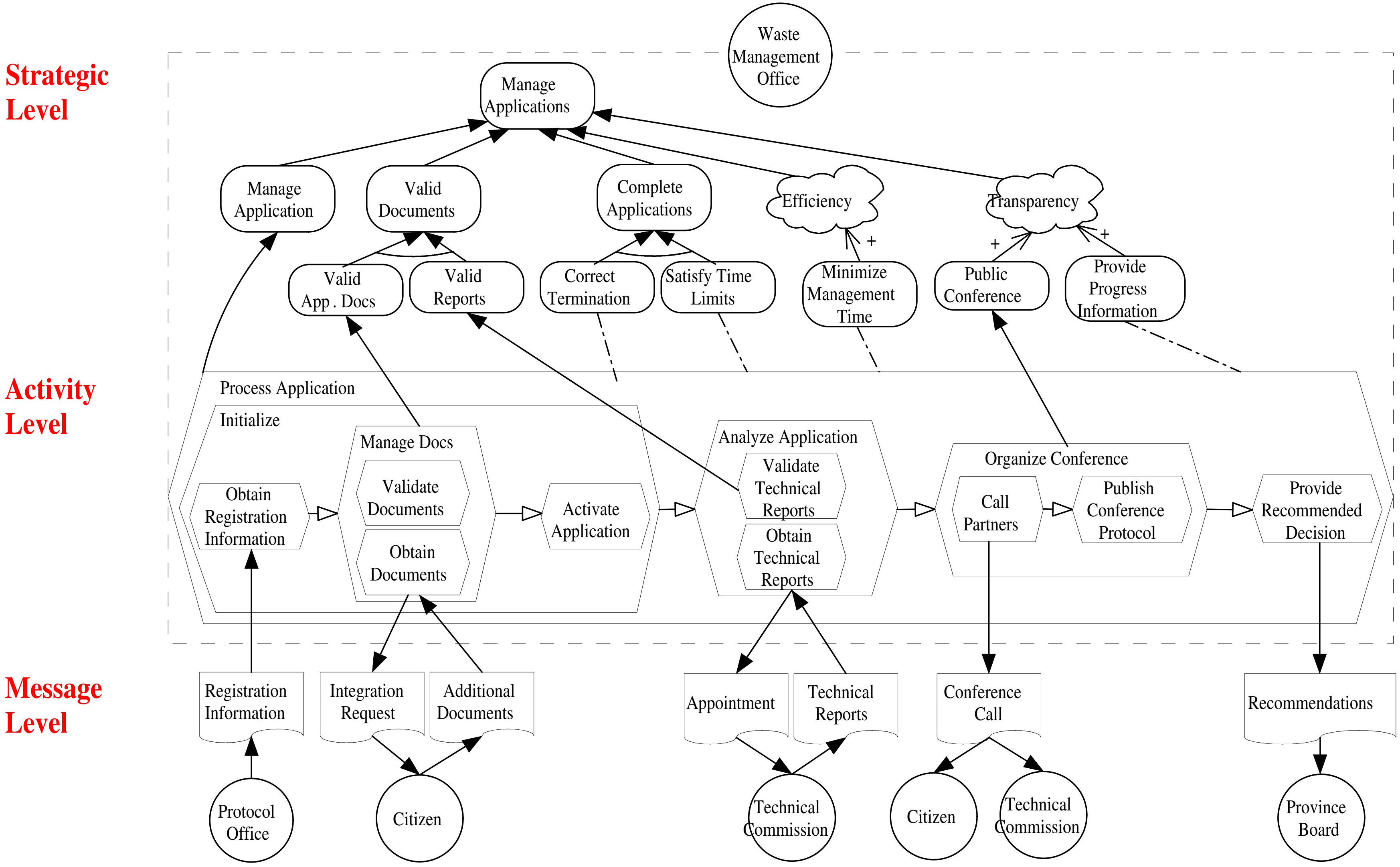


Business Requirements: Refinement

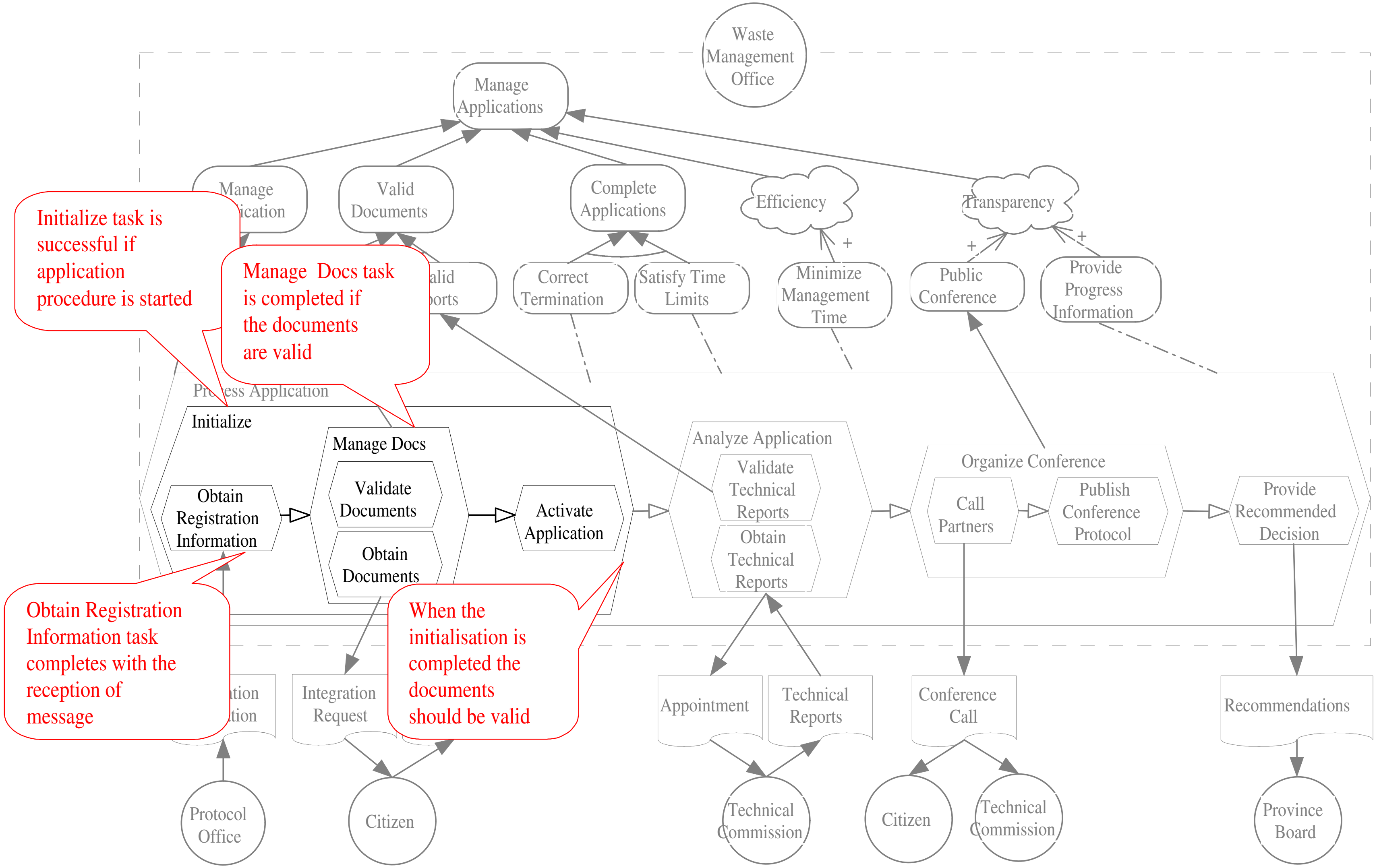


Business Requirements: Refinement

Representation of local requirements



Business Requirements: Formal Properties



Initialize task is successful if application procedure is started

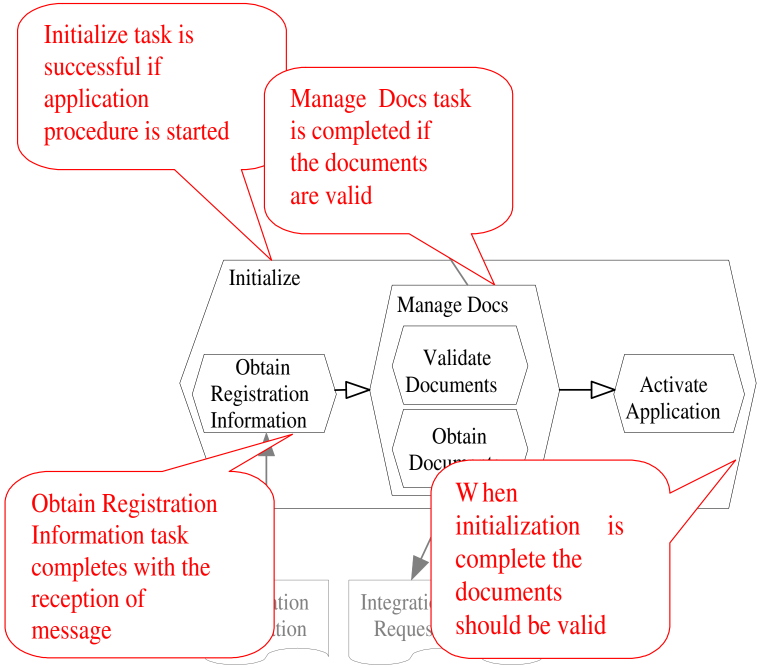
Manage Docs task is completed if the documents are valid

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



Initialize task is successful if application procedure is started

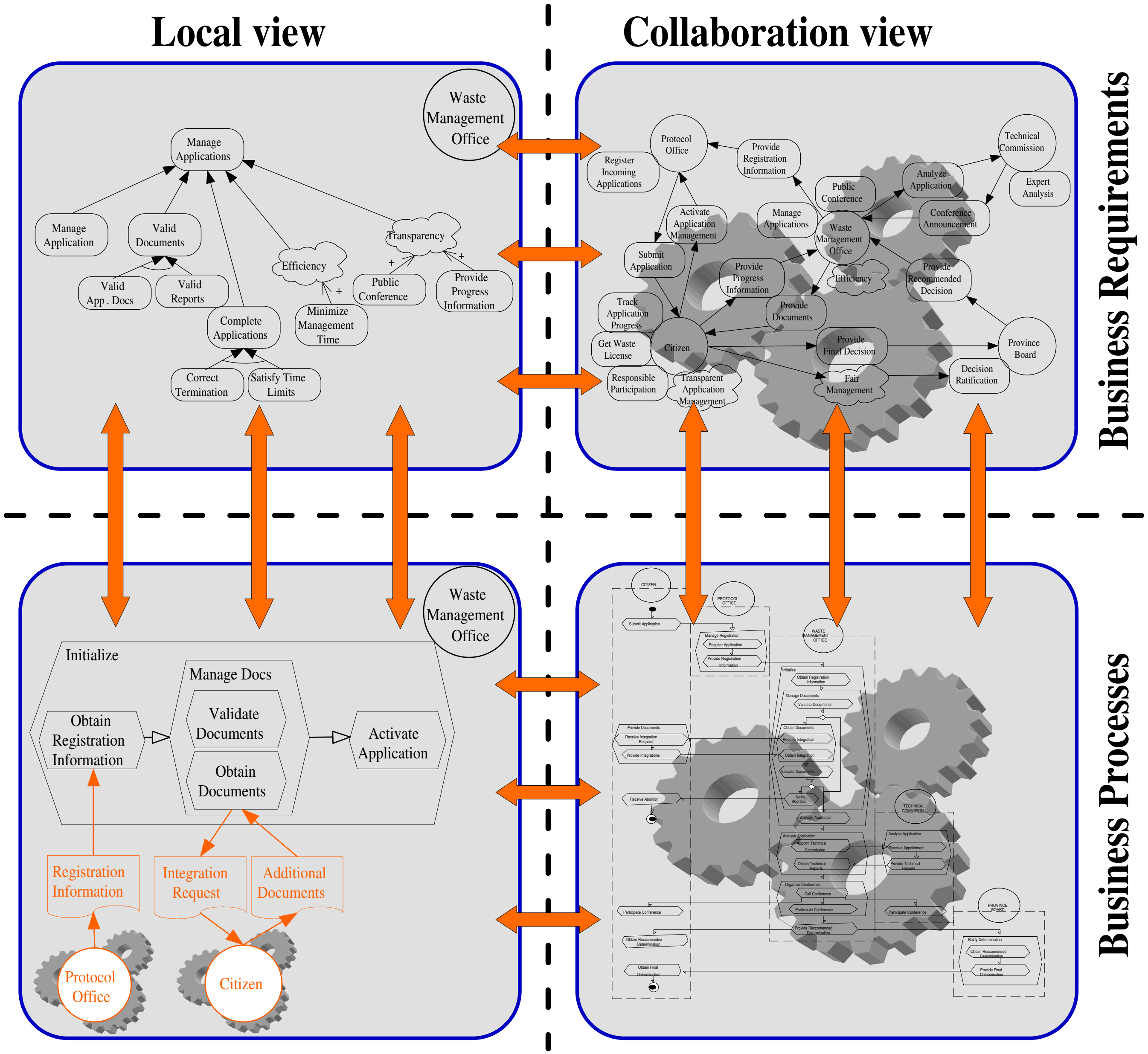
Manage Docs task is completed if the documents are valid

Obtain Registration Information task completes with the reception of message

When initialization is complete the documents should be valid

Task Initialize **mode achieve**
  Attribute docs: Documents
  Fulfillment trigger
    – The initialization task completes with the
    – application activation
    ∃ aa: ActivateApp (aa.**super** = **self** ∧ **Fulfilled**(aa))
  Fulfillment condition
    – when the initialization task completes,
    – the documentation should be valid
    docs.valid

Integrating Requirements and Processes



Formal Verification

- Verification of business requirements
  - consistency checks: “the specification admits valid scenarios”;
  - possibility checks: “there is *some* scenario that respects **possibility** property”:

    $\exists$ 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 $\wedge$ in.docs = ri.docs $\wedge$ **Fulfilled**(in))))

Formal Verification

- Verification of business requirements
  - consistency checks: “the specification admits valid scenarios”;
  - possibility checks: “there is *some* scenario that respects **possibility** property”:

    $\exists$ 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 $\wedge$ in.docs = ri.docs $\wedge$ **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”:

    $\exists\ \text{in: Initialize}\ (\textbf{Fulfilled}(\text{in}))$
  - assertion validation: “*all* scenarios respect **assertion** property”:

    $\forall\ \text{ri: RegInfo}\ (\forall\ \text{wmo: WMO}\ (\text{ri.receiver} = \text{wmo} \wedge \text{ri.docs.valid} \rightarrow$

    $\quad \text{F}\ \ \exists\ \text{in: Initialize}\ (\text{in.actor} = \text{wmo} \wedge \text{in.docs} = \text{ri.docs} \wedge \textbf{Fulfilled}(\text{in}))))$
- 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

Conclusions…

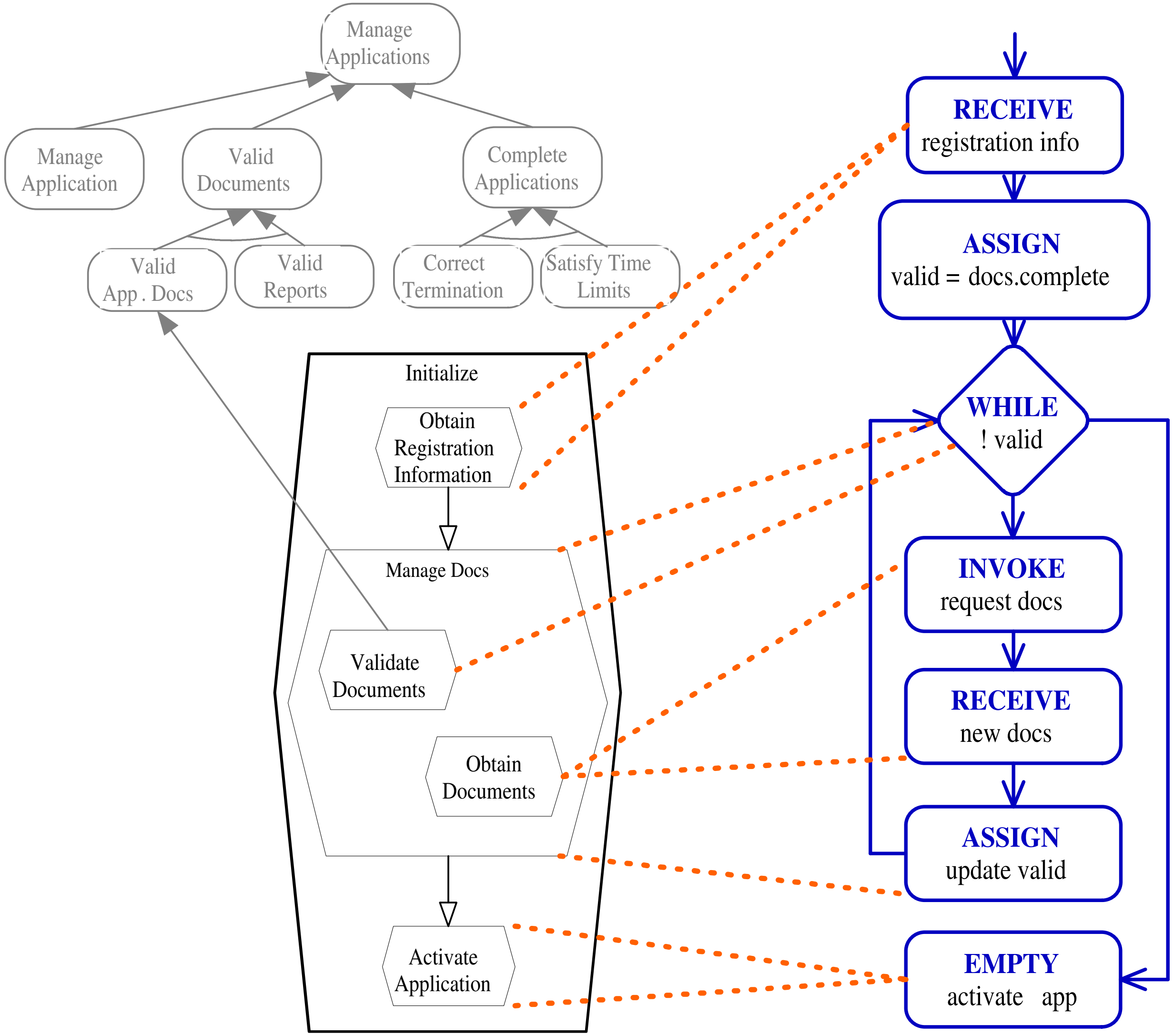
- A methodology for business requirements modelling
  - based on (extension of) Tropos modelling language
  - starting from strategic goals and constraints
  - refining business requirements into business processes

Conclusions

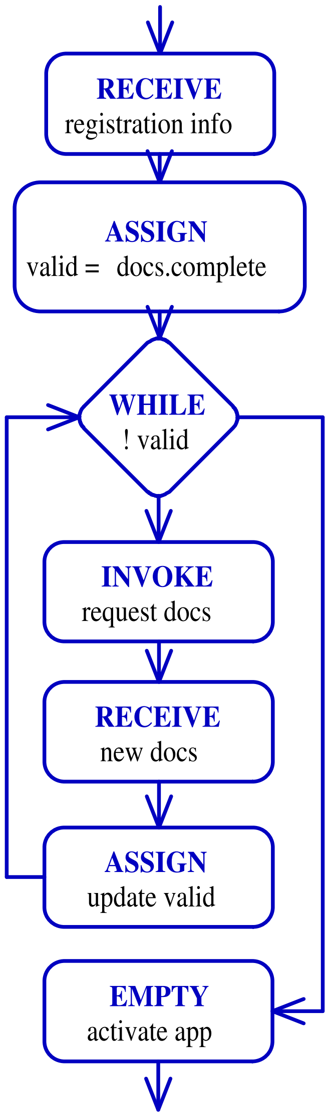
Conclusions…

- 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



Extracting Web Service Processes



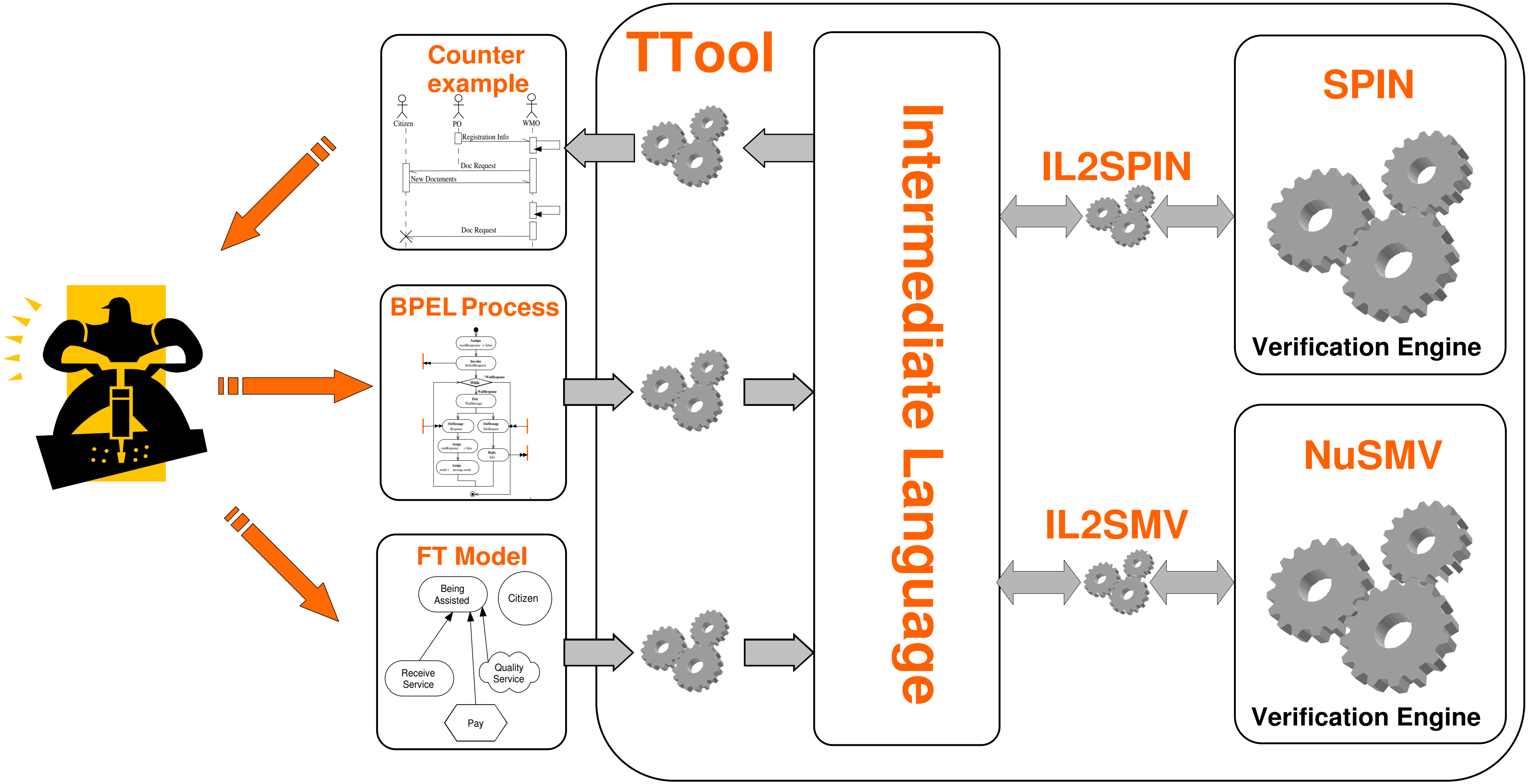
```
<sequence name="Initialize">
  <receive name="receive reg info"
     operation="manageApp" variable="vAppRequest"/>
  <assign>
    <copy>
      <from variable="vAppRequest " query="/docs/complete"/>
      <to variable="valid"/>
    </copy>
  </assign>

  <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

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

Counter example

BPEL Process

FT Model

Intermediate Language

IL2SPIN

IL2SMV

SPIN

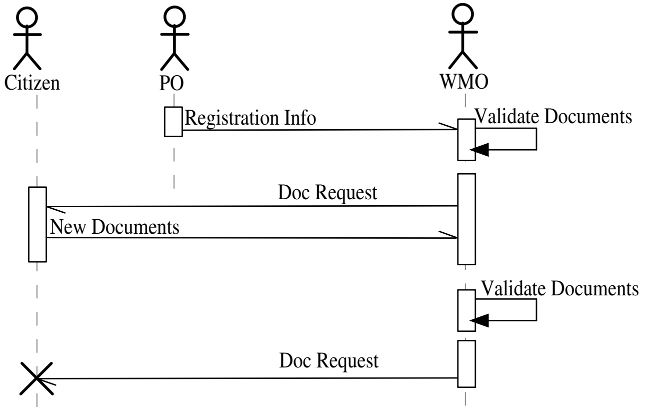
Verification Engine

NuSMV

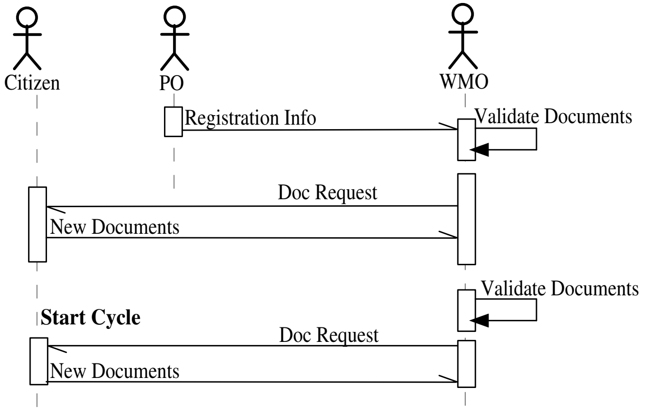
Verification Engine

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

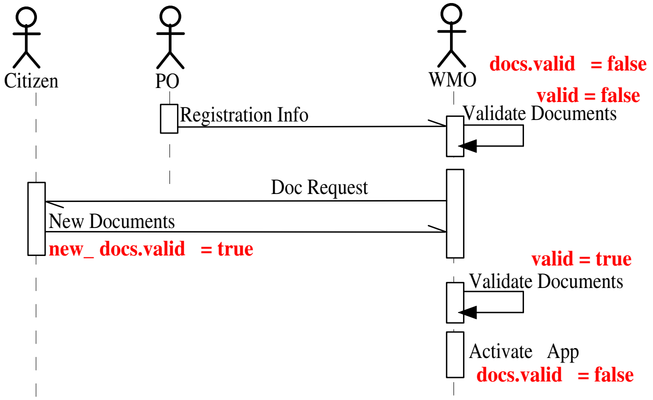


Processes against requirements:

when initialisation completes the documents should be valid

∀ in: Initialize (**Fulfilled**(in) → in.docs.valid)

missed assignment



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!