Specification and Management of Policies

In service oriented business collaboration

Bart Orriens[1] and Jian Yang[2]

[1]Department of Info Management, University of Tilburg, Netherlands

[2]Department of Computing, Macquarie University, Sydney, Australia

Agenda

- Motivation and requirements
- Approach
- Context for business collaboration development (BCD)
- Modeling BCD context
- Specifying and managing policies
- Conclusions and future research
- Discussion

Motivation

- Nowadays enterprizes need to be dynamic and adaptive in order to stay competitive.
- This has given rise to the need for adaptable corporate business services exchanged via business collaboration.
- To realize this the development of business collaborations must be flexible and adaptive.

A Missing Perspective in BPM

- Policies and rules:
  - Where do they exist--- how to find them?
  - What types are they---- what to be found?
  - How should they be captured and managed?
  - How should they be used in the business collaboration development?

- Where and how RuleML can be used?

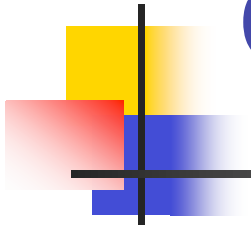
Requirements for Policy and Rule Specification and Management

- Ability to capture both high level business and technical requirements, and their alignment.
- Ability to define the different roles in business collaboration, and their consistency.
- Ability to properly manage changes, meaning their definition, verification, versioning and consistent execution.

Approach

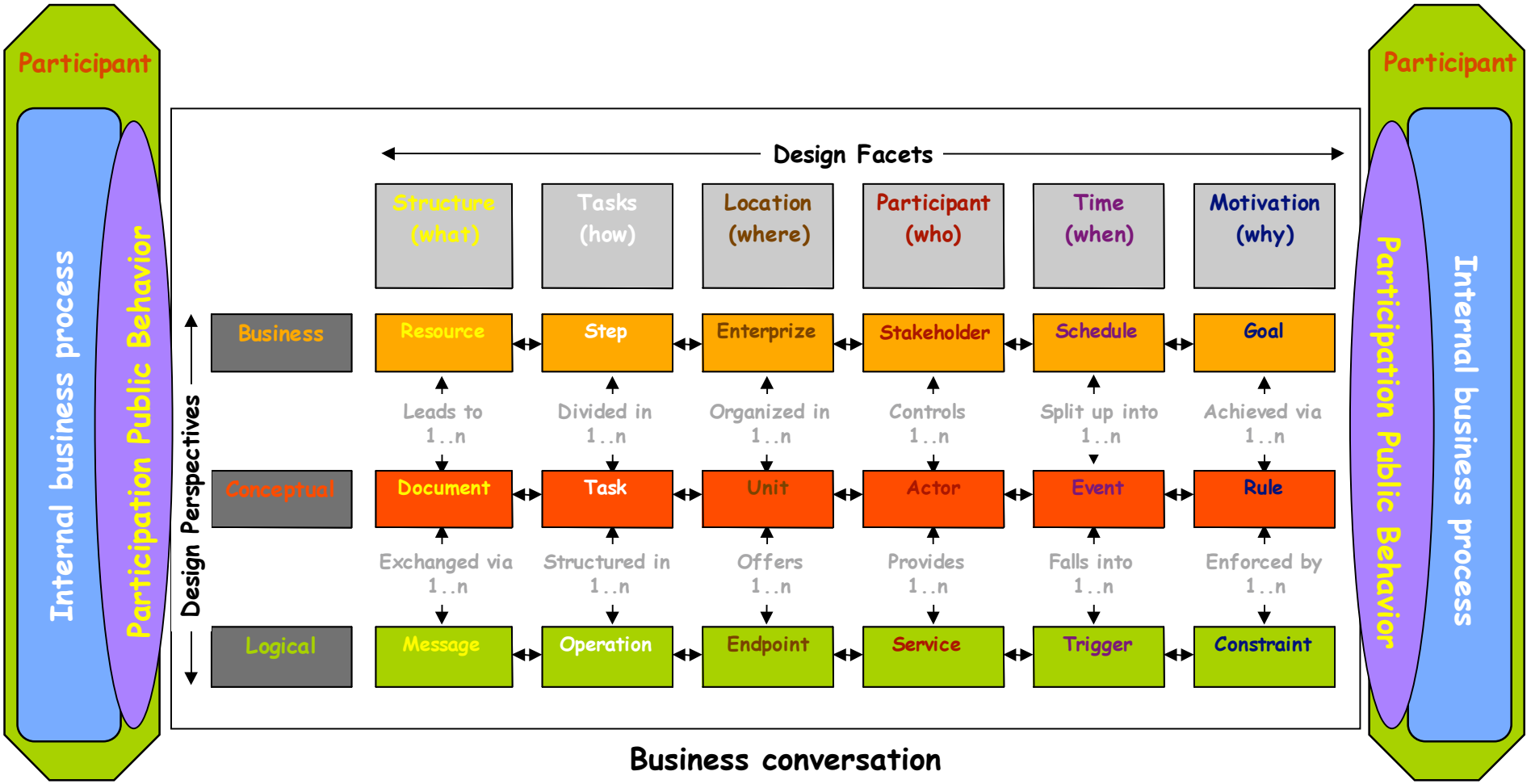
- Explicitly specify policies and use them to drive and constrain the process of business collaboration development
- Flexibility is realized as policies are used for appropriately chaining complex decisions and diagnoses
- Adaptability is achieved as changes can be managed with minimum disruption to existing collaborations through policy modification

Context for business collaboration



- Perspectives: Support 'separation of concern' which allows designers to focus on one level of abstraction without having to concern about the other levels.
- Facets: depict the elements in collaboration design that have different contexts when observed from different perspectives.
- Aspects: represent the different standpoints of the participants and the collaboration itself.

Framework


Participant
Internal business process
Participation Public Behavior
Design Facets
Structure (what)
Tasks (how)
Location (where)
Participant (who)
Time (when)
Motivation (why)
Design Perspectives
Business
Resource
Step
Enterprize
Stakeholder
Schedule
Goal
Leads to 1..n
Divided in 1..n
Organized in 1..n
Controls 1..n
Split up into 1..n
Achieved via 1..n
Conceptual
Document
Task
Unit
Actor
Event
Rule
Exchanged via 1..n
Structured in 1..n
Offers 1..n
Provides 1..n
Falls into 1..n
Enforced by 1..n
Logical
Message
Operation
Endpoint
Service
Trigger
Constraint
Participant
Participation Public Behavior
Internal business process
Business conversation

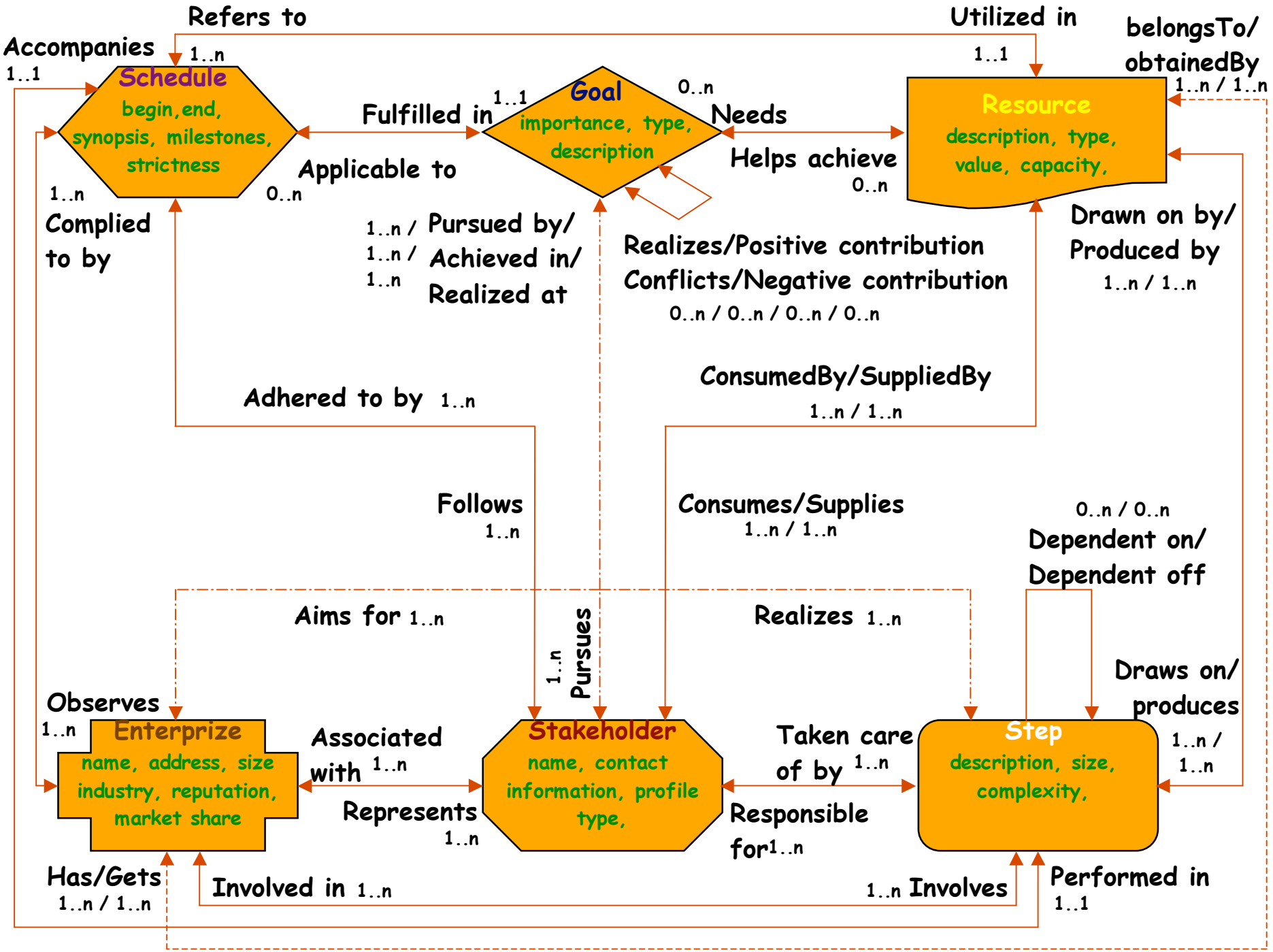
Modeling in the BCDF

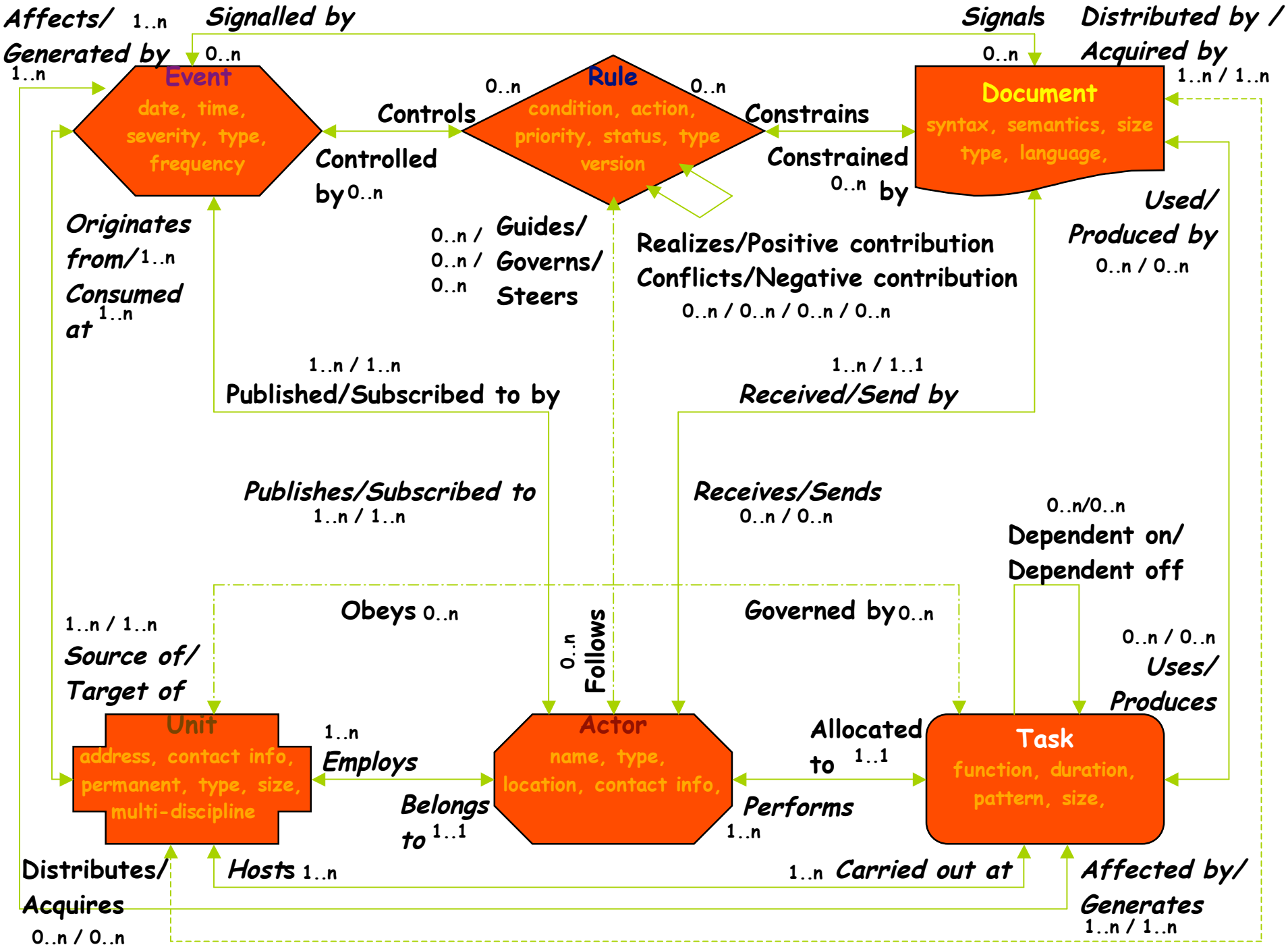
- There are three models in BCDF:
  - The **business model**
  - The **conceptual model**
  - The **logical model**
- Each model describes the relevant elements of the six design facets as classes.
- The relationships connect the classes to indicate the interactions between the design facets.

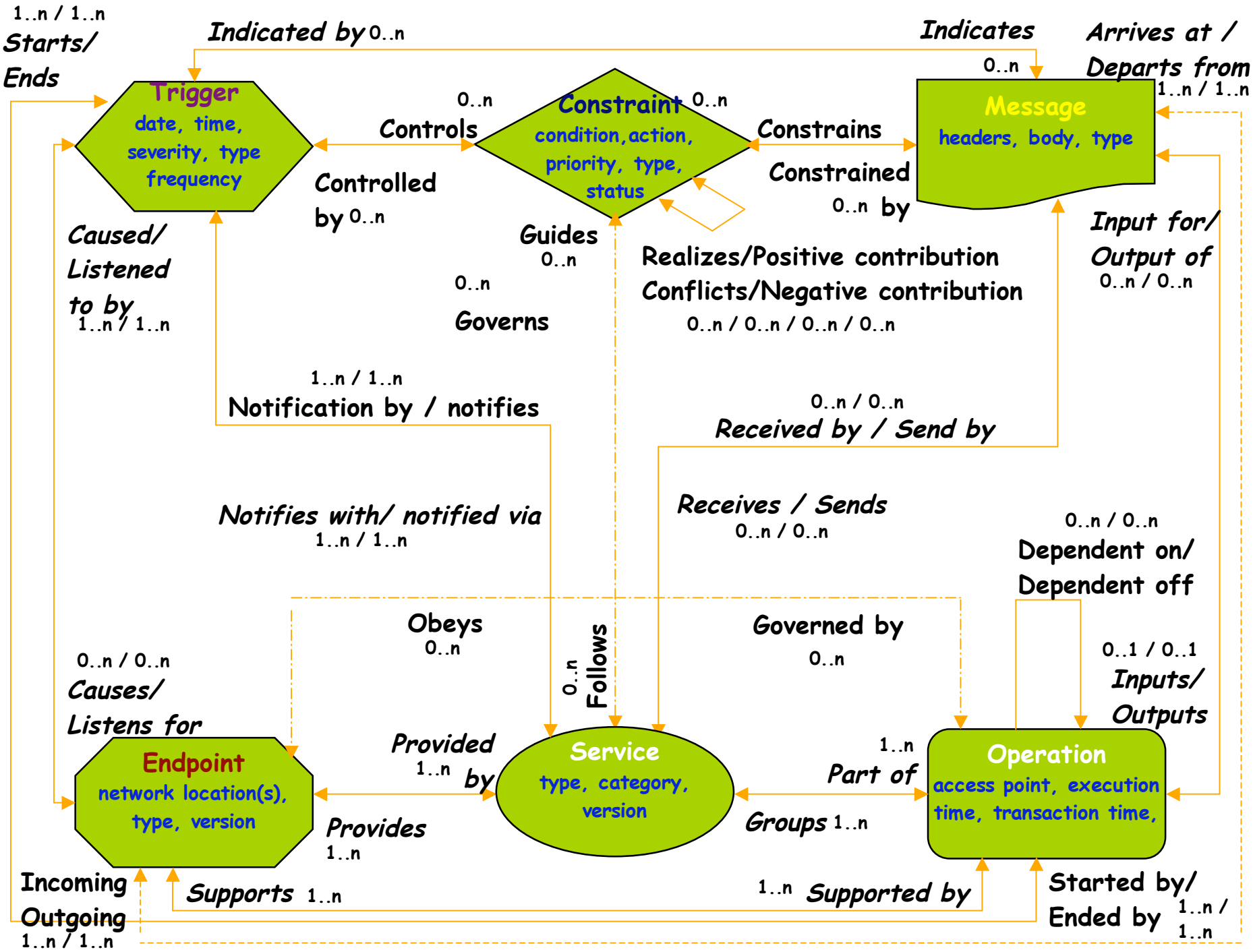
Example

AGFIL cooperates with several contract parties to provide a service that enables efficient claim settlement. The parties involved are Europ Assist, Lee Consulting Services, Garages and Assessors. Europ Assist offers a 24-hour emergency call answering service to policyholders. Lee C.S. coordinates and manages the operation of the emergency service on a daily basis on behalf of AGFIL. Garages are responsible for car repair. Assessors conduct the physical inspections of damaged vehicles and decide to whether to proceed based on the figures provided by the garages.

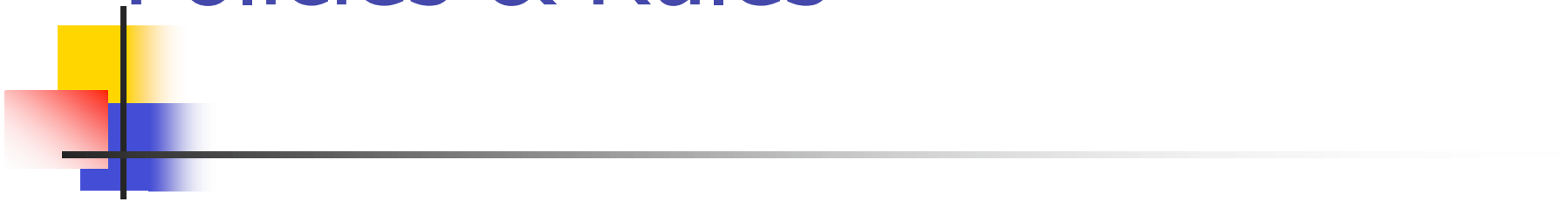
- Parties involved are Europ Assist, Lee Consulting Services, Garages and Assessors.


Refers to
1..n
Accompanies
1..1
Schedule
begin,end,
synopsis, milestones,
strictness
Fulfilled in
1..1
Goal
importance, type,
description
0..n
Needs
Utilized in
1..1
belongsTo/
obtainedBy
1..n / 1..n
Resource
description, type,
value, capacity,
Applicable to
1..n
0..n
Helps achieve
0..n
Complied
to by
1..n /
1..n /
1..n
Pursued by/
Achieved in/
Realized at
Realizes/Positive contribution
Conflicts/Negative contribution
0..n / 0..n / 0..n / 0..n
Drawn on by/
Produced by
1..n / 1..n
Adhered to by 1..n
ConsumedBy/SuppliedBy
1..n / 1..n
Follows
1..n
Consumes/Supplies
1..n / 1..n
0..n / 0..n
Dependent on/
Dependent off
Aims for 1..n
1..n
Pursues
Realizes 1..n
Draws on/
produces
1..n /
1..n
Observes
1..n
Enterprize
name, address, size
industry, reputation,
market share
Associated
with 1..n
Represents
1..n
Stakeholder
name, contact
information, profile
type,
Taken care
of by 1..n
Responsible
for1..n
Step
description, size,
complexity,
Has/Gets
1..n / 1..n
Involved in 1..n
1..n Involves
Performed in
1..1


Affects/ 1..n
Generated by 1..n
Signalled by 0..n
Signals 0..n
Distributed by / Acquired by 1..n / 1..n
Event
date, time, severity, type, frequency
Controls
Controlled by 0..n
0..n
Rule
condition, action, priority, status, type version
0..n
Constrains
Constrained by 0..n
Document
syntax, semantics, size type, language,
Originates from/ 1..n
Consumed at 1..n
0..n / 0..n / 0..n
Guides/ Governs/ Steers
Realizes/Positive contribution
Conflicts/Negative contribution
0..n / 0..n / 0..n / 0..n
Used/ Produced by
0..n / 0..n
1..n / 1..n
Published/Subscribed to by
1..n / 1..1
Received/Send by
Publishes/Subscribed to
1..n / 1..n
Receives/Sends
0..n / 0..n
0..n/0..n
Dependent on/ Dependent off
Obeys 0..n
Governed by 0..n
1..n / 1..n
Source of/ Target of
0..n
Follows
0..n / 0..n
Uses/ Produces
Unit
address, contact info, permanent, type, size, multi-discipline
1..n
Employs
Belongs to 1..1
Actor
name, type, location, contact info,
Allocated to 1..1
Performs 1..n
Task
function, duration, pattern, size,
Distributes/ Acquires
0..n / 0..n
Hosts 1..n
1..n Carried out at
Affected by/ Generates
1..n / 1..n

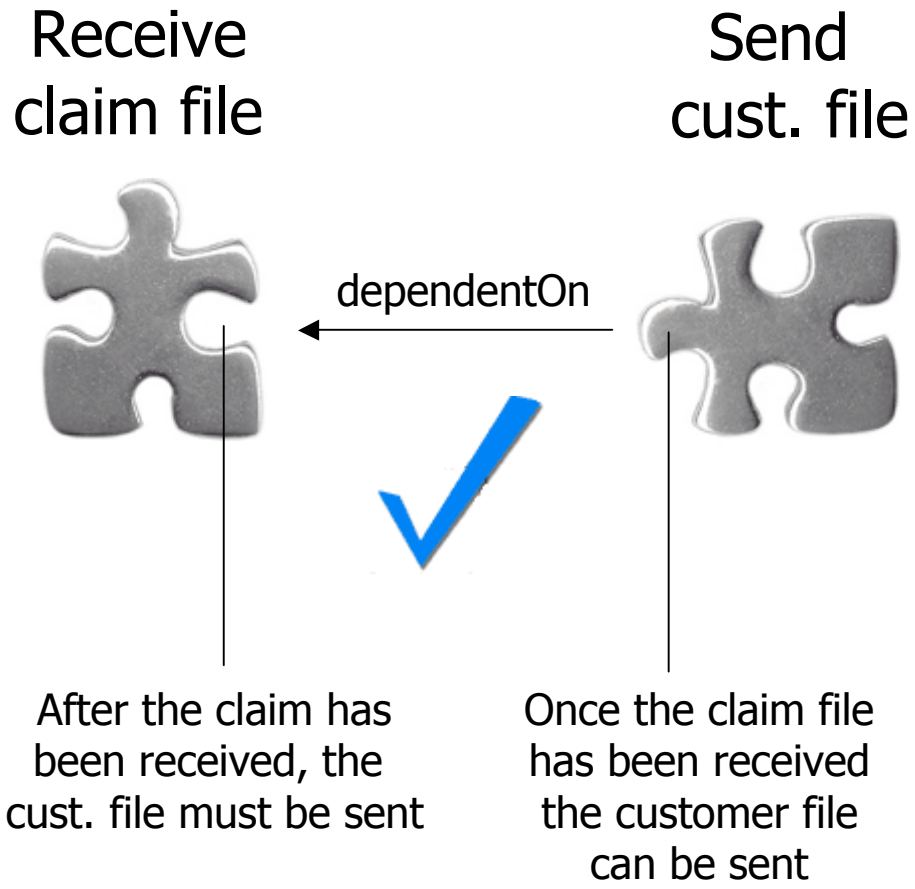

1..n / 1..n
Starts/
Ends
Indicated by 0..n
Indicates
0..n
Arrives at /
Departs from
1..n / 1..n
Trigger
date, time,
severity, type
frequency
0..n
Controls
Constraint
condition, action,
priority, type,
status
0..n
Constrains
Message
headers, body, type
Controlled
by 0..n
Constrained
0..n by
Caused/
Listened
to by
1..n / 1..n
Guides
0..n
Input for/
Output of
0..n / 0..n
0..n
Governs
Realizes/Positive contribution
Conflicts/Negative contribution
0..n / 0..n / 0..n / 0..n
1..n / 1..n
Notification by / notifies
0..n / 0..n
Received by / Send by
Notifies with/ notified via
1..n / 1..n
Receives / Sends
0..n / 0..n
0..n / 0..n
Dependent on/
Dependent off
Obeys
0..n
0..n
Follows
Governed by
0..n
0..1 / 0..1
Inputs/
Outputs
0..n / 0..n
Causes/
Listens for
Provided
1..n
by
Service
type, category,
version
1..n
Part of
Operation
access point, execution
time, transaction time,
Endpoint
network location(s),
type, version
Provides
1..n
Groups 1..n
Incoming
Outgoing
1..n / 1..n
Supports 1..n
1..n Supported by
Started by/
Ended by
1..n /
1..n

Policies & Rules

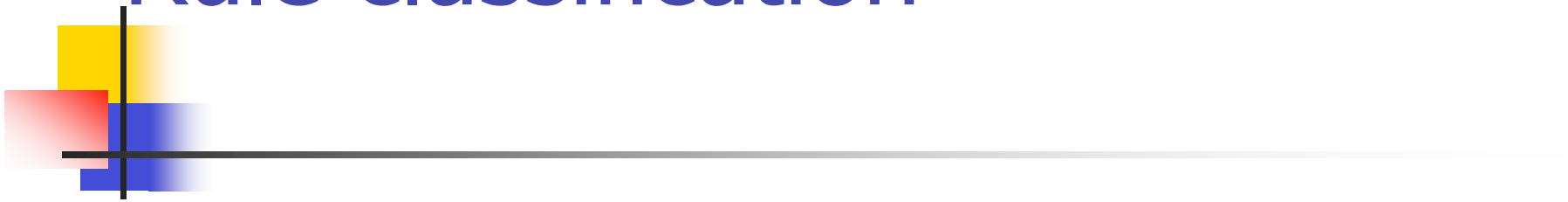


- Informally constitute a plan to guide part of the business collaboration behavior; formally a set of logically related rules
- Are associated with the modeling description elements in the BCDF models, constraining the manner in which they can be connected to other description elements
- In this manner elements can be combined in a plug and play manner, where the policies ensure that the resulting design is conform requirements

Puzzle analogy

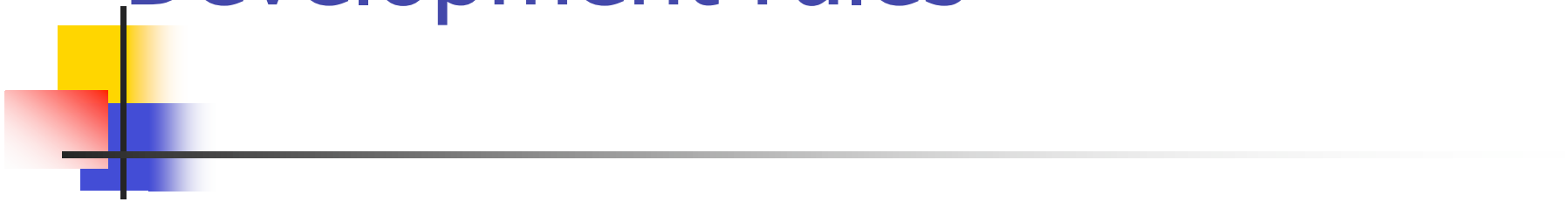


Rule classification



- Development rule are specific to a business collaboration
  - By perspective: business, conceptual, logical policies
  - By facet: structural, functional, organizational, participant, temporal
  - By aspect: internal, behavorial, collaboration policies
- Management rules; are generic for all business collaborations
  - By purpose: completeness, correctness, consistency
  - By model: normal, mapping, connection

Development rules



- Each element has development policies associated with it
- Type of development rules is determined by
  - Type of element
  - Type of aspect in which element is used
- For example, the policies of ***claim file*** in AGFIL Conceptual Model can be classified as:
  - By perspective: conceptual rule
  - By facet: structural rule
  - By aspect: collaboration rule

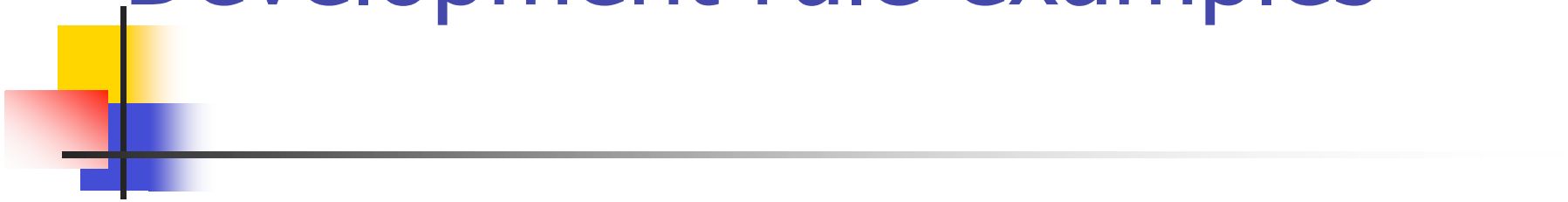
Management rules

- By purpose:
  - Completeness: e.g. that no elements or links missing
  - Correctness: e.g. that properties do not have illegal values
  - Consistency: e.g. that there is no activity deadlock
- By model:
  - Normal: manage models that describe an individual perspective and aspect
  - Mapping: manage how two perspective models are related to each other
  - Connection: manage how two aspect models are related to each other
- For example:
  - Mapping between ***AGFIL Conceptual Model*** and ***AGFIL Logical Model***
  - Consistency rule ensures that they are properly aligned;
  - By constraining the mappings in the mapping model

Specification

- Policies, business rules, and constraints constitute
  - logically related sets of rules
  - alternatives describe how to handle different scenarios
- Each rule is grounded on the modeling description elements, constraining
  - existence of elements
  - existence of links
  - value of properties
- Currently uniformly expressed using RuleML, a standard for business rule specification (http://www.ruleml.org)

Development rule examples



- If claim file concerns a claim higher than $500, then make sure that it can not be tampered with

  property(prop^ integrity,true,claimFile) :- element(claimFile^,document)^ link(link^ has, claimFile, claimFileValuePart) ^ property(prop1^ value,?X, claimFileValuePart)^ greaterThan(?X,500)

- After a claim has been received, the corresponding customer file must be sent

  link(myLink^ dependentOn, sendCustomerFile, receiveClaim) :- element(receiveClaim^,task)

Management rule examples

- Every document must be sent by someone

  elementLinkMissing(sendBy, ?Element) :- element(?Element^ document, ?Model), Naf(link(?Link^ suppliedBy, ?Element, ?Actor, ?Model)).

- The begin date of an event corresponding to a schedule must be greater or equal to the schedule's begin date

  mappingPropertyConflict(?Mapping, ?PropUp, ?PropLow) :-
  mapping(?Mapping^ leadsTo, ?Upper, ?Lower, ?ModelMap),
  property(?PropUp^ begin, ?ValueUp : DateTime, ?Upper, ?ModelUp),
  property(?PropLow^ date, ?ValueLow : Date, ?Lower, ?ModelLow),
  greaterThan(?ValueUp, ?ValueLow).

Usage of development rules

- Apply development rules at design time:
  - to determine possible options, e.g. to which actors a task may be allocated
  - to inform the designer that a performed modeling action is in violation of specified policies, e.g. selected service does not meet set out requirements to carry out task
- Use development rules at runtime:
  - to derive collaboration models just prior to execution
  - to gradually generate collaboration models dynamically so any decision is postponed until the last possible moment

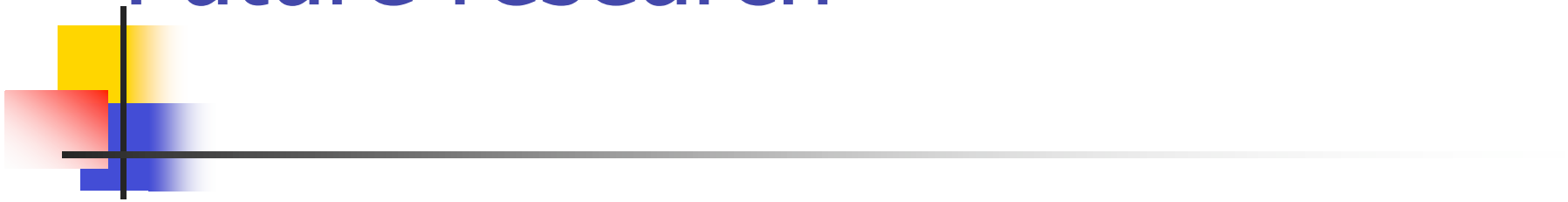
Usage of management rules

- Apply management policies at design time:
  - to inform the designer that a model is incomplete, incorrect or inconsistent in some way
  - to automatically prompt the designer to provide the necessary information to complete, correct or make the model consistent
- Use management rules at runtime:
  - to ensure that the generated model just before runtime is syntactically and semantically sound
  - to enforce that every step in the execution is in line with specified policies in terms of completeness, correctness and consistency

Conclusions

- Currently no proper support for capturing policies, business rules, and contraints concerning both business and technical requirements, as well as policies from different viewpoints
- BCDF provides context for policy, business rules, constraints specification via its perspectives, facets and aspects, and dependencies among them
- Using BCDF context policies, business rules, constraints are classified in terms of their purpose in business collaboration development
- Subsequently they are used to drive and constrain business collaboration development, making it flexible and adaptive

Future research



- Complete the analysis of different types of policies, business rules, and constraints and their application during design and runtime
- Expand the BCDF to support additional requirement specification like security, QoS, assessment, etc.
- Finish development of a small prototype, called Icarus, to illustrate the approach and its benefits.

Thank you for your attention!

?