ICEIS 2018 - Keynote Address

The Future of Information Systems: Direct Execution of Enterprise Models, Almost Zero Programming

David Aveiro

Assistant Prof. at University of Madeira
Researcher at Madeira Interactive Technologies Institute
Member of the CIAO! Enterprise Engineering Research Network

CIAO!

Communication **I**nformation
Action and **O**rganization

The CIAO! Network

東京工業大学
Tokyo Institute of Technology

IBM
Research Almaden, USA

TU Prague

UNIVERSIDADE da MADEIRA

NOVI

TU Delft
Delft University of Technology

INSTITUTO SUPERIOR TÉCNICO
TU Lisboa

HIGHER SCHOOL OF ECONOMICS
NATIONAL RESEARCH UNIVERSITY
Moscow and Nizhniy Novgorod, Russia

Universiteit Antwerpen

tudor
Public Research Centre, Luxembourg

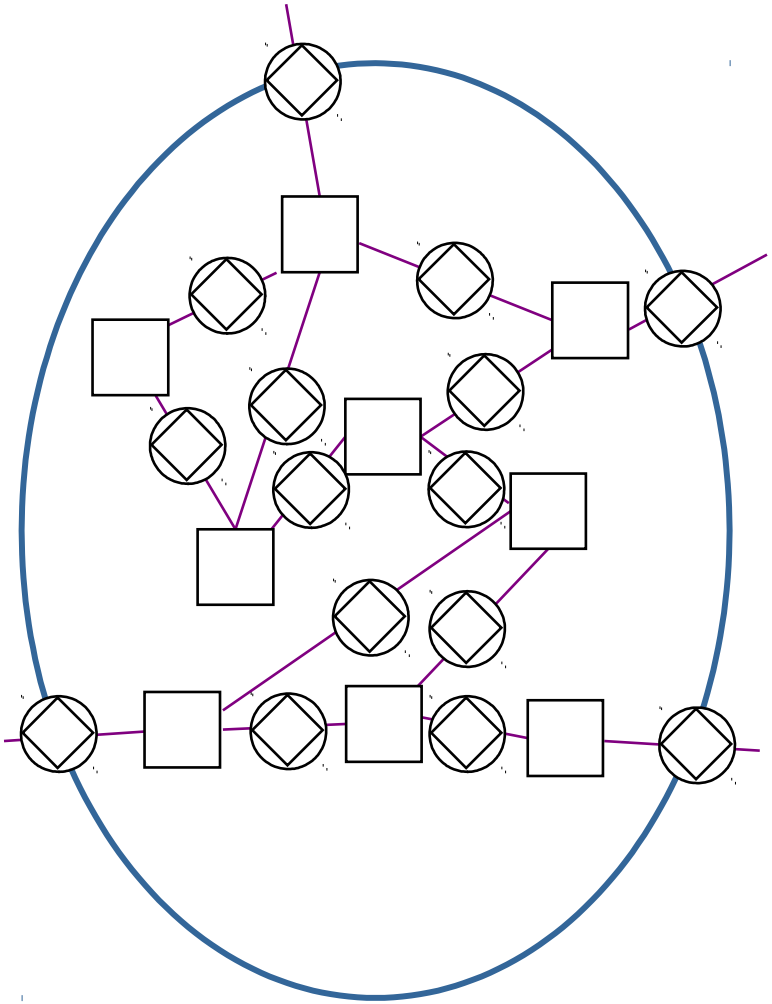
University of St.Gallen

The key notion of organisation

Every organized human activity - from the making of pots to the placing of a man on the moon - gives rise to two fundamental and opposing requirements: the division of labor into various **tasks** to be performed and the **coordination** of these tasks to accomplish the activity.

Henry Mintzberg, The Structuring of Organizations, 1979

It's all about production and coordination ...

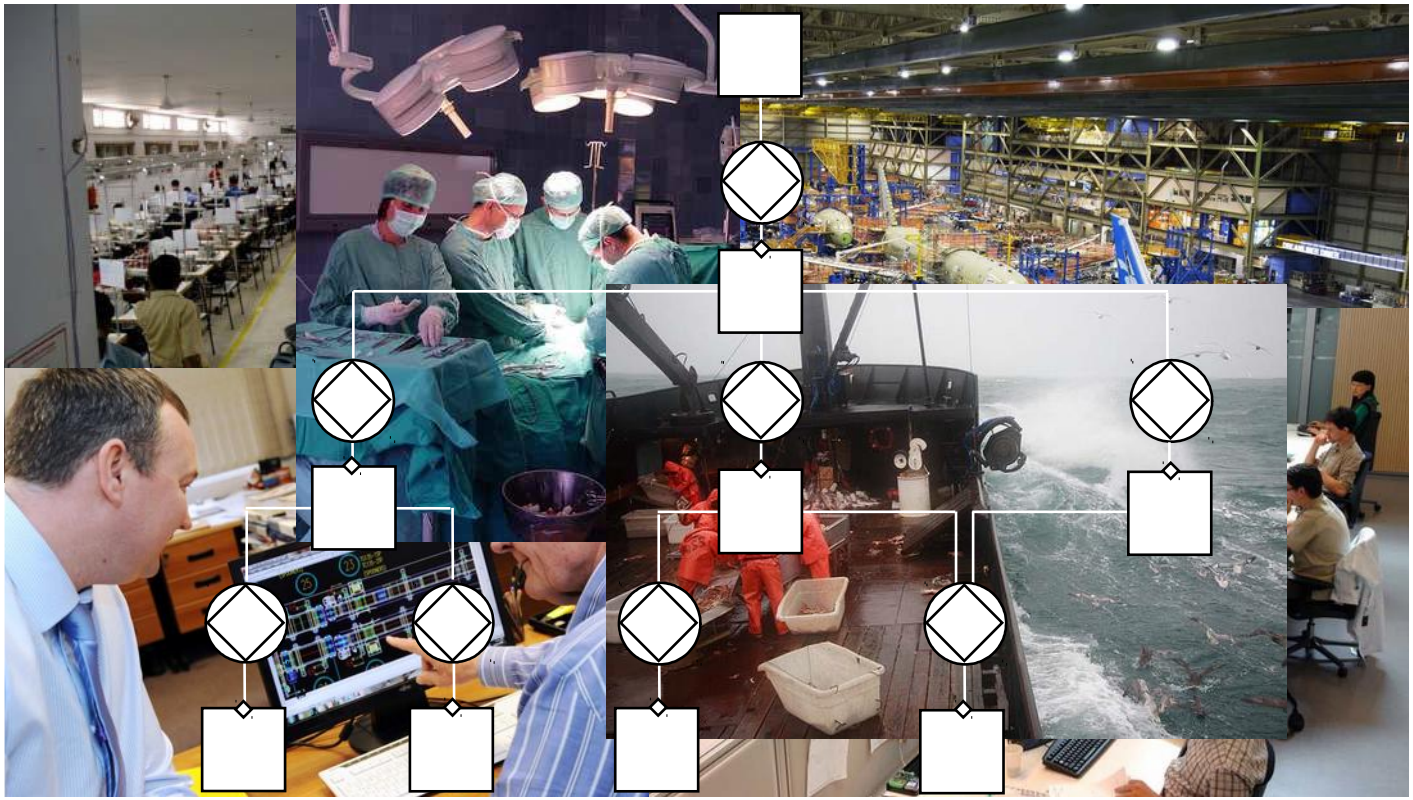


Mintzberg's division of labor is actually a division in **actor roles**: the 'production units' that bring about the (sub and end) **products** of the organisation.

Production and coordination occur in universal patterns, called **transactions**. A transaction comprises 4 to 20 generic **coordination steps** regarding 1 specific **production step**.

The **operating principle** of every organisation is that **actors** in performing **coordination steps**, *enter into and comply with commitments* regarding a **production step**.

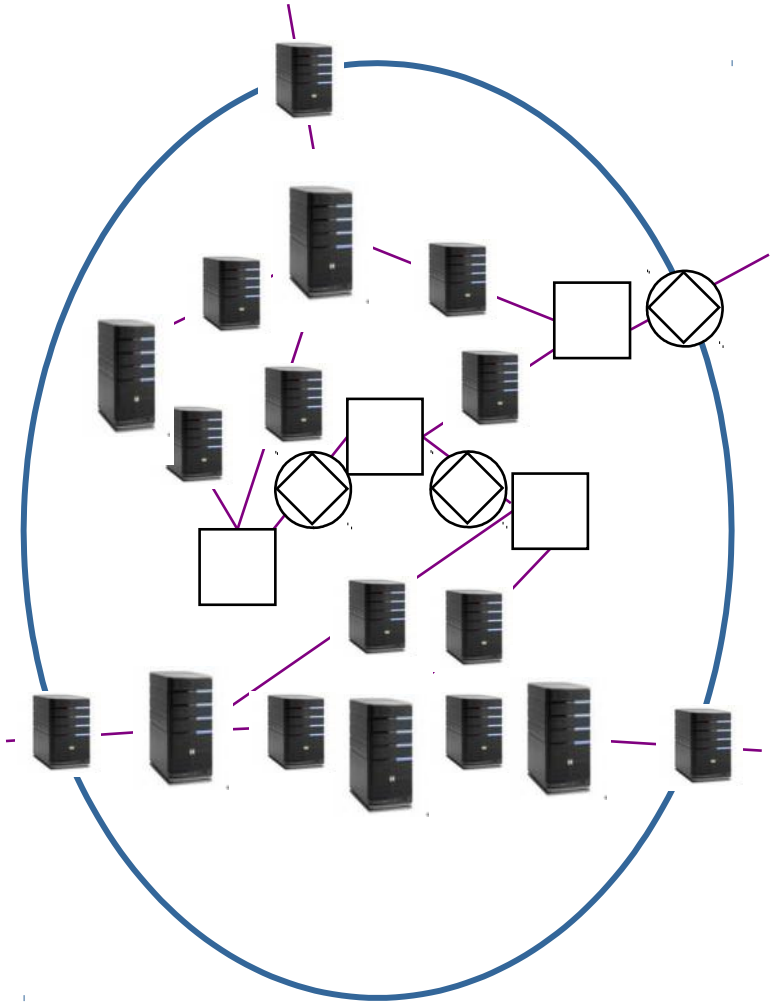
… and about organisational structure



Mintzberg states that the **structure** of an **organisation** is simply the sum total of the ways in which it divides its labor into distinct tasks and then achieves coordination among them.

Enterprise Engineering states that this structure consists of **trees** of **building blocks** that correspond with the **structures** of **products**.

Does ICT matter?



Yes and no…

Information storage and retrieval, as well as communication, needs some **technological means**.

All **coordination steps** that were performed in the past using 'human' and paper technology, can also be performed using **modern ICT**

In addition, many **production steps** that were performed in the past by human actors, can be supported by **modern ICT-applications**

but … only **human actors** can be and are **responsible** for these steps!

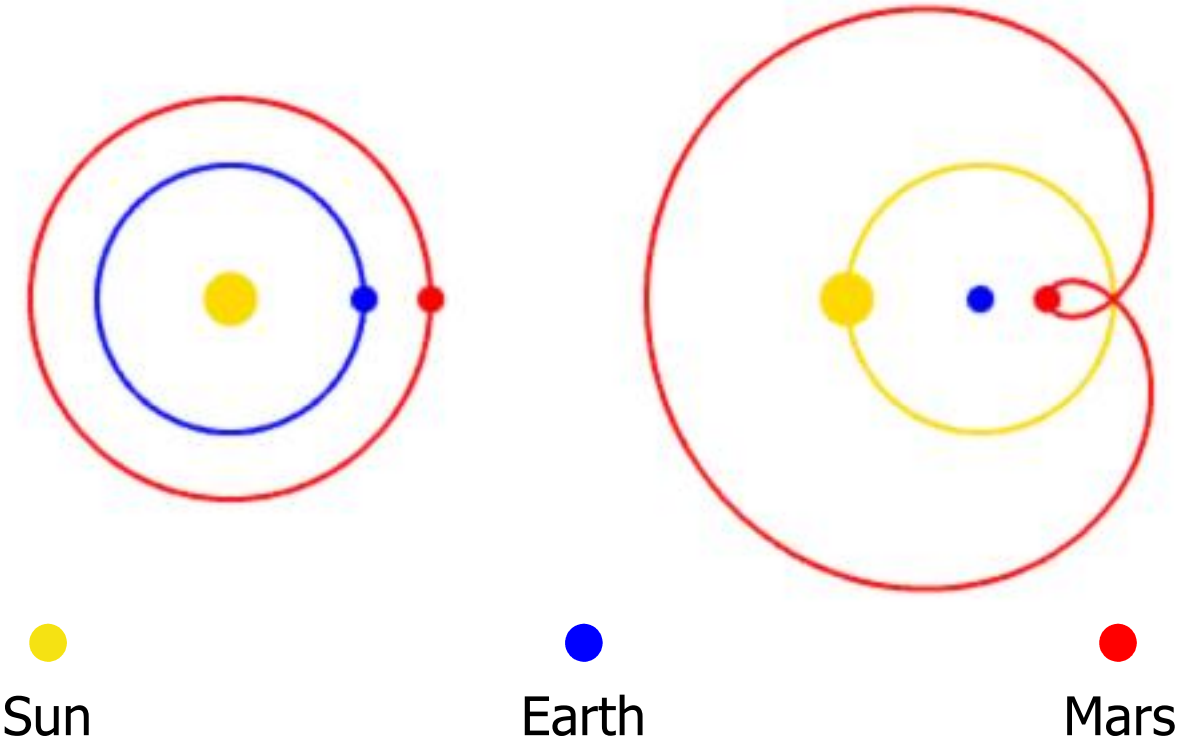
The CIAO

Paradigm

The paradigm shift in astronomy

The *heliocentric view*
(from 1550 on - Copernicus)

The *geocentric view*
(till 1550 - Ptolemy)



Sun Earth Mars

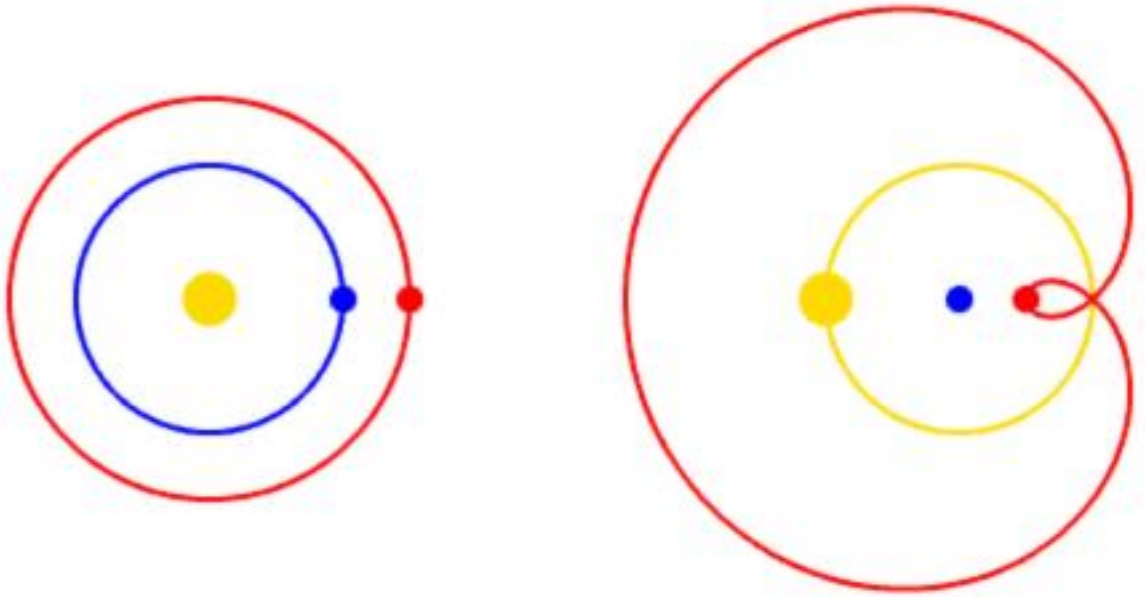
"We can't solve problems by using the same kind of thinking we used when we created them"

(Albert Einstein)

The paradigm shift in information systems

The *communication-centric view* (from 2000 on)

The *information-centric view* (from 1975 on)



communication

information

action
organisation

The information-centric view: basics

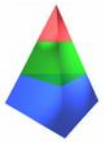
- **Information** is the primal notion, defined as the *representation of (factual) knowledge*
- **Communication** is the *exchange of information* between subjects (but also between animals and machines)
- **Action** is not connected to information and communication, although it may be triggered by (the receiving of) information, and although it may create information
- **Organisation** is considered to be something different, although it generally implies action and communication and information

The information-centric view: effects

- The task of the *information system engineer* is to develop an information system (ICT-application) 'on the side', and to 'implant' it in the organisation, once it is completed
- The development process is roughly: *requirements determination*, *functional design*, *technical design*, *implementation*
- The *functional requirements are determined* basically *by interviewing* the customer and the future users
- The development methods focus on information flows, data bases, and ICT opportunities.

The information-centric view: outcomes

- The delivered system rarely meets the 'real' functional requirements (expectations): requirements determination mostly falls short
- Standard software packages (like ERP-systems), hardly solve the problem. Instead, they put organisations in 'armours'
- The notion of business process is ill-understood. There is no clear distinction between business process and work or information (or data) flows
- Consequently, information systems engineering is quite disconnected from the supported (people in the) organisation
- Even recent trends such as automatic code generation; Software as a Service (SaaS) and others, suffer from the same problems above



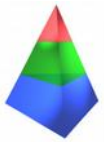
Automatic Code Generation

Advantages:

- Time Saving
- Repeatable
- Code (probably) Works

Disadvantages:

- Hard to maintain (e.g. a lot of unnecessary code lines)
- Low flexibility and complexity in customizations that need to be configured on the generator or edited in the resulting code to fit the needs
- Dependency on the code generator for new versions of the system, complexity in rollout



Software as a Service

Advantages:

- Time Saving
- Low Costs
- Scalable
- Easy to integrate with other SaaS's
- Upgradable
- Easy to use

Disadvantages:

- Applications focused in a specific field (e.g. invoicing, CRM, etc.)
- Low flexibility

Business Process as a Service

Advantages:

- Cost effective
- Scalable and flexible
- Standardized
- Specialized Staff while outsourcing business processes

Disadvantages:

- Dependent on external providers and other service stacks: SaaS, PaaS, IaaS
- Focus on multiple organizations/value chains
- Lack of flexibility

The communication-centric view: basics

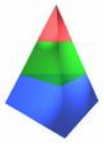
- **Communication** is the primal notion, defined as the *sharing of* (thoughts between) *human minds* – **thus a human and social centered construct**
- **Information** is the means for communication. It is the dyad of content (thought) and form (expression). In other words: *information is embodied thought*
- **Action** is either *production* related, both immaterial (deciding, judging) and material (fabricating, transporting), or *communication* related (requesting, promising, stating, accepting, etc.)
- **Organisation** emerges from communication, both in the operational sense (the cooperating people) and in the constructional sense (the devising of cooperation structures): *communication is the thread of which organisation is woven*

The communication-centric view: effects

- The focal point of the information system engineer, now turned into *enterprise engineer*, is the enterprise
- The task of the enterprise engineer is to develop and install a new implementation of (a part of) the *organisation* of an enterprise
- The development process is roughly: *producing the ontological model of the organisation*, *devising a new implementation model, implementing*
- The *functional requirements* are basically *determined by the ontological model* of the organisation

The communication-centric view: outcomes

- An information system is some implementation (probably using ICT) of (a specific part of) an organisation
- Business processes become simple tree structures of transactions, which are generic patterns of human cooperation
- Organisations may themselves be subject to redesign. This comes down to devising a new ontological model, and to properly implement it
- Because the ontological model of an organisation is fully formalisable, **automatic generation of ICT applications is a realistic option**



Our Vision: Enterprise Modeling and Execution as a Service (EMEaaS)

EMEaaS

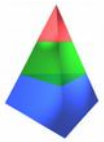
- Any worker can design enterprise models based on the CIAO! Paradigm
- No need of programming knowledge
- Designed models can be executed instantly
- Pre-designed models available, fully customizable
- Interfaces automatically generated based on model elements
- Service provided locally or on the cloud

The communication-centric view: summary

COMMUN
information
organisation
action
ICATION

communication is the thread of which organisation is woven

Enterprise Engineering



The mission of Enterprise Engineering

Addressing the challenges mentioned before requires a paradigm shift.

It is the mission of the discipline of Enterprise Engineering to **develop new, appropriate theories, models, methods and other artifacts for the analysis, design, implementation, and governance of enterprises** by combining (relevant parts of) management and organization science, information systems science, and computer science.

The ambition is to address (all) traditional topics in said disciplines from the Enterprise Engineering Paradigm.

In addition, the results of our efforts should be ***theoretically rigorous*** and ***practically adequate***

Theoretical foundations of EE

THEORY CLASS	INSPIRATIONAL SOURCES	EE-THEORY
Ideological *devising and choosing things to make* ethical, political, etc. ideas	W.E. Deming, P. Drucker R. Likert, D. McGregor, D. Katz & R.L. Kahn J.M. Burns	σ-theory
Technological *designing and implementing things* analysis and synthesis	C. Alexander, H. Simon, L. von Bertalanffy, P. Checkland, E.W. Dijkstra, M.D. McIlroy	β-theory ν-theory
Ontological *understanding the nature of things* explanation and prediction	J. Austin, J. Searle, J. Habermas, M. Bunge, P. Checkland, B. Langefors J.R. Taylor & E.J. Van Every K.Z. Lewin	ψ-theory
Philosophical *theoretical foundations* epistemology, mathematics, phenomenology, logic	C.S. Peirce, C.W. Morris, M. Bunge, L. Wittgenstein, J.F. Sowa, P. Simons M. Heidegger, K.H. Marx	φ-theory δ-theory τ-theory

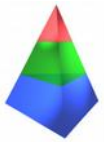
The ψ-theory: organisation

ψ (PSI) stands for Performance in Social Interaction. Primarily rooted in Habermas' social theory and Bunge's systemic ontology.

- The operating principle of organisations is that *human beings* enter into and comply with *commitments* regarding the production of things. They do so in *communication*, and against a shared background of cultural norms and values.

- Commitments occur in processes that follow the *universal transaction pattern*. This is a structure of *coordination acts/facts* between two actors, concerning one *production act/fact*. One is the *initiator* (consumer) and the other one the *executor* (producer).

- An organisation is a network of actors and transactions. Every actor has a particular *authority*, assigned on the basis of *competence*. Actors are assumed to exercise their authority with *responsibility*. They operate *autonomously*.



Examples of coordination acts

Alicia: *I'd like to have a bouquet of red tulips*

Alicia : request : Celestine : order 387 is fulfilled

Celestine: *Just a moment*

Celestine : promise : Alicia : order 387 is fulfilled

proposition

Celestine: *Here you are*

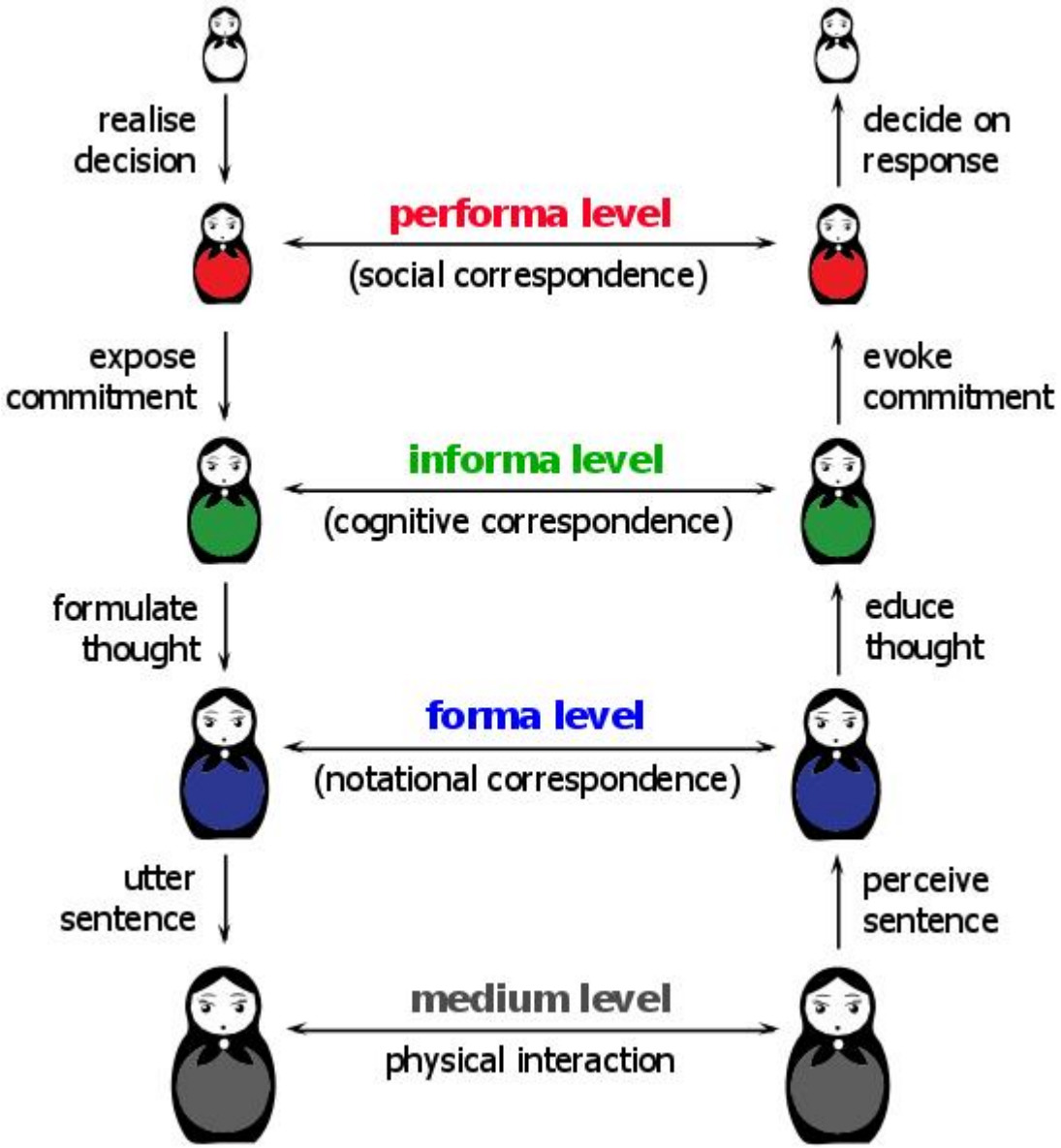
Celestine : state : Alicia : order 387 is fulfilled

Alicia: *Thanks*

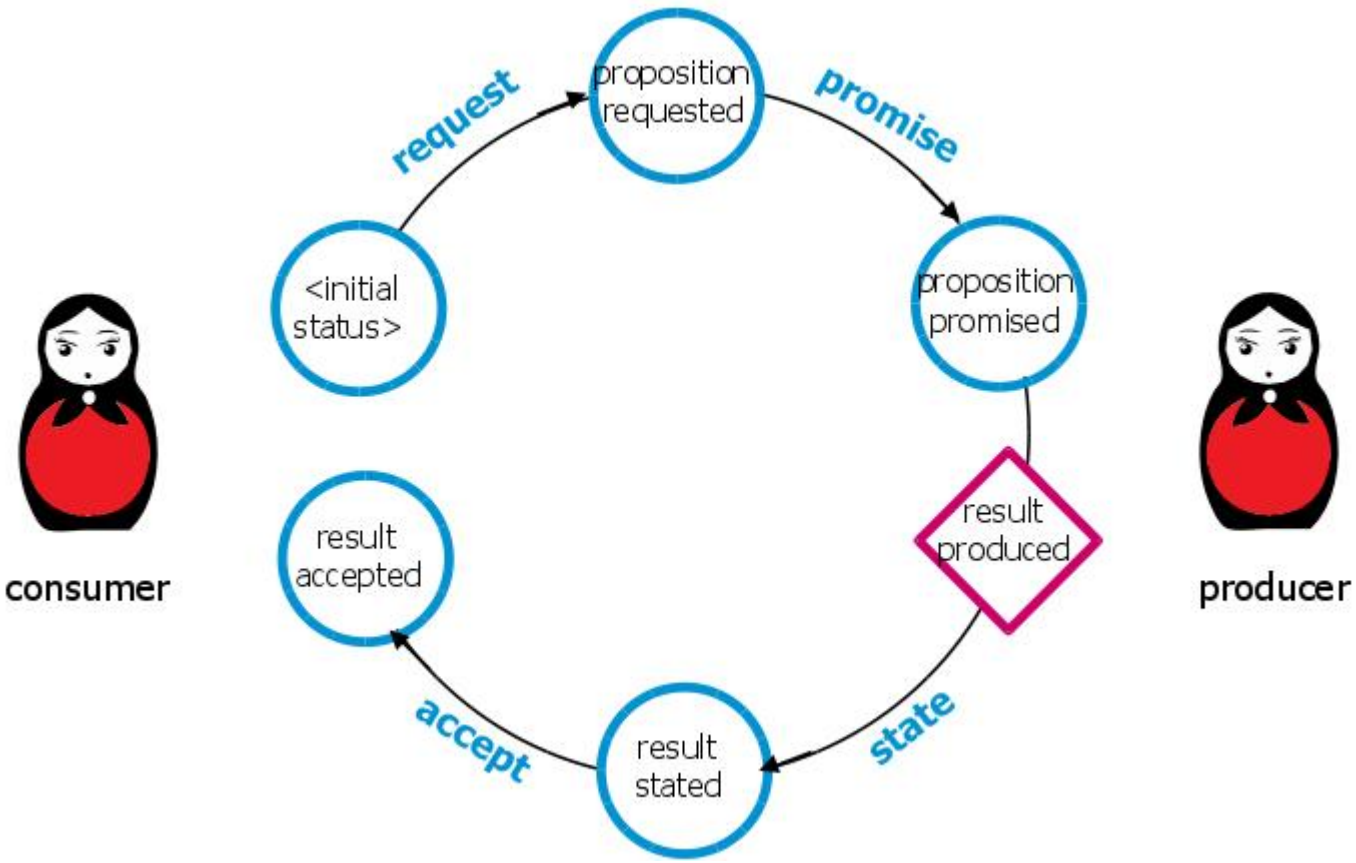
Alicia : accept : Celestine : order 387 is fulfilled

result

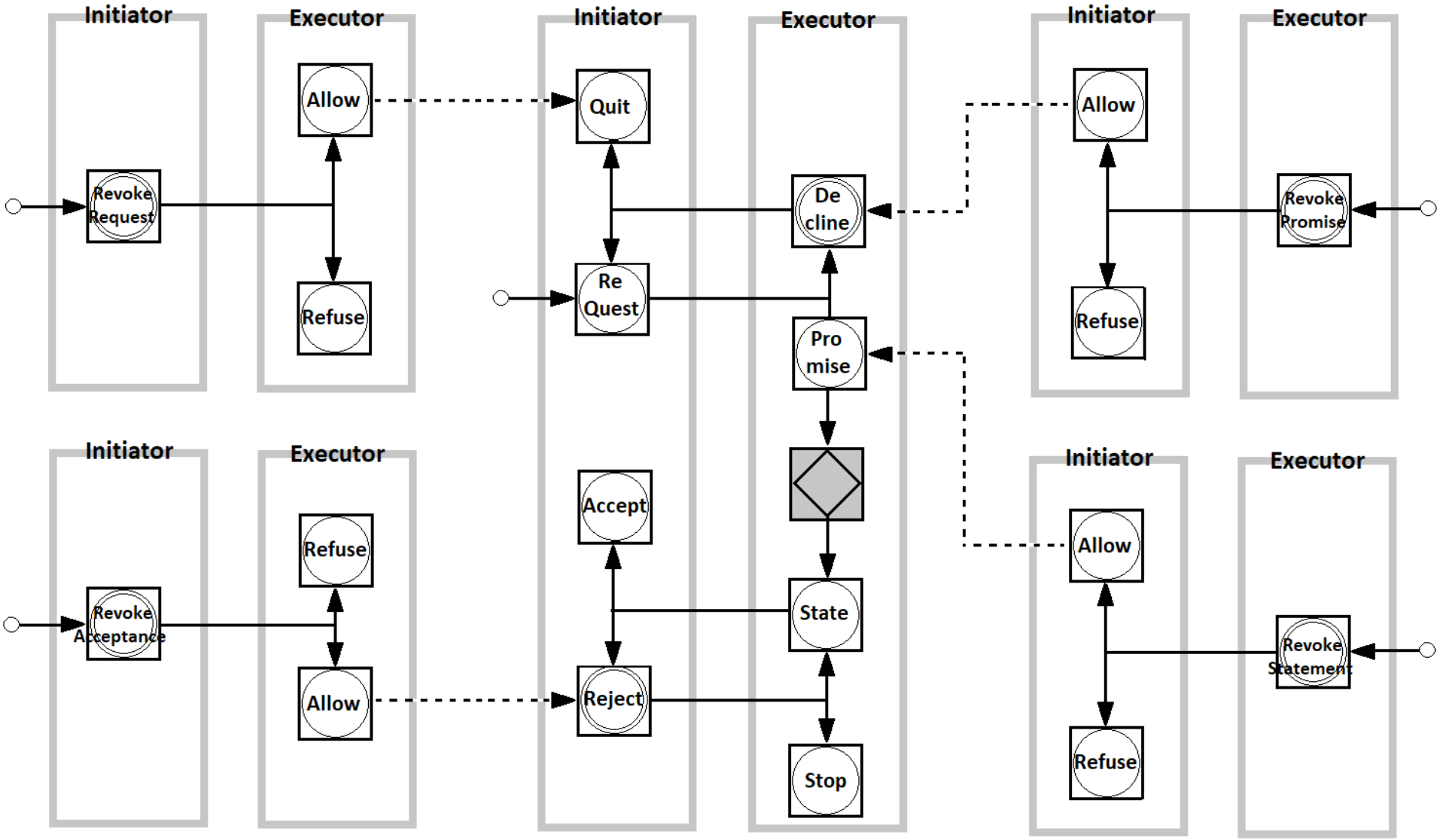
The ψ-theory: coordination



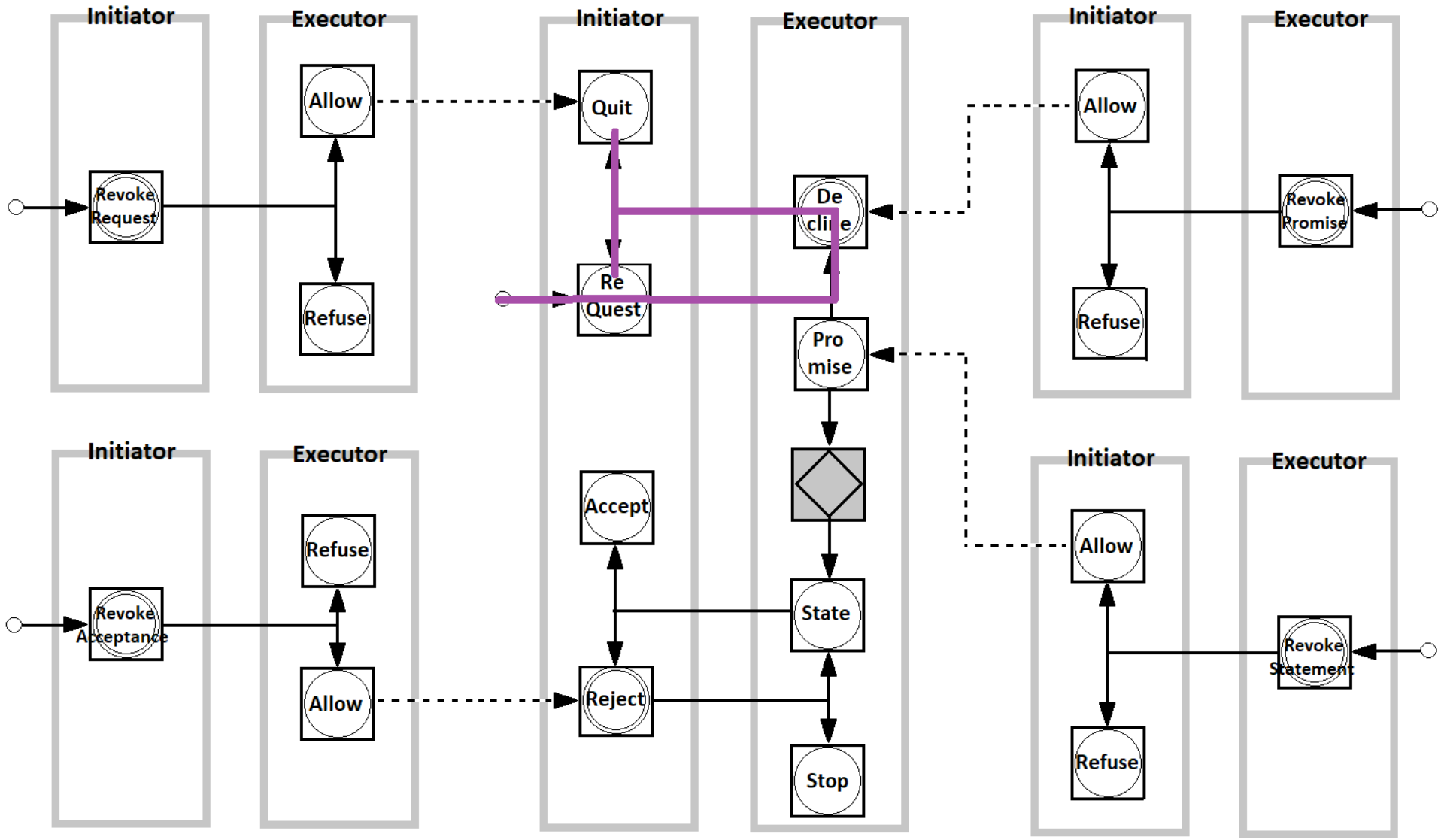
The basic transaction pattern



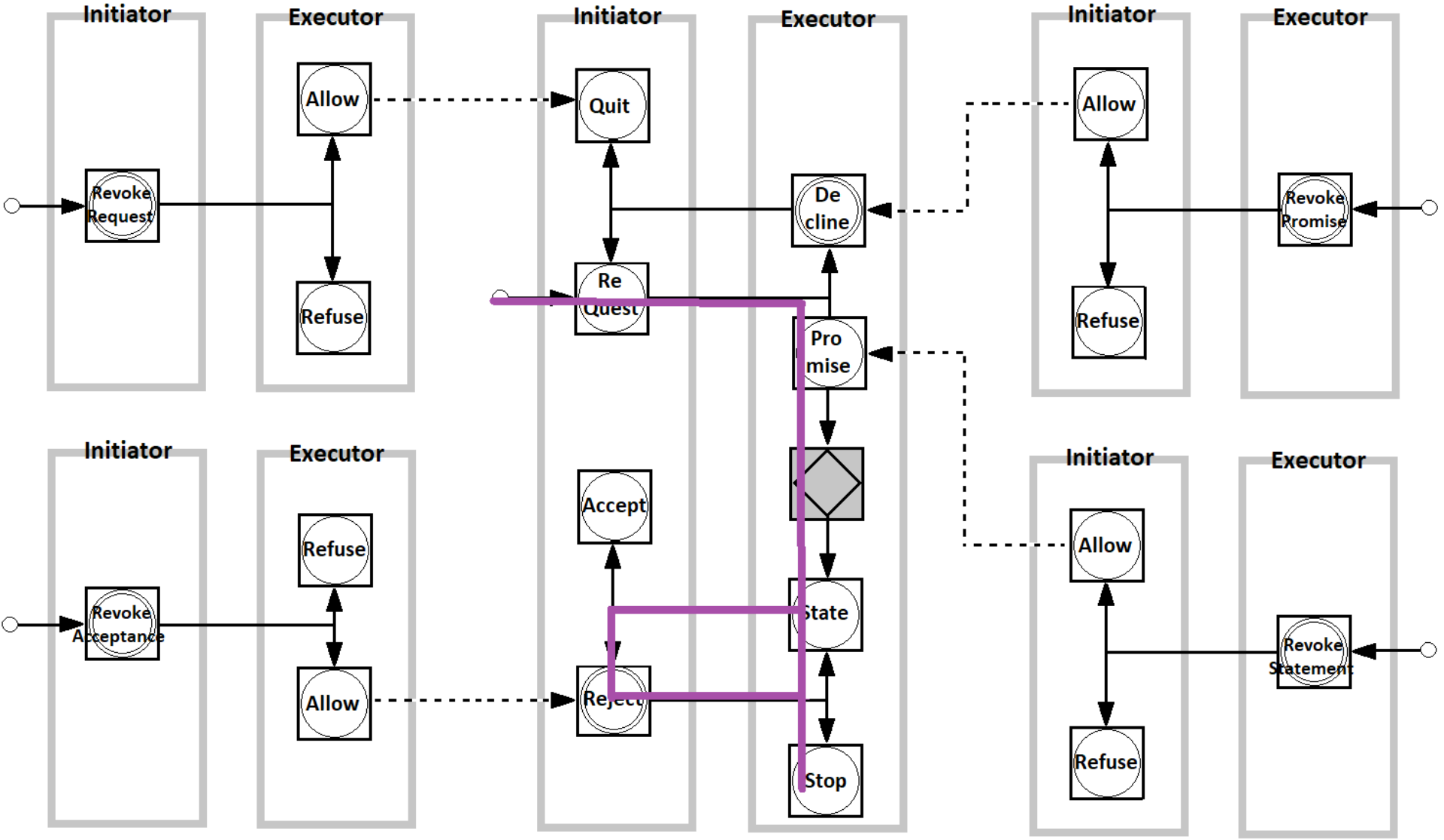
The complete transaction pattern (1)



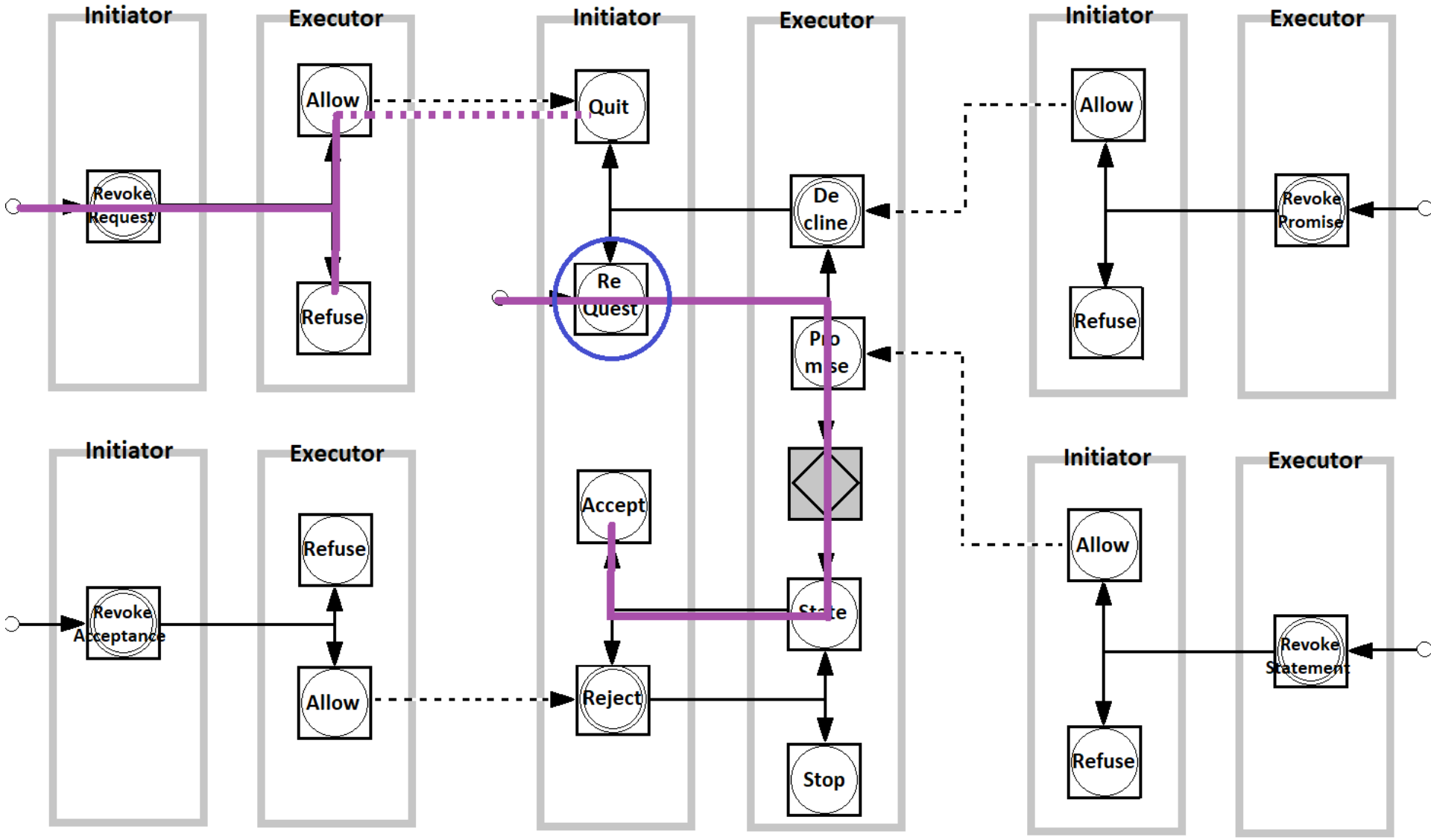
The complete transaction pattern (2)



The complete transaction pattern (3)

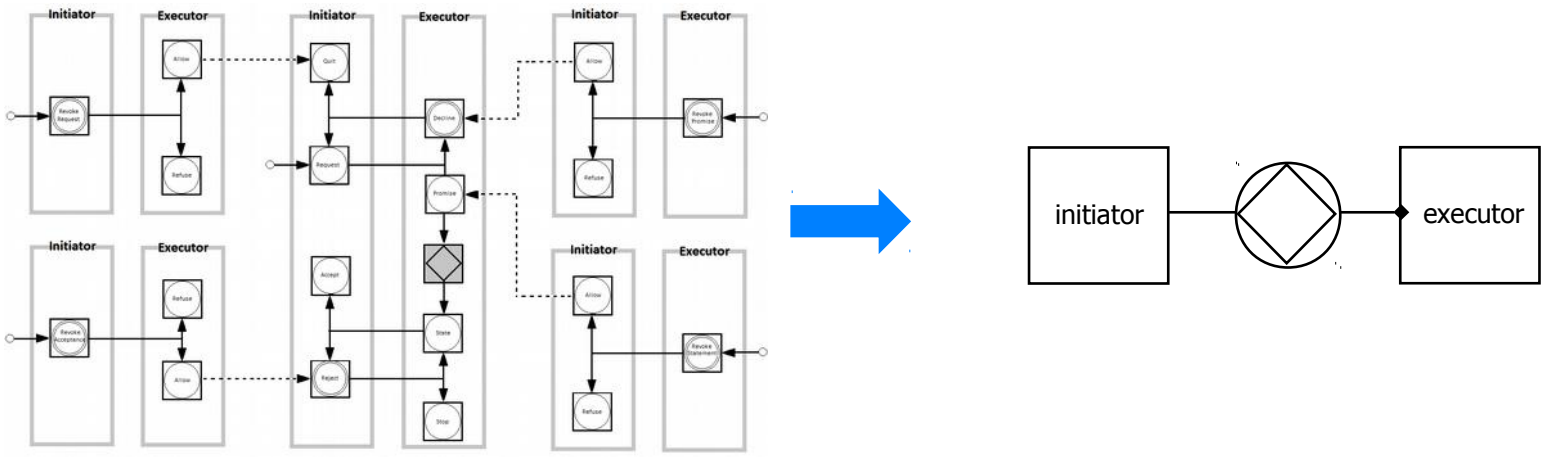


The complete transaction pattern (4)



The organisational building block

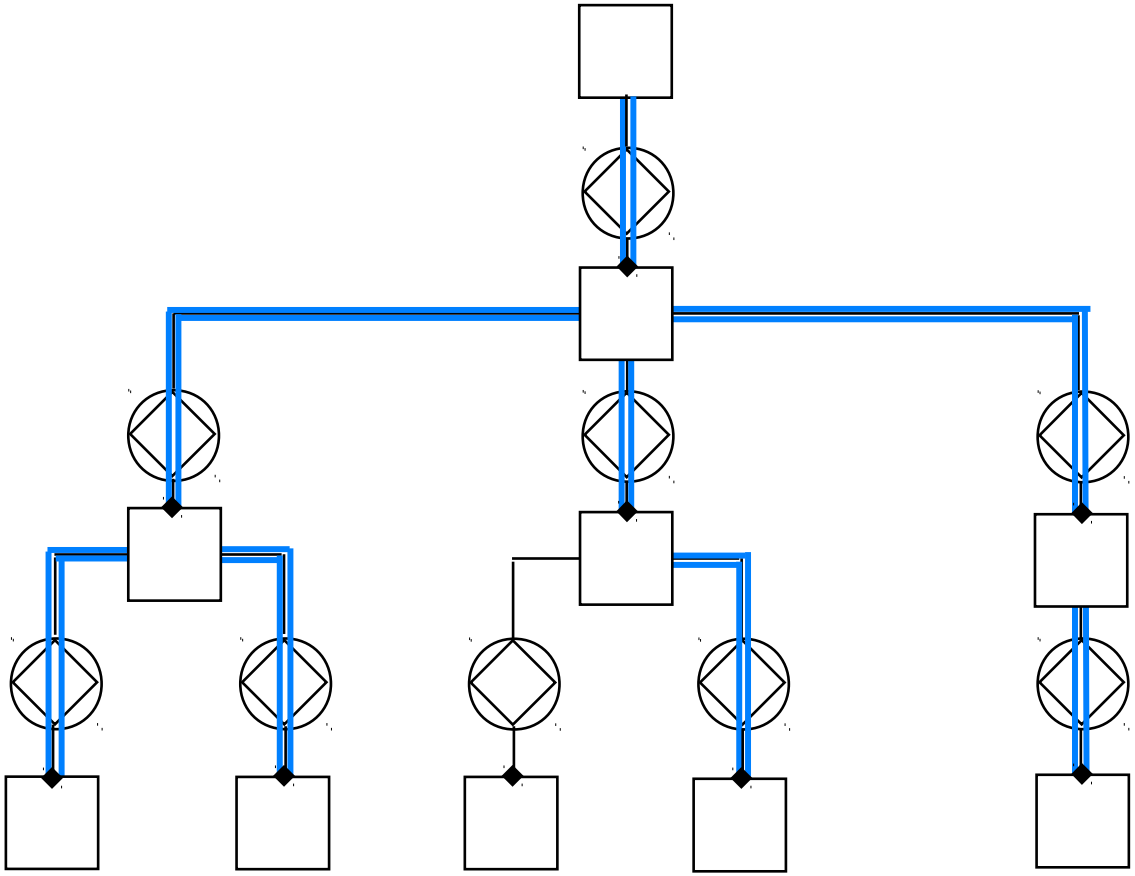
Every (elementary) *actor role* is the executor of exactly one transaction kind, and initiator of 0, 1 or more transaction kinds. An *actor* is a person in fulfilling an actor role.



Next to the *process* interpretation of the transaction symbol, there is the *state* interpretation:

it is the conceptual container of all coordination facts that are created in all transactions up to now. In the state interpretation, the transaction symbol is called a transaction bank.

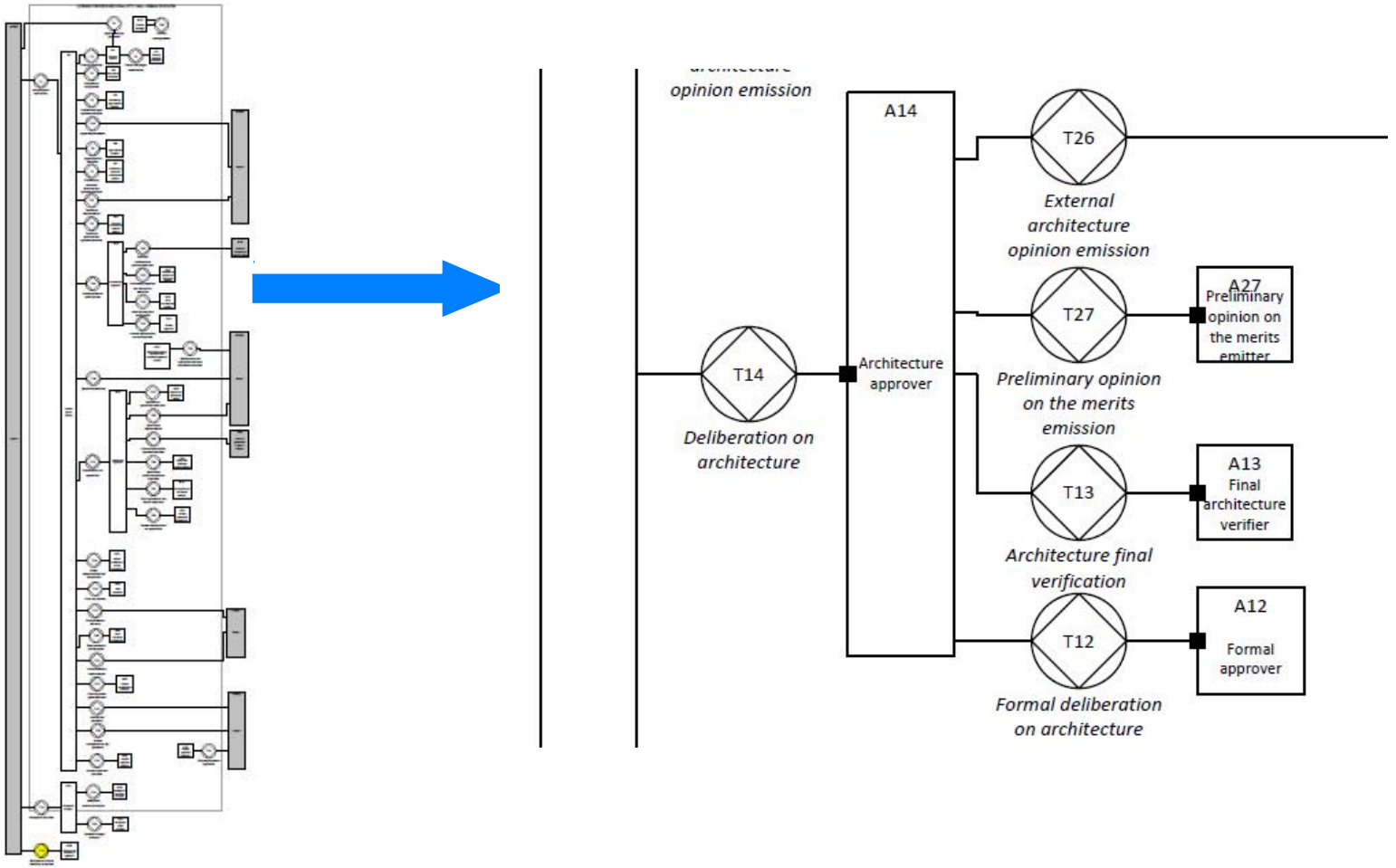
A business process is a tree of transactions



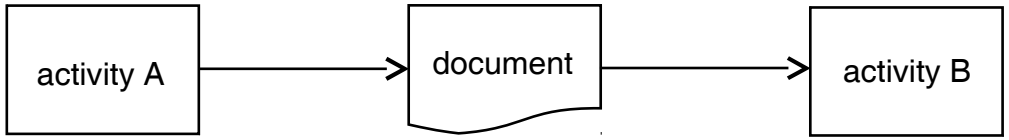
Note. Component transactions may also be carried out in parallel

A business process is a tree of transactions

- City hall project licencing process case
  - ~ **50 A4 pages** - flowcharts with hundreds of tasks abstracted to:
  - = **2 A3 pages** - 36 transactions



The ambiguity of process modeling



Is passing the document from A to B:

Only a **datalogical** act?

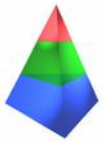
Example: A hands over the document to B to archive it.

Or (also) an **infological** act?

Example: A informs B about the content of the document.

Or (also) an **ontological** act?

Example: A requests B to do something.



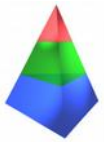
Violation of the ψ-theory

Current business process modeling approaches, like Flowchart, BPMN, EPC, and Petri Net *reduce* business processes to sequences of (observable) actions and results.

Thereby loosing the *essential deep structure* (which is always a tree of transactions) and neglecting all *tacitly* performed coordination acts.

Therefore they are ambiguous (if not dangerous) for business process re-design and re-engineering.

Even worse are the function-oriented techniques (SADT, IDEF0) since by definition they reflect the personal interpretation of the modeler (black-box model)!



The ψ-theory: production

The three human *abilities* also apply to *production*:

Performa
The ability to perform ***original*** production acts, such as to *create (manufacture, transport, observe), decide, judge*.

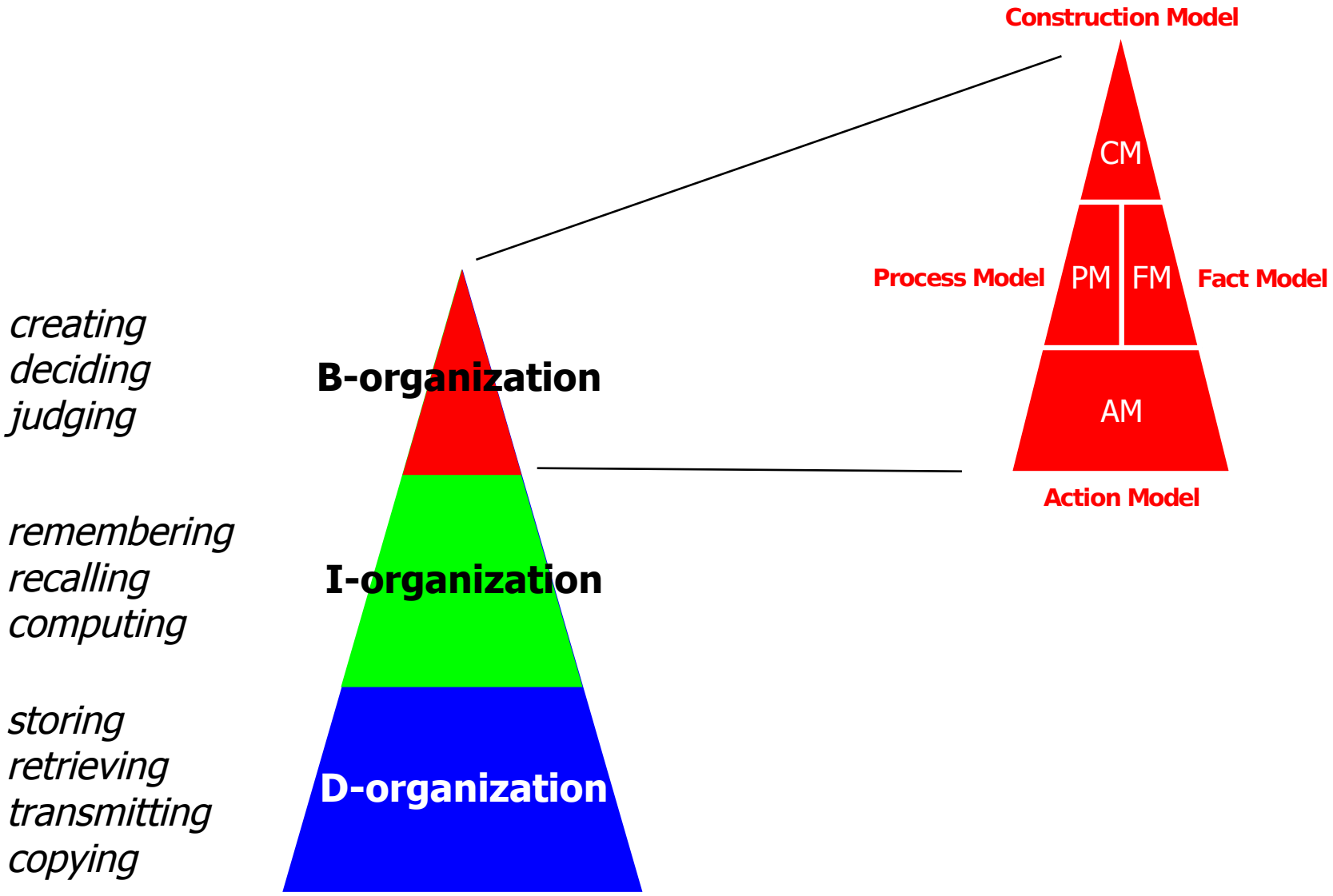
Informa
The ability to perform ***informational*** production acts, such as to *remember, recall, compute* (facts)

Forma
The ability to perform ***documental*** production acts, such as to *store, retrieve, transmit, copy* (sentences, documents).

The ψ-theory: summary

COORDINATION	HUMAN ABILITY	PRODUCTION
exposing commitment evoking commitment	 **performa**	original acts/facts (creating, deciding, judging)
formulating thought educing thought	 **informa**	informational acts/facts (remembering, recalling, computing)
uttering sentence perceiving sentence	 **forma**	documental acts/facts (storing, retrieving, transmitting)

The three aspect organisations



The essential model of an enterprise (1)

In the huge network of organisations (interacting social individuals), we make the next selections and abstractions:

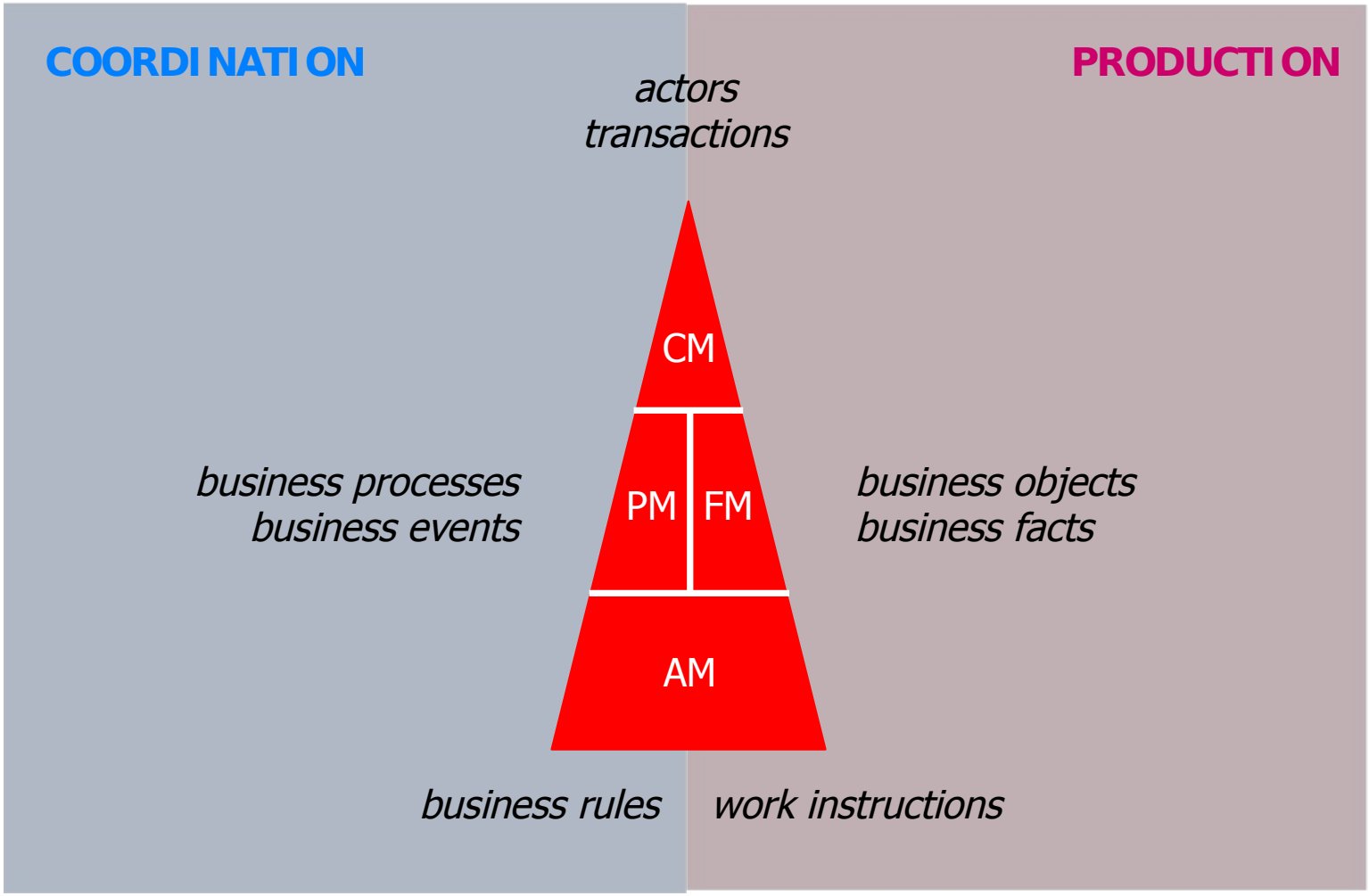
1. We select our scope of interest, so we see only (the part of) the enterprise's *organisation* we want to investigate.

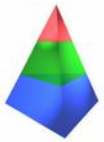
2. We put the building block 'template' on the organisation, so we see a network of *transaction kinds* and connected *actor roles*.

3. We only consider the **performa** level of coordination: we have got the *ontological model* of the enterprise's organisation.

4. We leave out the D-organisation and the I-organisation network: we have got the *ontological model* of the **B-organisation** of the enterprise, which is **the essential model of the enterprise**: concise, coherent, consistent, and comprehensive.

The essential model of an enterprise (2)



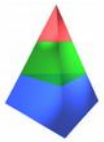


Benefits of Enterprise Engineering

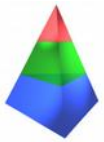
- Re-establishing people as the 'pearls' of your organization
- Unequaled deep and coherent insight in your organization.
- Service-oriented analysis and design of your business processes.
- Full transparency of your (service-based) organization.
- Clear identification of data and process ownership
- Truly objective basis for requirements engineering.
- Unequaled reduction of model complexity (> 95%).
- Unequaled return on modeling effort (5-10 times higher).

Who needs Enterprise Engineering?

- **Managers** - need to understand the ontological essence of their enterprise because they are held accountable.
- **Architects** - need to understand organizations (and information systems) abstracted from their implementation, for making the right design decisions.
- **Employees** - only the ontology of an enterprise shows the roles they really fulfill, and the relationships with others that really exist.
- **Customers** - why should the operation of an enterprise be fully opaque to its customers? Enterprise Ontology provides them the transparency they deserve!



Realizing our EMEaaS vision with the Dynamic Information System Modeller and Executer (DISME)



DISME Overview

Main functionalities:

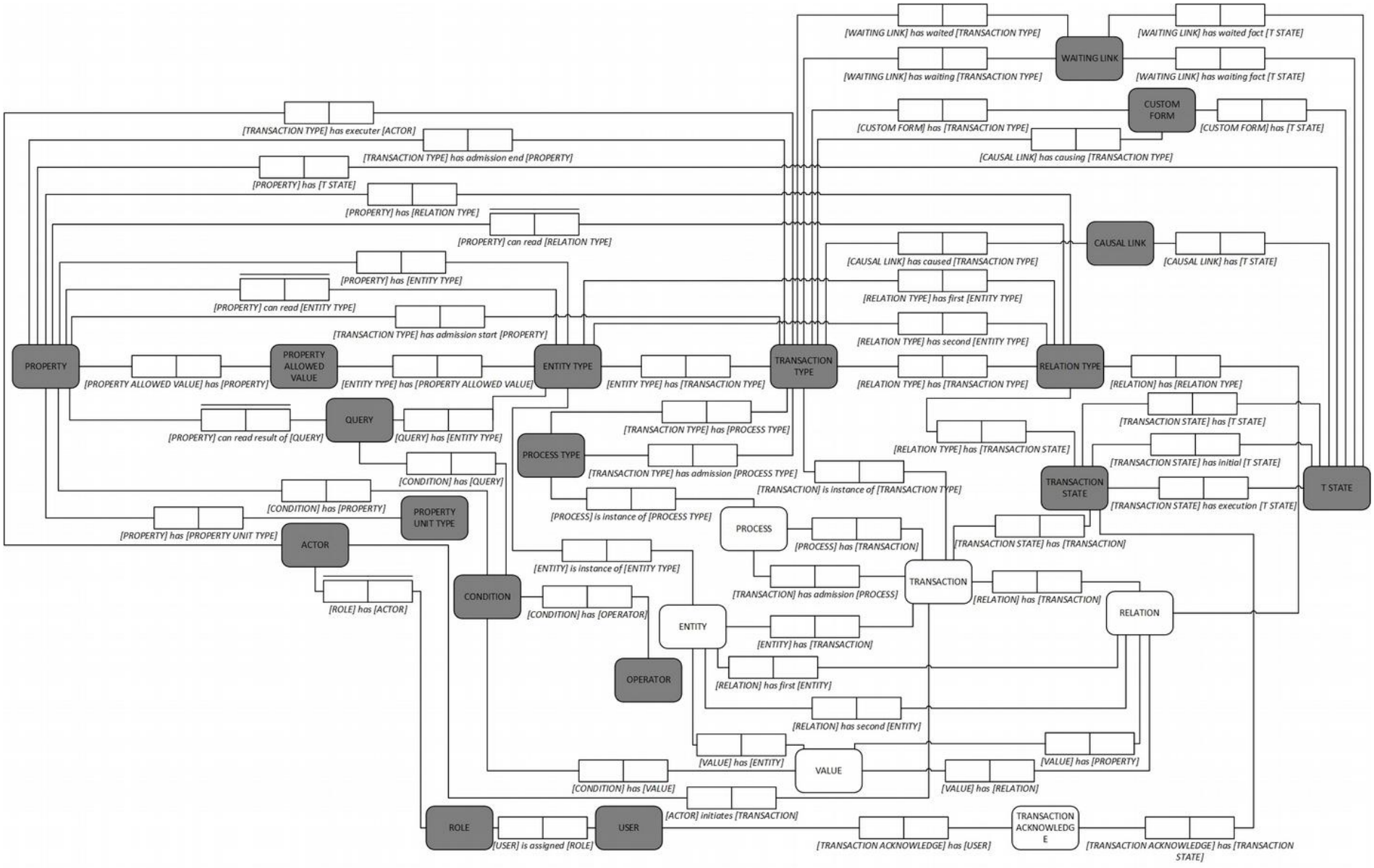
- Diagram Editor
- System Modeler
- System Execution

Conceptual model and prototype development evolving with collaboration with the international EE research network CIAO! and with application in local collaboration with private and public sectors:

- Logistics Company
- Municipality

No programming skill required; basic knowledge of enterprise engineering modeling is sufficient

DISME Conceptual Model (simplified)



DISME Main components:

Diagram Editor – SVG editor based on GraphEditor allowing:

- Designing of new process and data types in a user-friendly way
- Visual presentation of models directly specified in the System Modeling component

System Modeling – modeling of the enterprise's information system by use of user-friendly forms that allow the specification of processes, associated transactions, data/entity types, relationship types, properties, roles, actors, users, authorizations, queries and many other parameters

System Execution – by means of a Dashboard, logged in users can initiate or execute transactions according to the specified permissions, following DEMO's transaction pattern leading to the creation of new fact instances in the system and/or interaction/communication with other systems; they can also dynamically create and execute queries in a user-friendly way

Diagram Editor: Actor Transaction Diagram

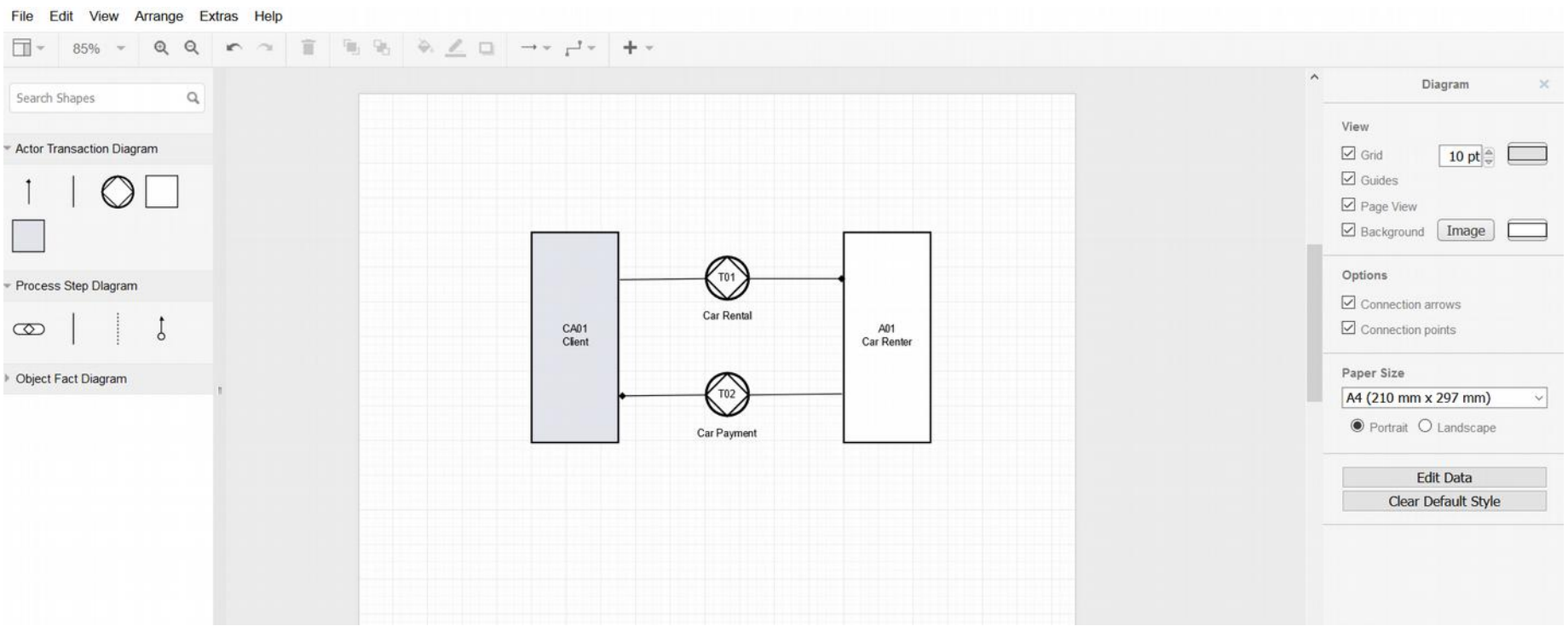


Diagram Editor: Process Structure Diagram

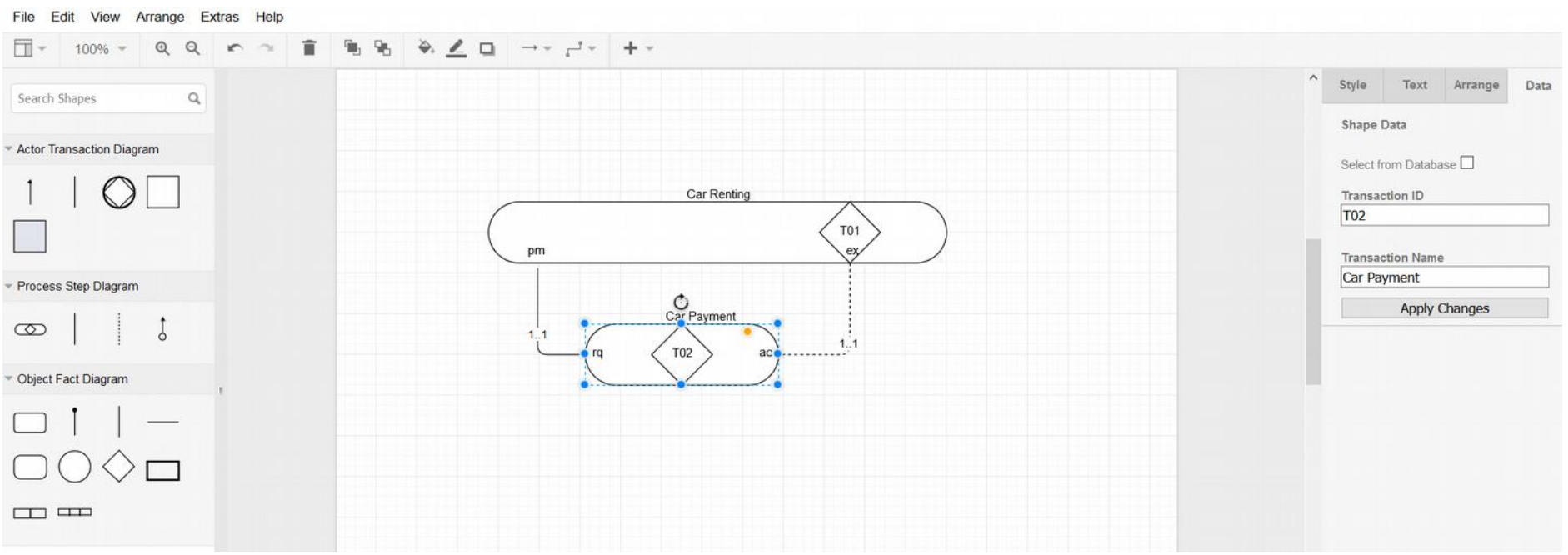
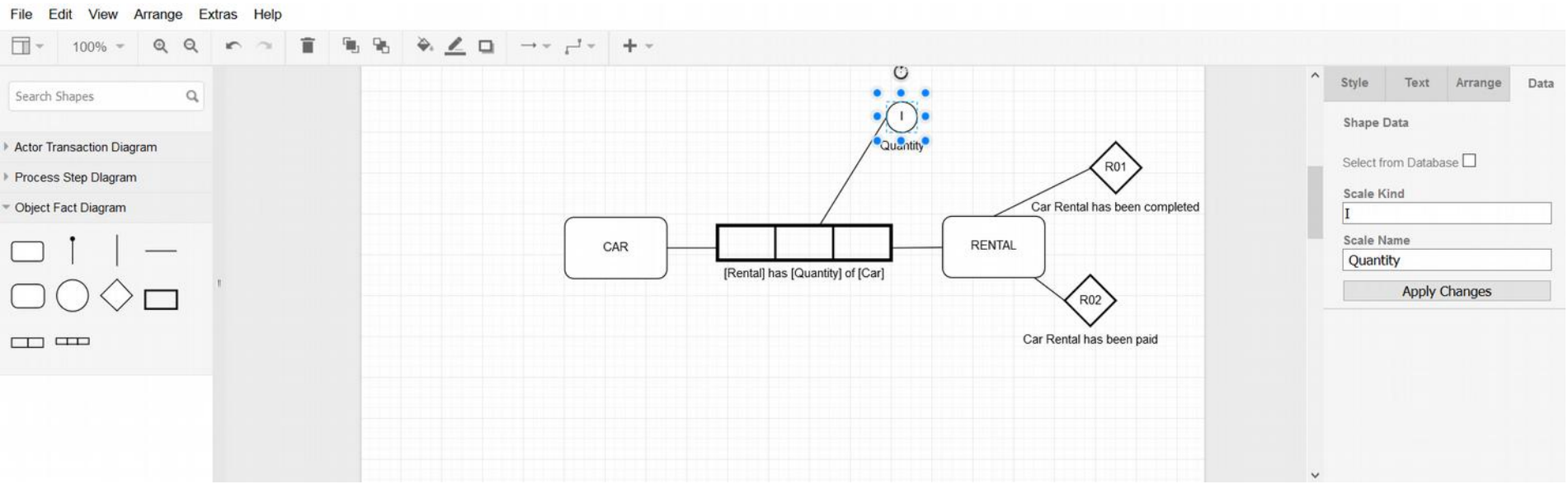
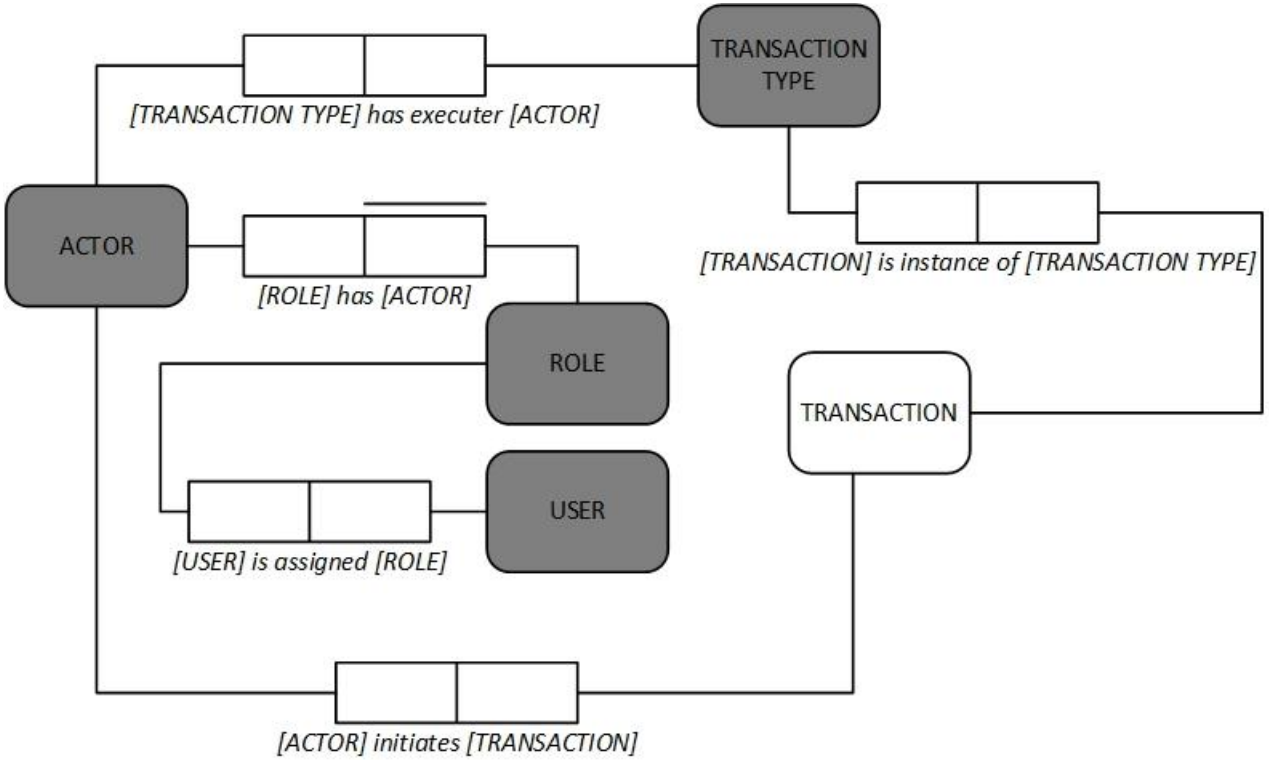


Diagram Editor: Object Fact Diagram

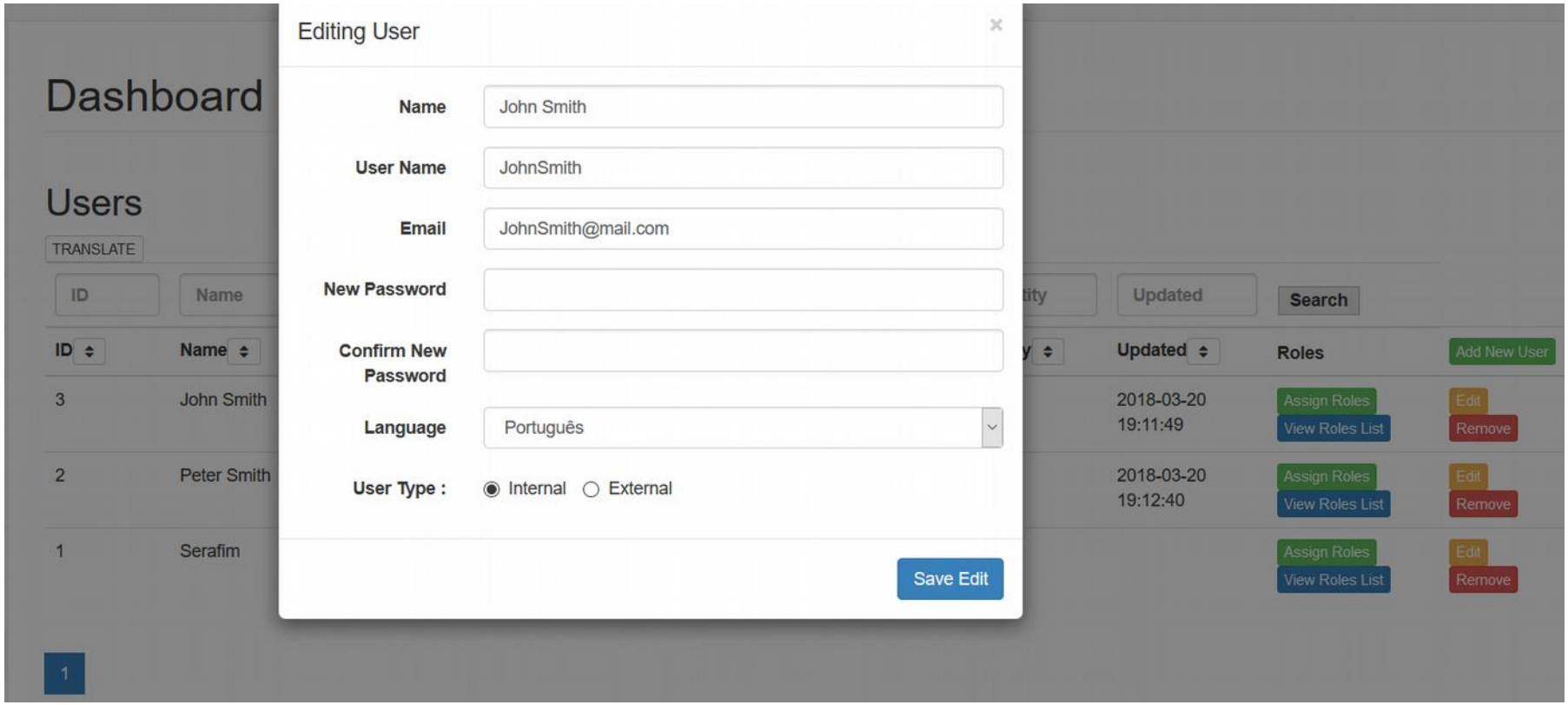


System Modeling Functions



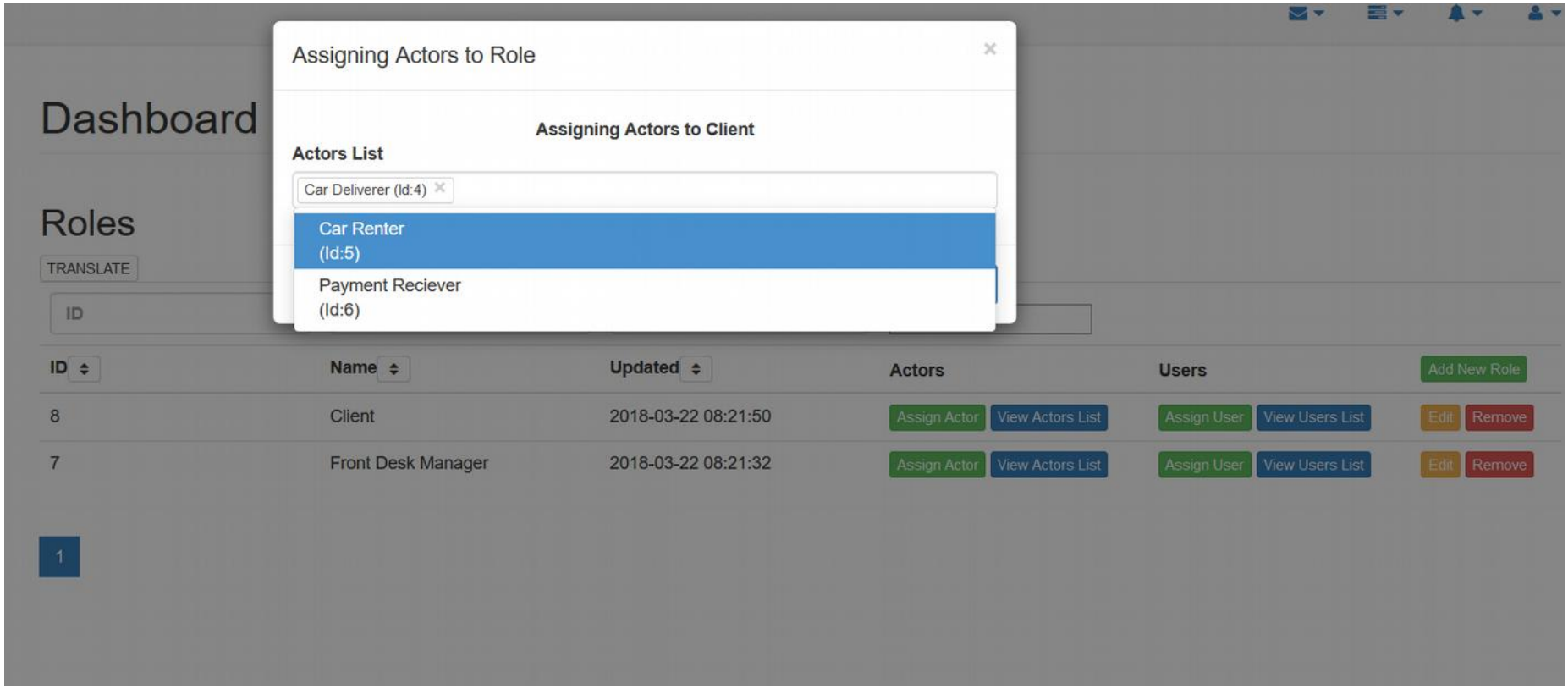
System Modeling Functions

User Management – creation of new users and/or editing their data



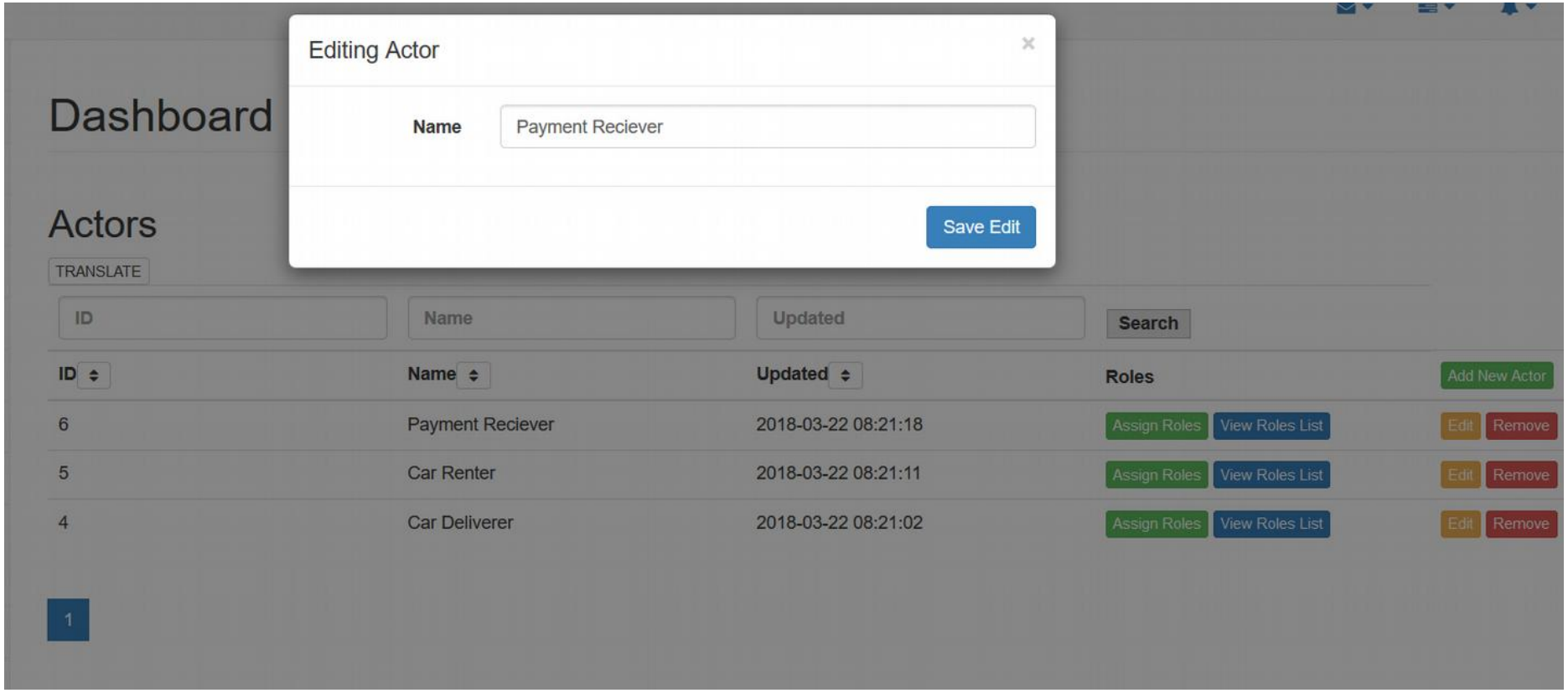
System Modeling Functions

Role Management – specification and association of roles with actors and users; one role can fulfil various actors and an actor can be fulfilled by several roles.

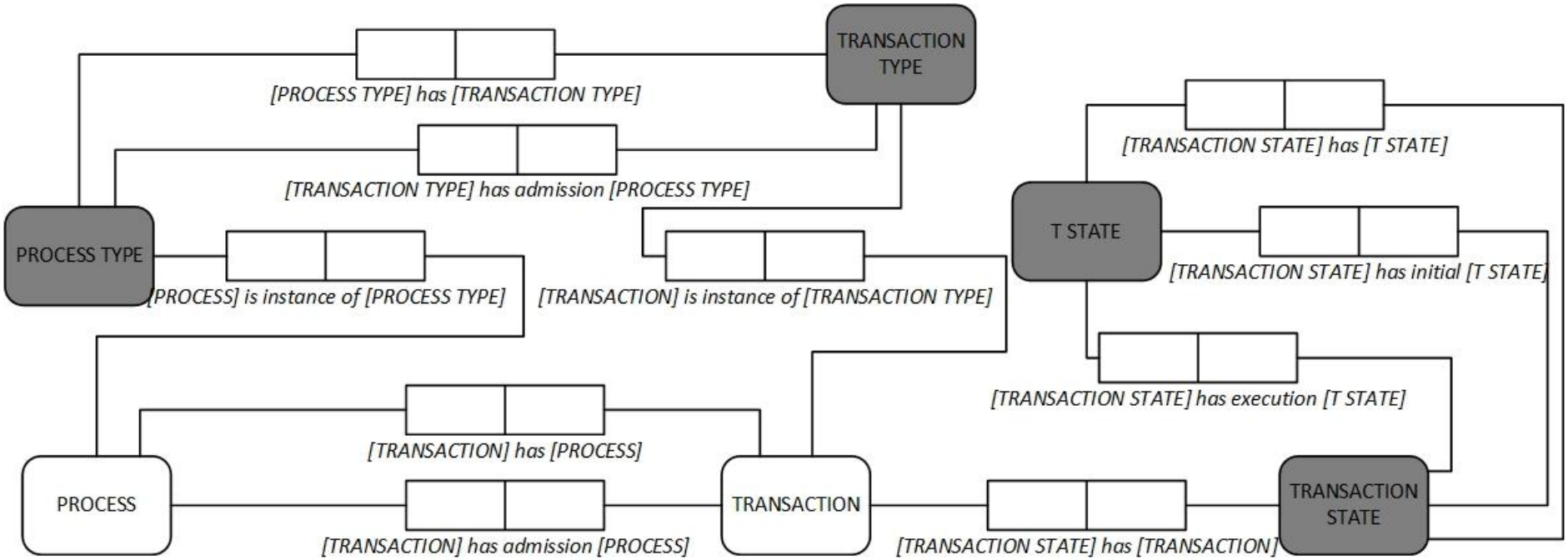


System Modeling Functions

Actor Management – specification of the actors responsible for initiating and executing transactions. An actor may be associated with several organizational roles.

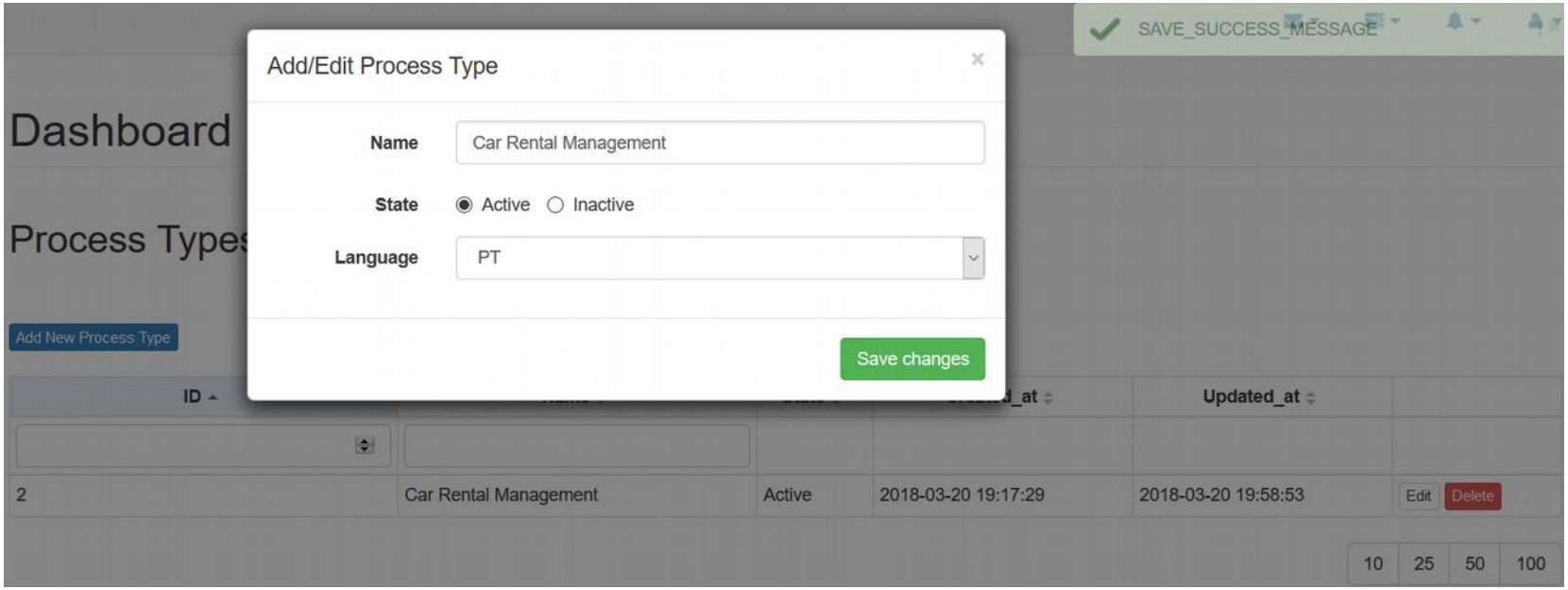


System Modeling Functions



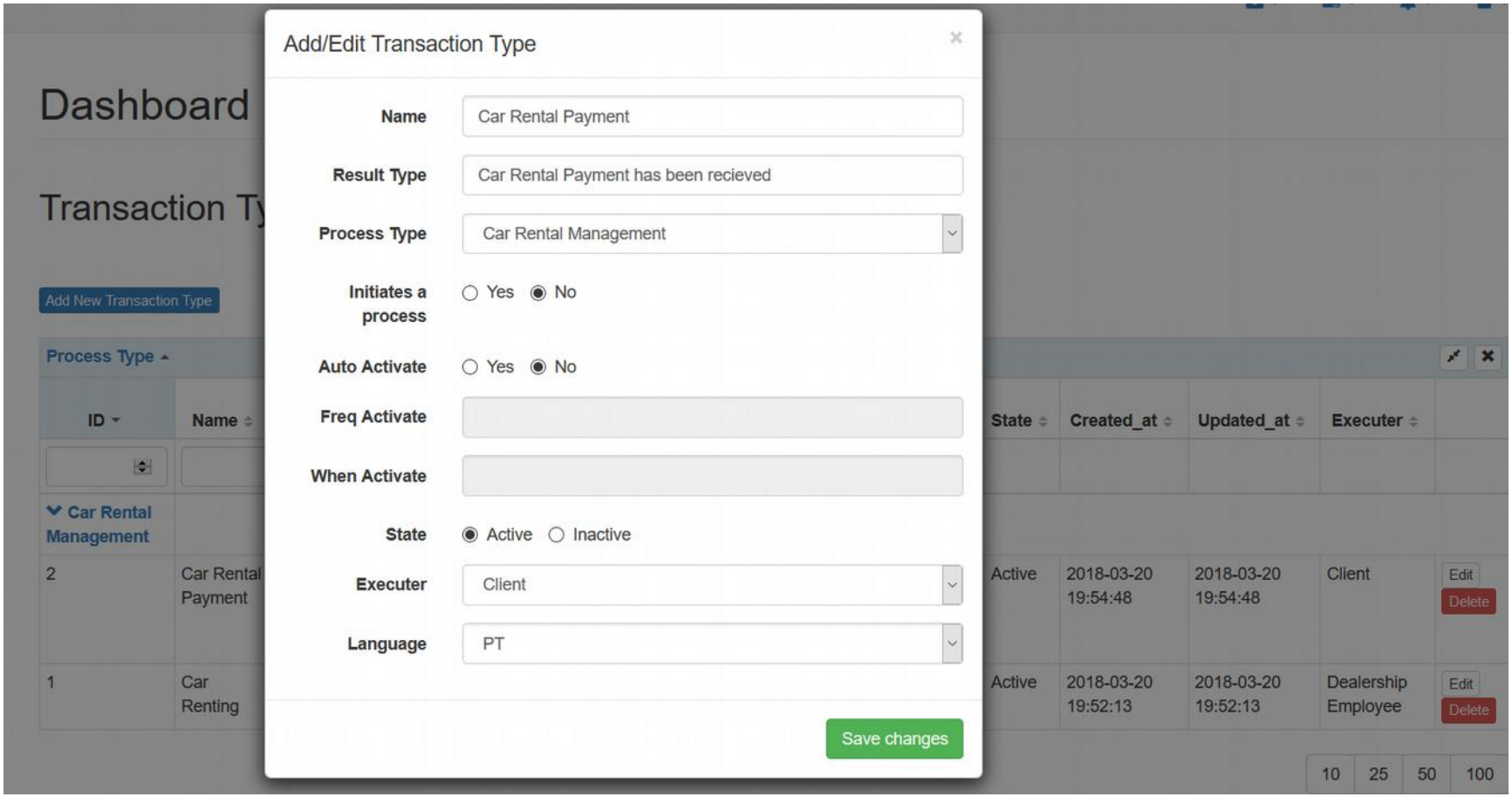
System Modeling Functions

Process Management – specification of process types

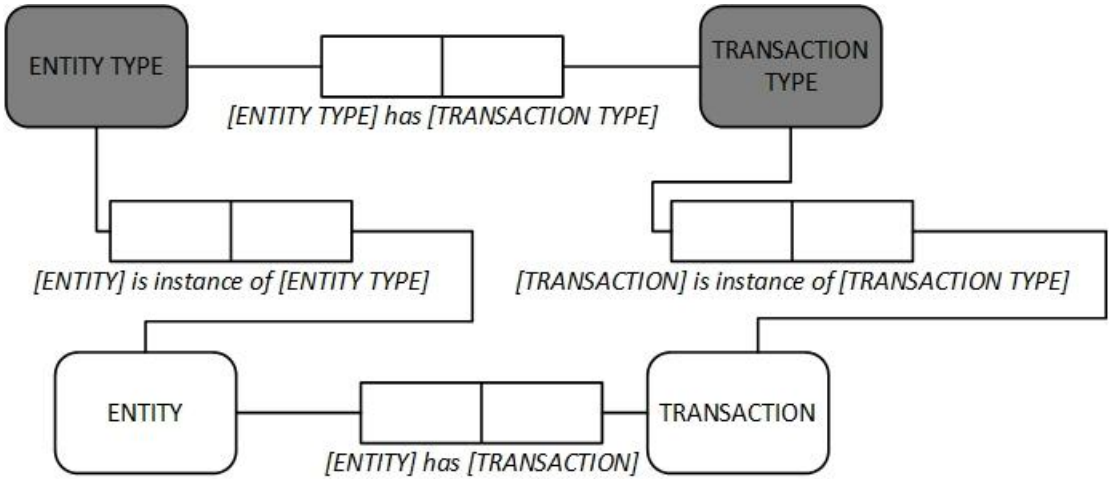


System Modeling Functions

Transaction Management – specification of transaction types, always associated with a process type and an executor



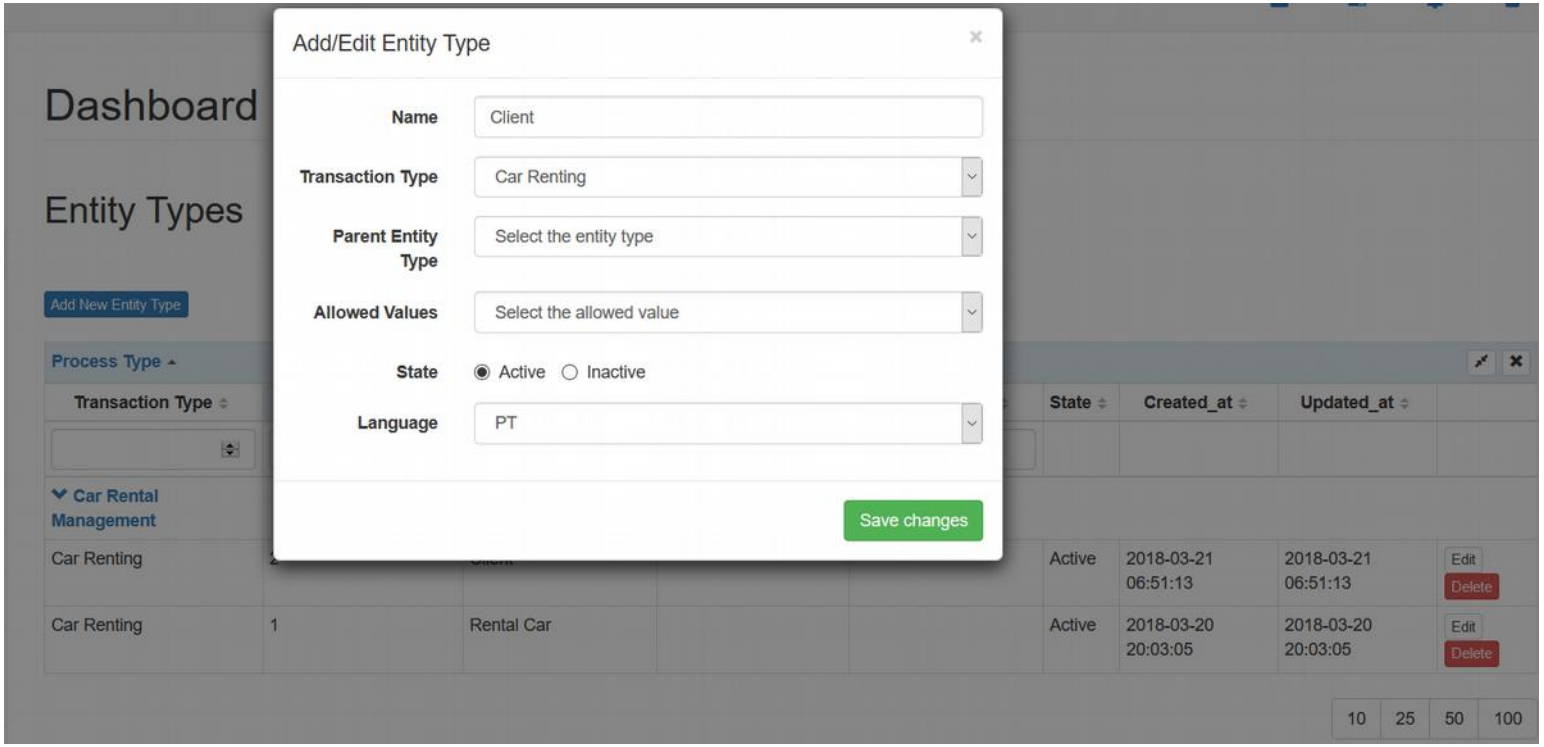
System Modeling Functions



System Modeling Functions

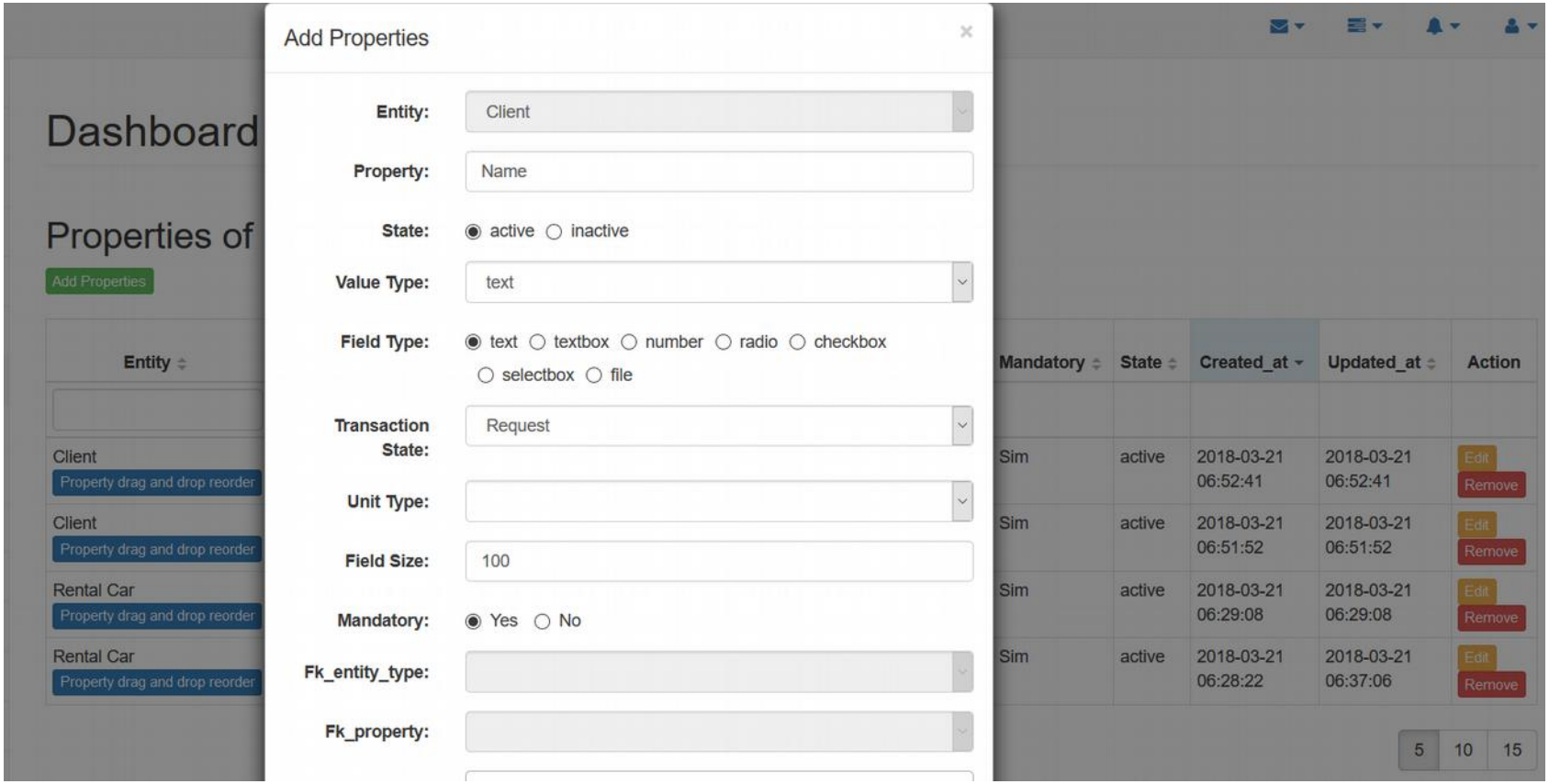
Entity Management – specification of the business entity types

- Comparable to the definition of the table in a database that will be responsible for saving the corresponding records / data.
- An entity type corresponds to an OFD class defined in the diagram editor



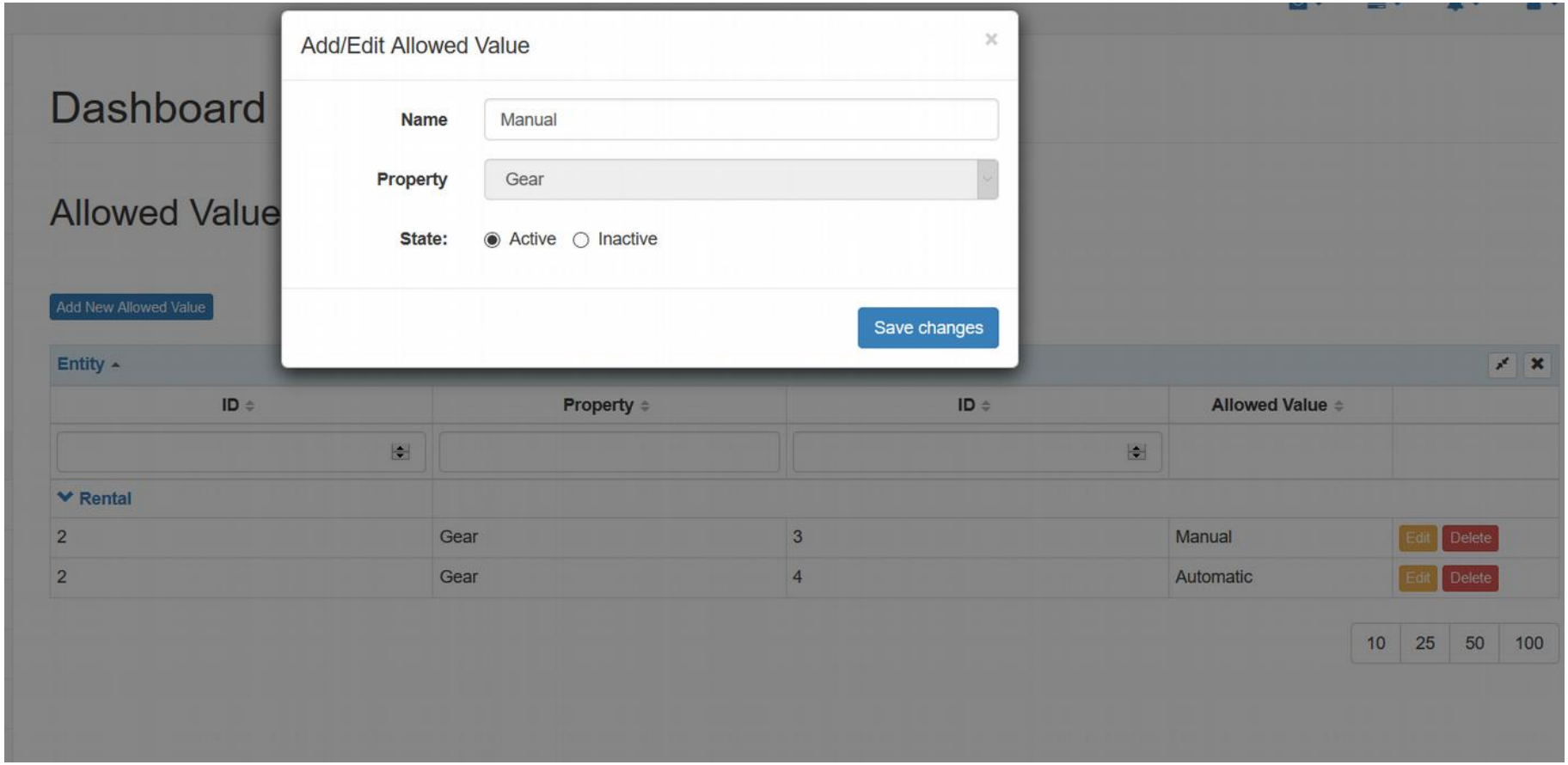
System Modeling Functions

Property Management – specification of property types to be associated to an entity type or relationship type, namely specifying its name, value type (text, int, enum, etc.) and field type (to be output in the automatically generated forms of the interface), etc.



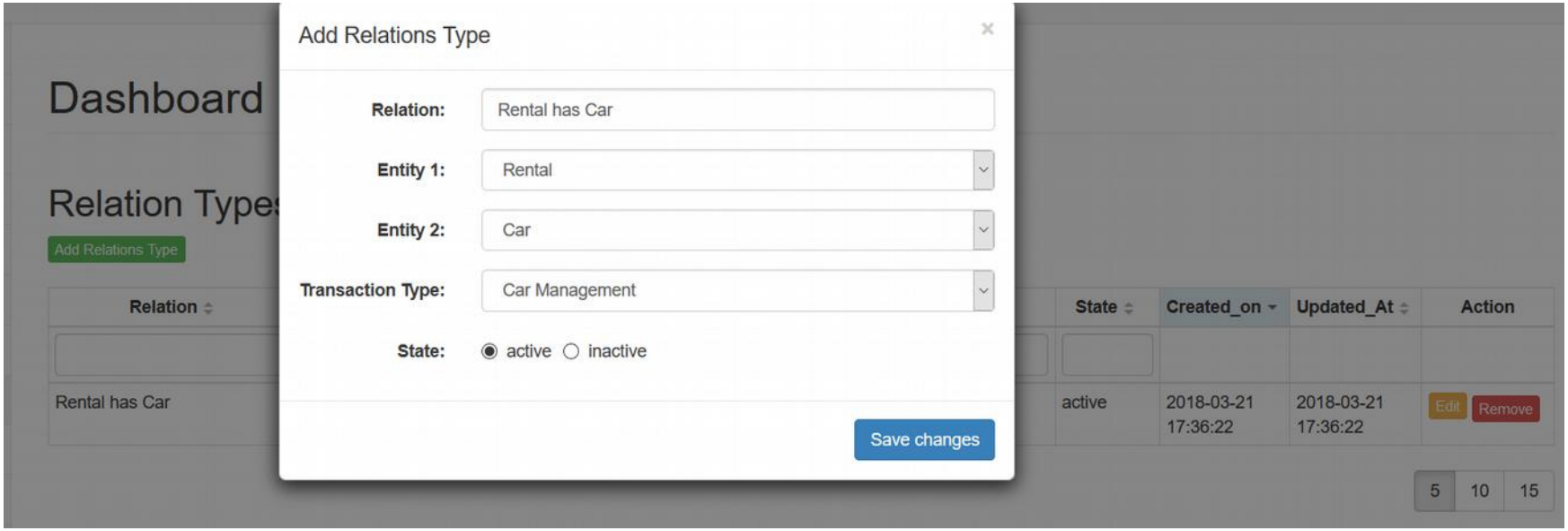
System Modeling Functions

Allowed Value Management – specification of values allowed for a property of type ENUM



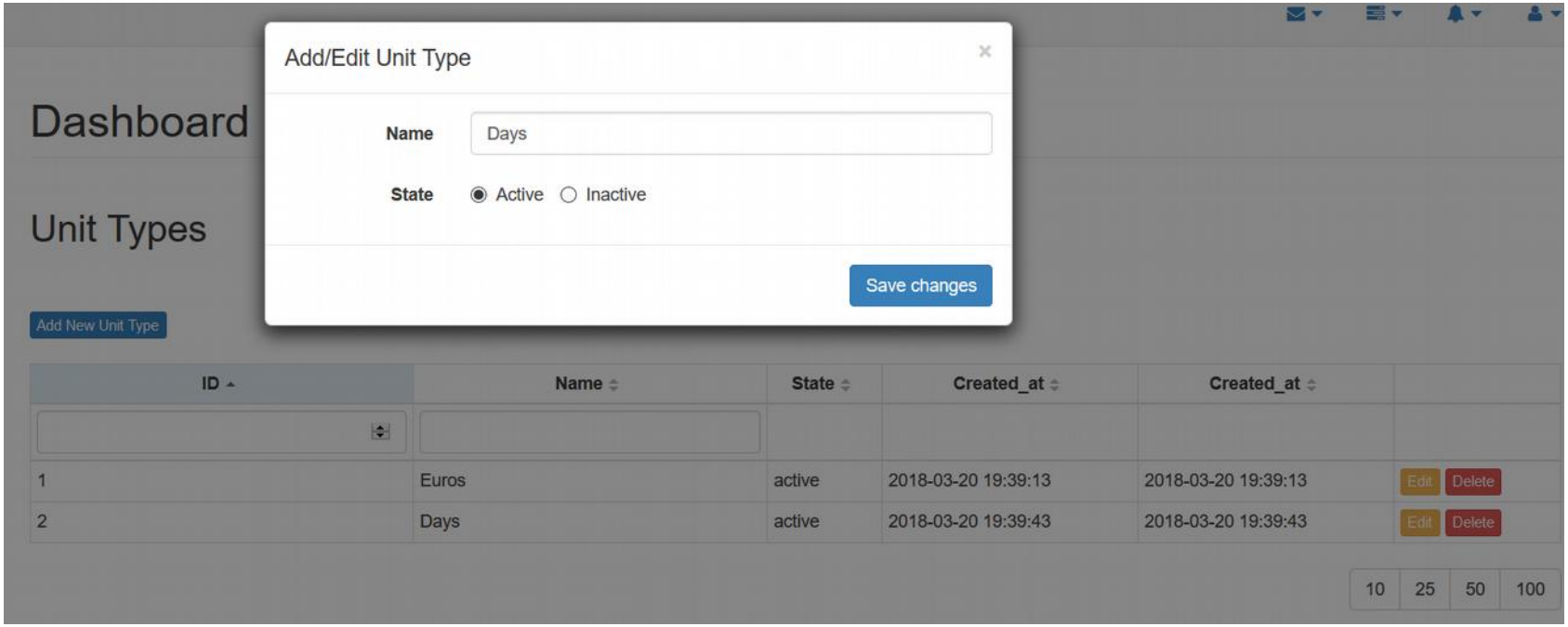
System Modeling Functions

Relation Type Management – used to specify many-to-many relationships between entities; properties can also be associated to relationships

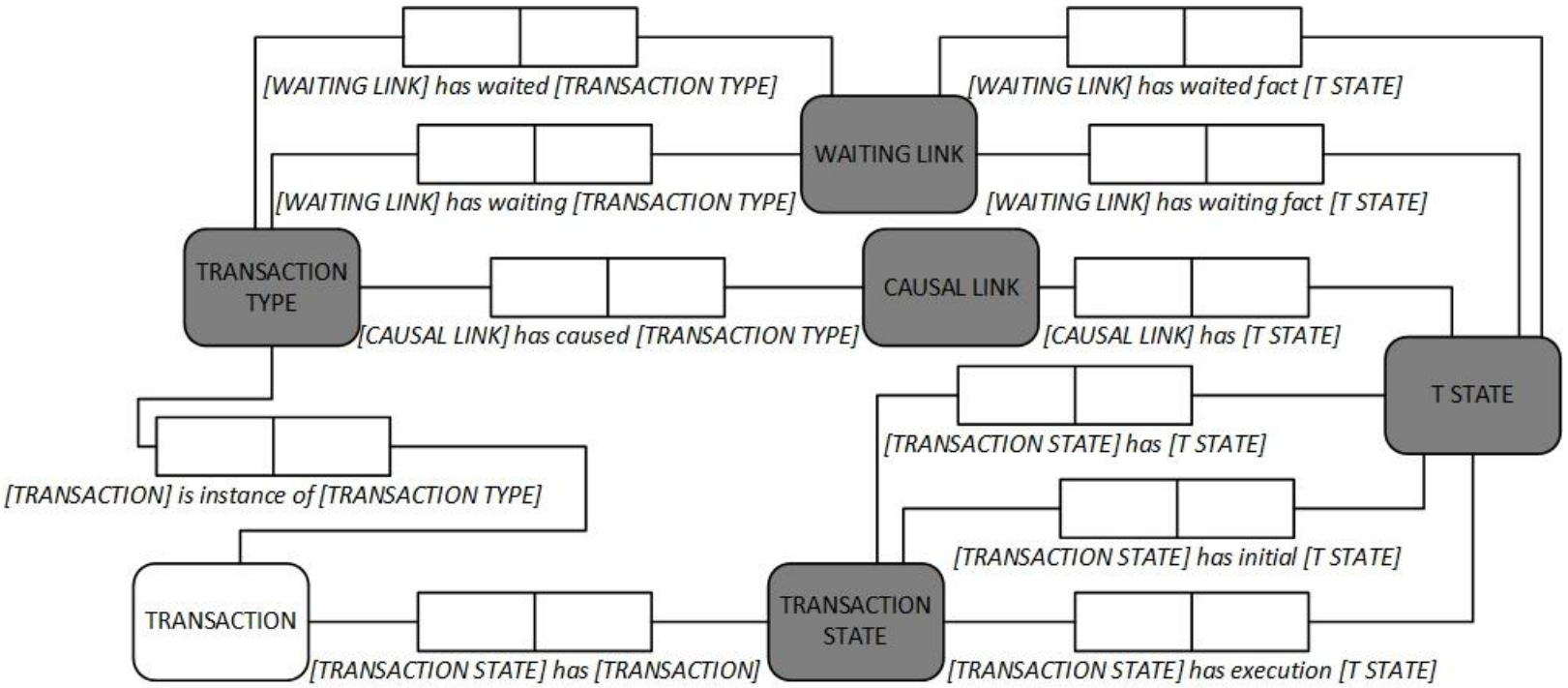


System Modeling Functions

Unit Management – Used to specify unit types that are needed for properties denoting a certain unit of measure to be used in the interface generation, such as kg (kilograms), l (liters), etc.

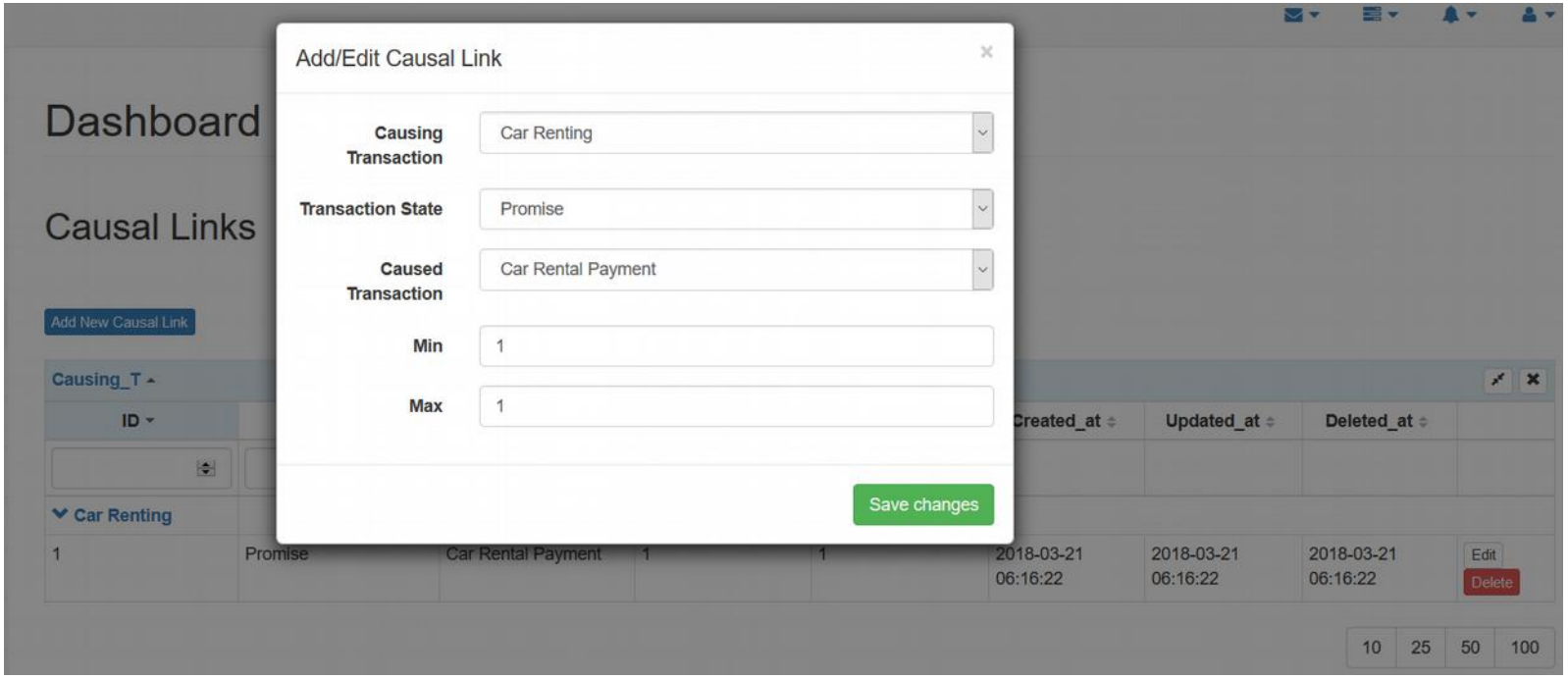


System Modeling Functions



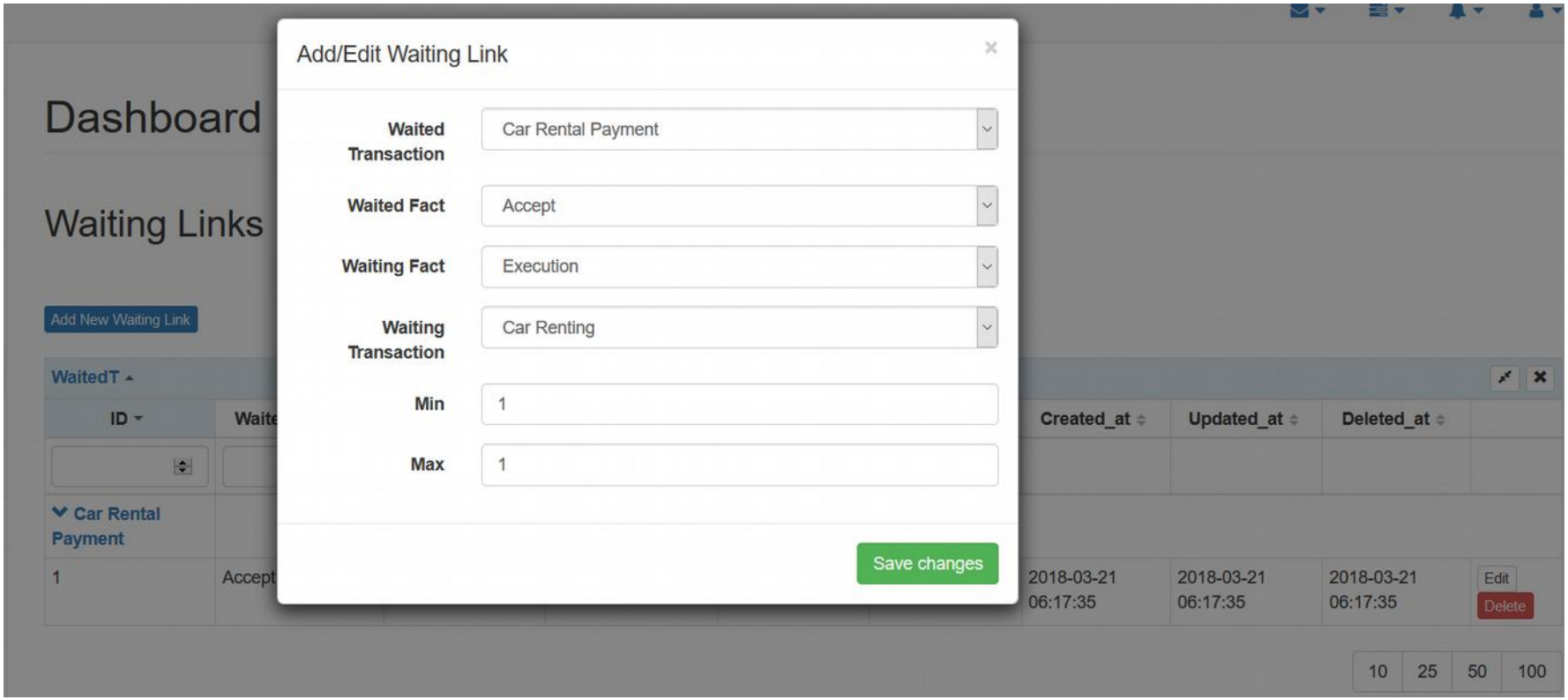
System Modeling Functions

Causal Link Management – specification of occurrence rules of transactions: an act in a transaction can immediately cause the initiation of a new transaction (e.g., promising the requested rental leads to an immediate request for the payment), hence users do not need to manually start transactions that naturally follow the process flow as these rules are automatically applied by the dashboard.

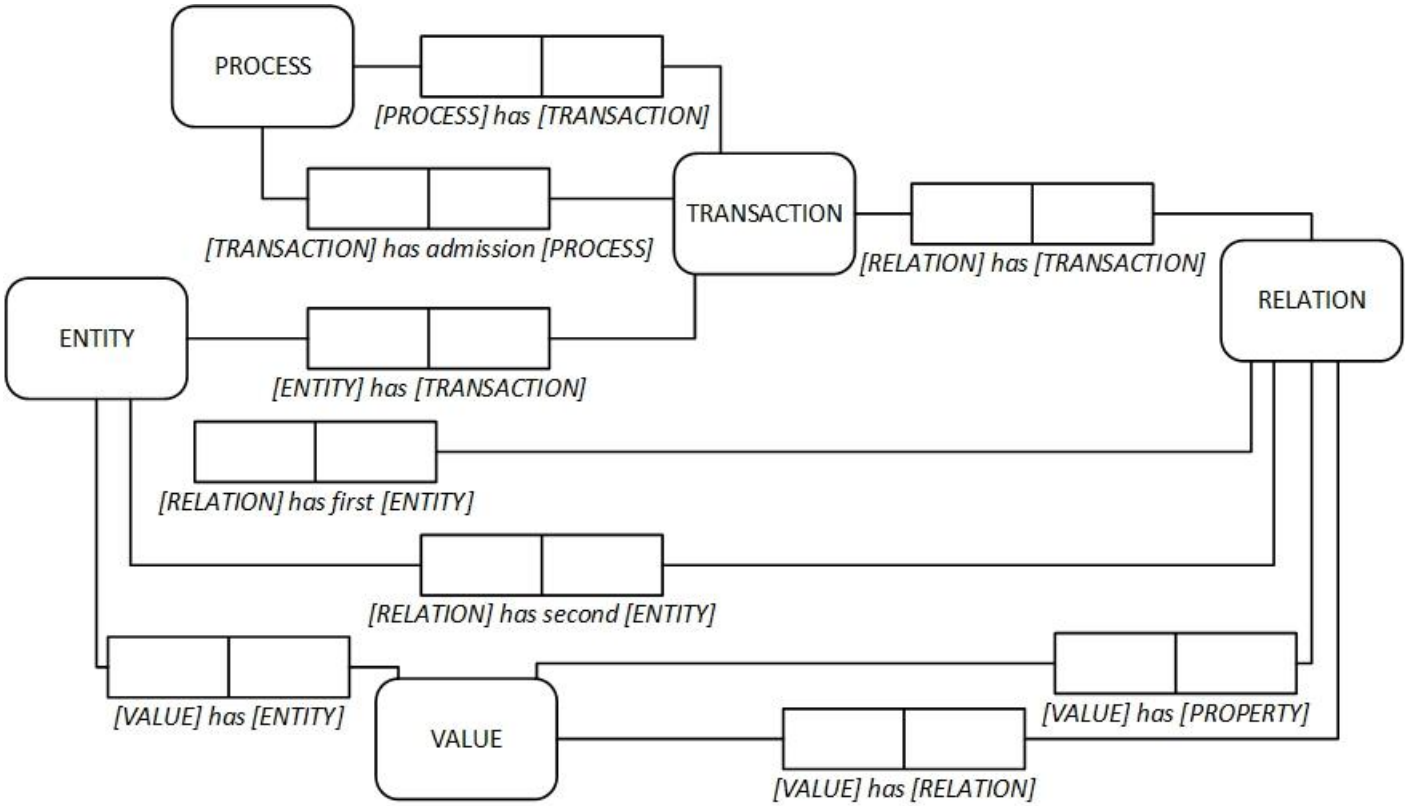


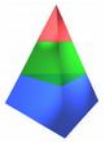
System Modeling Functions

Waiting Links – specification of conditional waiting rules: certain steps of a transaction can only continue if a certain act of another transaction has been performed.



System Execution





System Execution

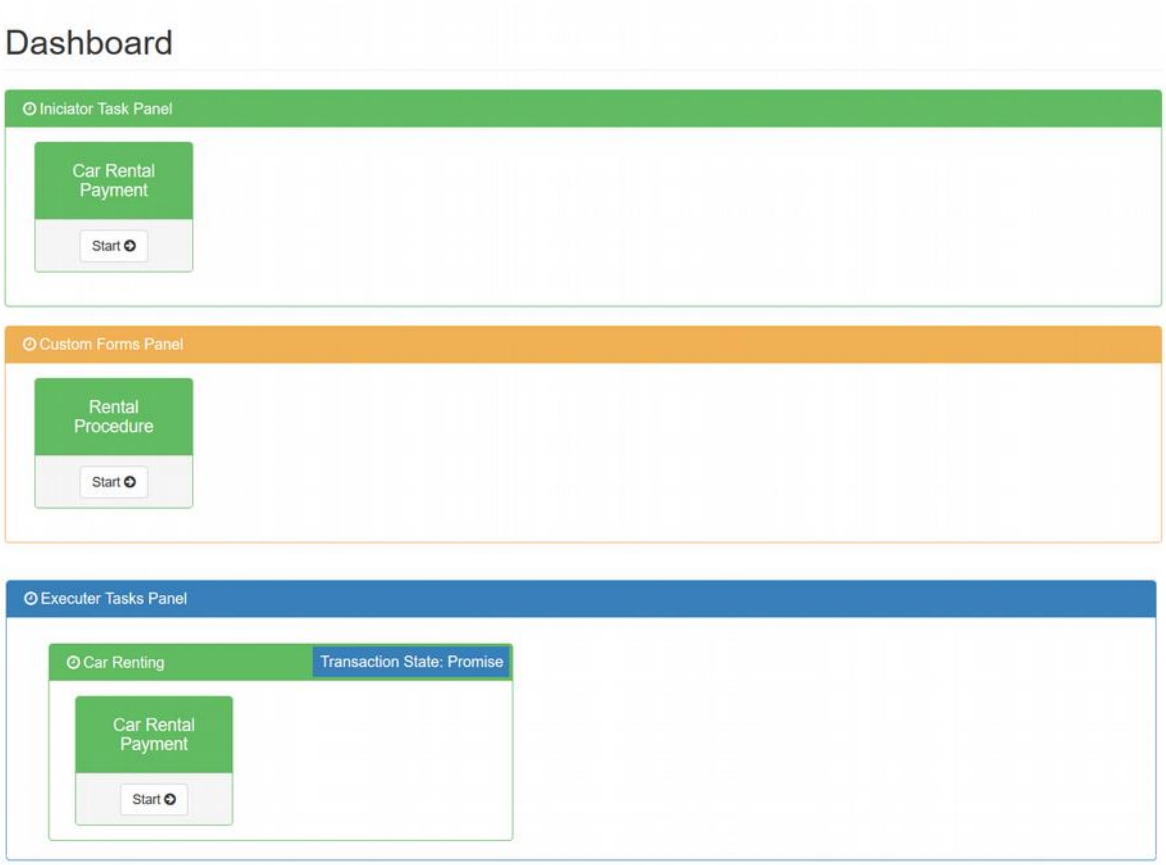
All users when logged in DISME are directed to the Dashboard where a list of the tasks they are allowed to perform is shown.

A user can execute a request act to start some specific process or react to a certain process state to which he or she were given authority and responsibility to do so – if some property/entity is associated to that act the user will have to fill out a form, automatically generated based on the specified parameters.

The Dashboard automatically controls the flow and state of all process instances and data, thanks to both the causal and waiting links that are specified in the respective modeling functions and the data submitted by the users.

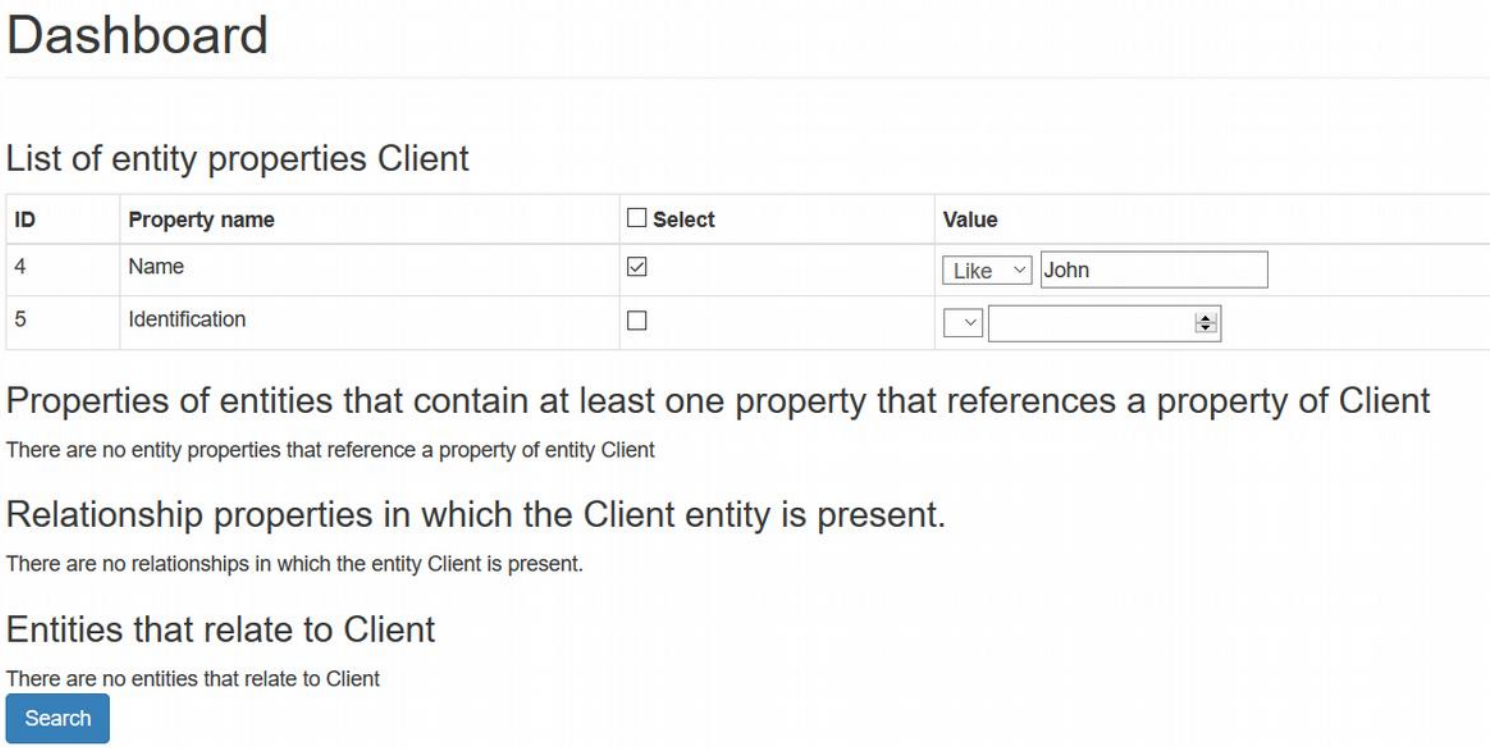
System Execution

System Execution



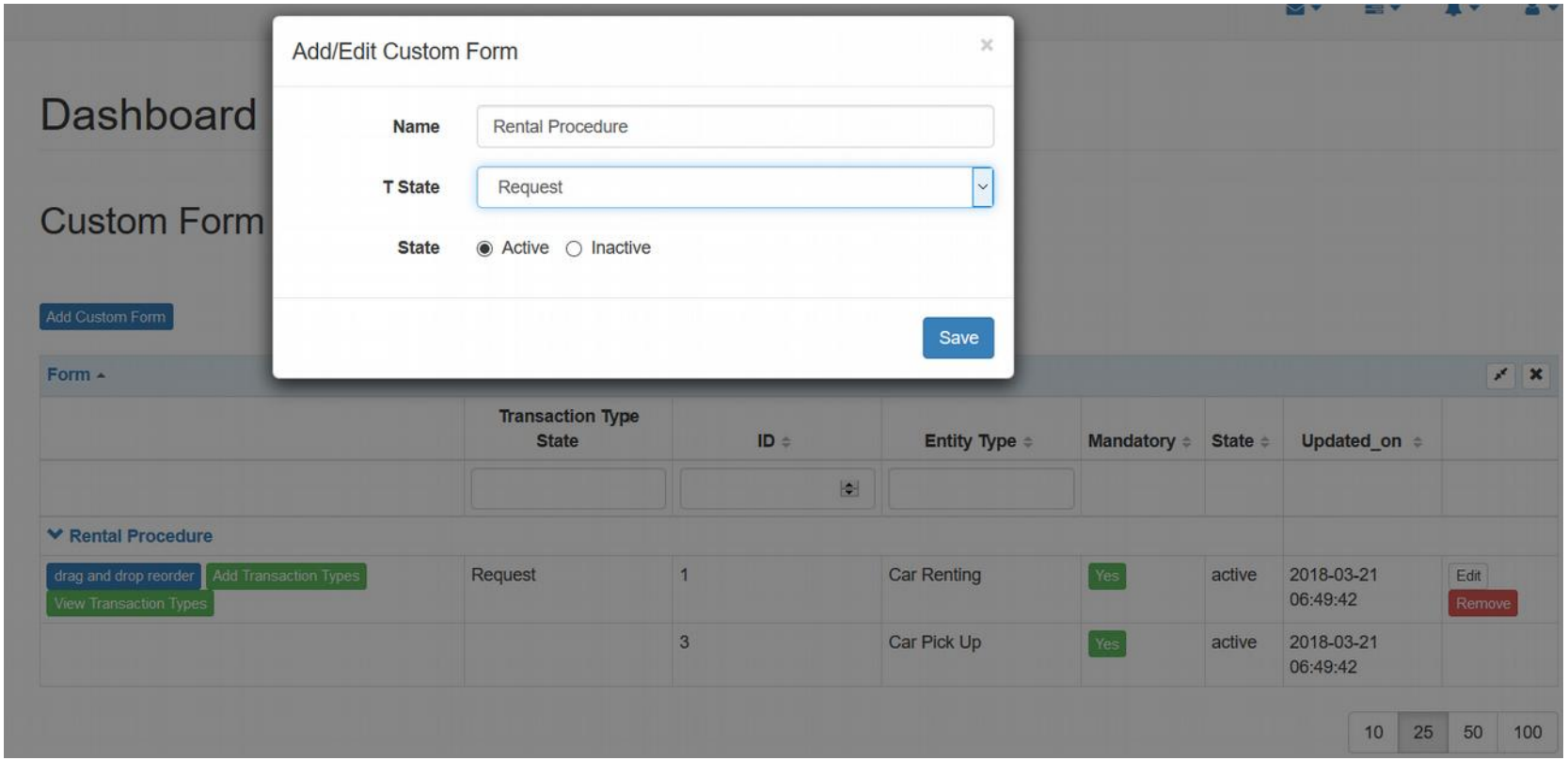
Other functions

Dynamic Search – specification of queries based on triplets of property-operator-value, chosen by the user selecting the relevant options in a user-friendly graphical interface and without the need of any programing (e.g. SQL). Specifications can be saved for later use or also used to display useful information in some form.



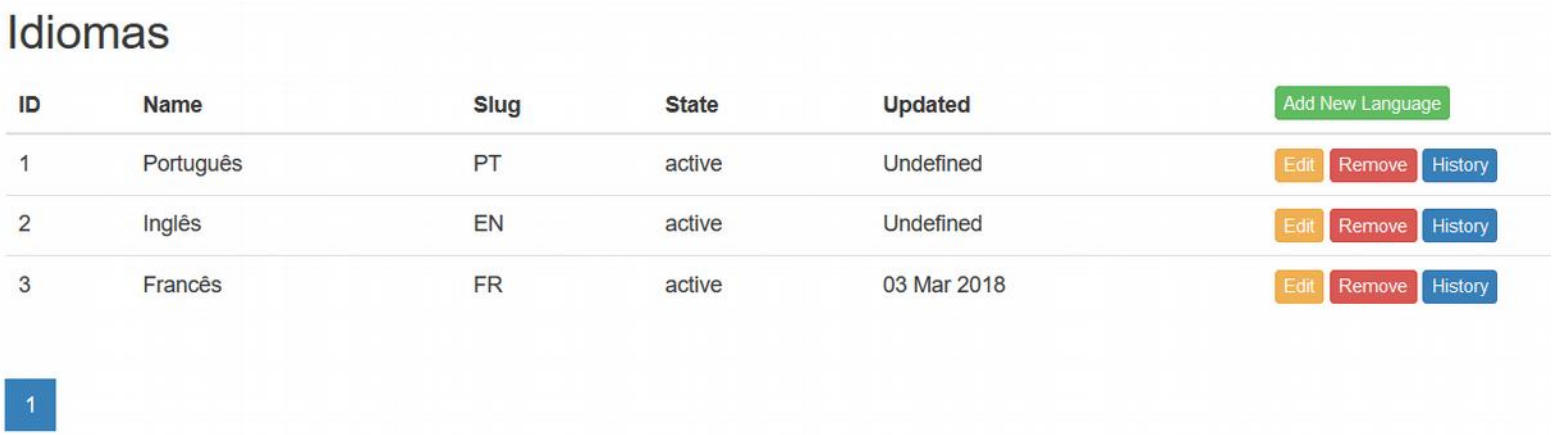
Other functions

Custom Forms – used to group forms from one or more transactions so that all fields are displayed and populated by the user at the same time. Useful in the case that a particular business task implies performing together acts of two or more transactions at the same time.



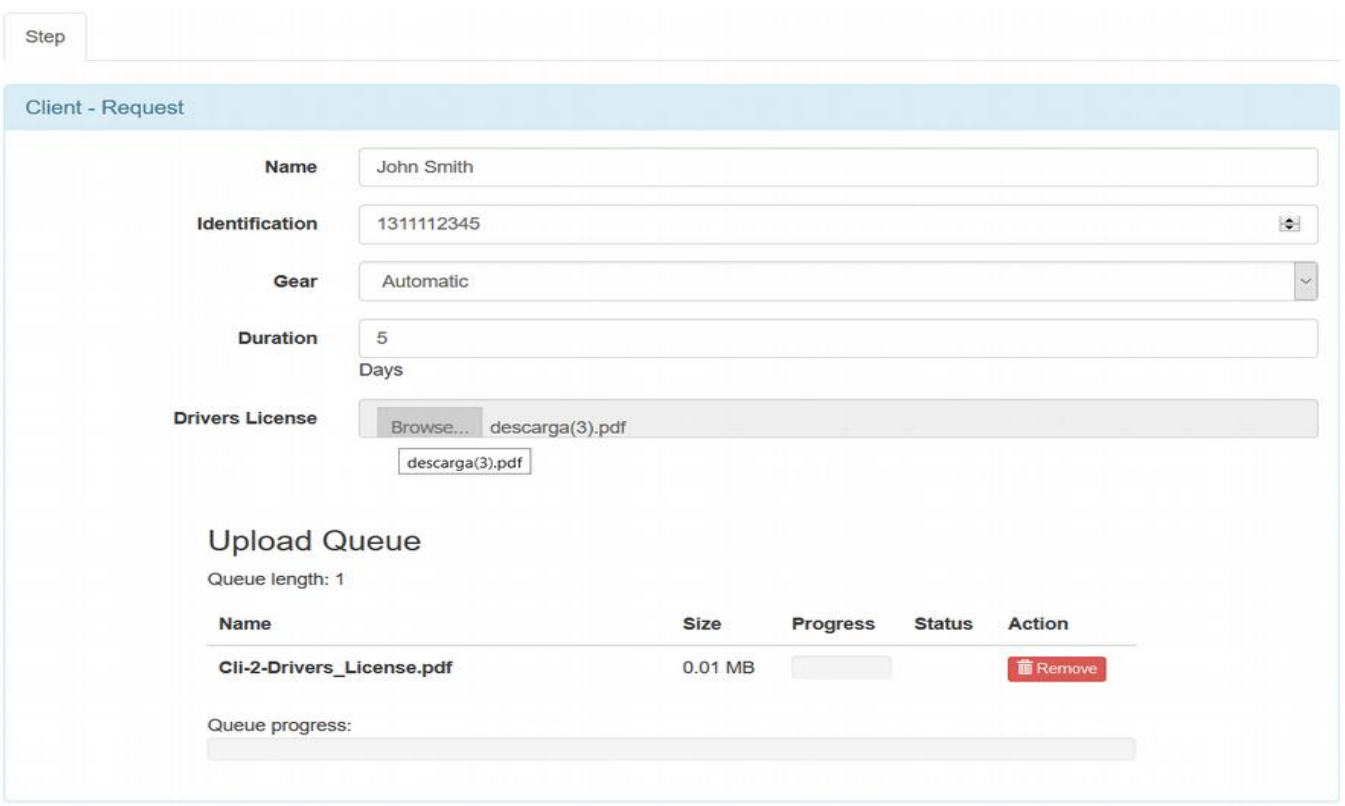
Other functions

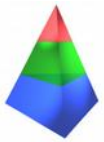
Language Management – DISME is multilingual ready, both in terms of its native interface and also regarding the runtime interface generated for each user in the Dashboard, and according to the language preference set by the user or administrator.



Other functions

Document upload – Specific Property Value Type that allows attachment of files as value to some entity.



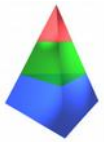


DISME: Important Remarks

Nothing is erased, an historic is kept of all changes in all concepts, both at type/model level and at run-time/instance level

If a change is made at type/model level, it immediately reflects on the system's behavior. For example, adding a new property to an existing entity type will result that the form generated in the respective transaction step will now show the respective field.

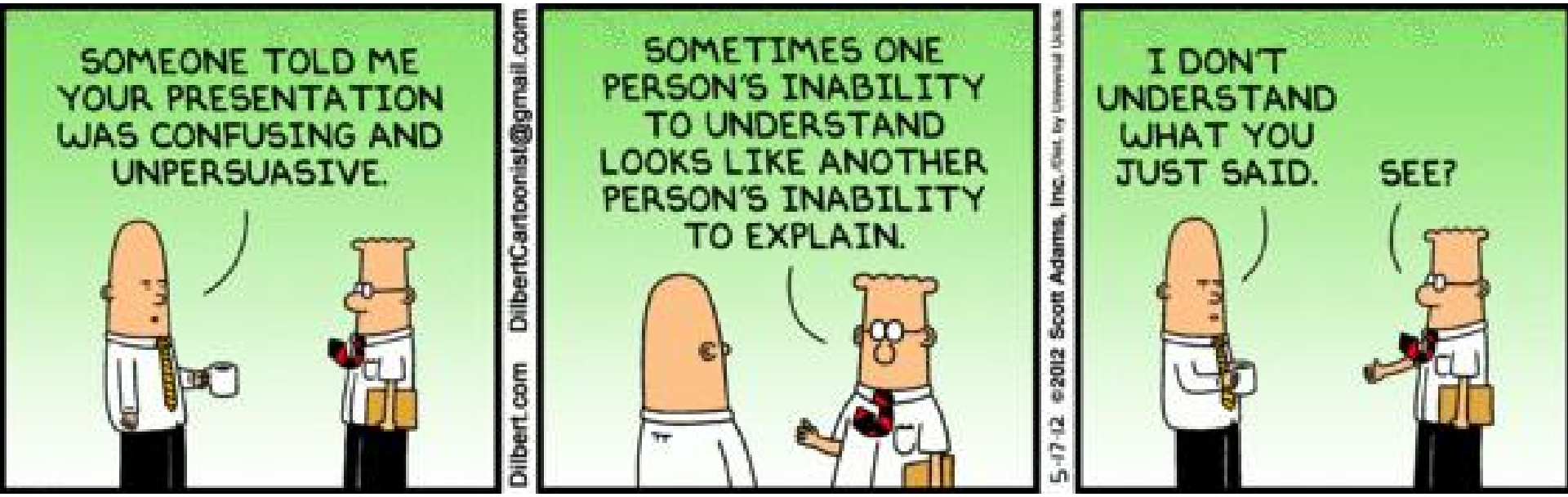
Our conceptual model follows in many parts the type square pattern and the principle of Adaptive Object Model and this is key to the ability of the system to immediately change its run-time behavior according to the change in the specification of some concept at type/model level



Ongoing and Future Work

- Implementation of business rules to:
  - restrict form behavior (e.g. restrict values in fields)
  - control and automate flow according to verification of certain conditions
- Integration of PHP/Javascript code snippets (for implementation of certain business rules) that are interpreted dynamically in the automatically generated forms and stored in the database
- Input and output of data in a SOA way (e.g. using JSON, REST) for seamless and automatic integration with other systems
- Different form generation/behaviour with multiple interfaces for the same entity types/properties, depending on the context
- Versioning of types leading to version and data transparency
  - different versions of a process type can co-exist in runtime
  - possibility of automatic or semi-automatic migration of instances from a previous version to a new one

Questions?



More info: david@ciaonetwork.org

www.ciaonetwork.org

www.ee-institute.com