DEMO-3 Models and Representations

From the experience in applying DEMO-2 (as described in the book Enterprise Ontology, 1st edition, 2006) a number of refinements and simplifications have emerged regarding the DEMO models and their representations. This document contains the models and their representations as proposed for DEMO-3.

The complete ontological model of an organization is divided into four aspect models that represent four views on the integral ontological model: the Construction Model, the Process Model, the State Model, and the Action Model. The division is based on the ψ-theory. Note that only the integrated whole of the four aspects models can provide a full understanding of the ontology of the modeled organization.

Construction Model

The *Construction Model* (CM) of an organization is the conceptual model of its construction: the composition, the environment, and the structure (i.e. the active and passive mutual influences among the elements in the composition and between these and the elements in the environment). Thus, the CM contains the identified actor roles in the composition and in the environment, the identified transaction kinds among the actor roles in the composition and between these and the actor roles in the environment, as well as the information links from the actor roles in the composition to the internal fact banks (i.e. the production banks and coordination banks of the identified transaction kinds), and to the identified external fact banks.

The CM is represented by means of the *Organization Construction Diagram* (OCD), the *Transaction Result Table* (TRT), and the *Bank Contents Table* (BCT).

Note
In DEMO-2, the CM was divided into two partial models: the Interaction Model and he Interstriction Model, the first one containing the active mutual influencing relationships between elements and the second one containing the passive mutual influencing relationships. Consequently, the Organization Construction Diagram was built up of two parts: the Actor Transaction Diagram (ATD) and the Actor Bank Diagram (ABD). To accommodate the professionals who want to keep making this distinction, DEMO-3 allows to have two distinct subviews on the CM: the Interaction View and the Interstriction View. They are similar to the Interaction Model and he Interstriction Model in DEMO-2. Regarding their representations, the Interaction View is represented by the ATD-part of the OCD and the Interstriction View is represented by the ABD-part. Usually, however, one adds the interstriction links to the ATD, and in doing so producing the OCD.

Process Model

The *Process Model* (PM) of an organization is the conceptual model of the state space and the transition space of the coordination world of the organization. Regarding the state space, the PM contains all identified process steps in all identified transaction kinds and the existence laws that apply. The steps, as well as the existence laws, are fully determined by the Universal Transaction Pattern. Regarding the transition space, the PM contains the occurrence laws that hold for the transaction steps, including the cardinalities of the occurrences. These occurrence laws are largely determined by the Universal Transaction Pattern. There are two kinds of occurrence laws: precedence laws (also called causal conditions) and prerequisite laws (also called wait conditions).

The PM is represented by means of the *Process Structure Diagram* (PSD) and the *Transaction Pattern Diagram* (TPD). In these diagrams, precedence laws are represented by solid arrows, whereas prerequisite laws are represented by dashed arrows.

State Model

The *State Model* (SM) of an organization is the conceptual model of the state space and the transition space of the production world of the organization. Regarding the state space, the SM contains all identified transaction result kinds, all identified fact kinds (both declared and defined, and both internal and external), as well as the existence laws that apply. Four kinds of laws are distinguished: reference laws, unicity laws, dependency laws, and exclusion laws. As follows from the ψ-theory, the transition space of the production world is completely determined by the transition space of the coordination world. Nevertheless, it may be illustrative to show the occurrence laws between the transaction result kinds, by declaring the one as a specialization of the other.

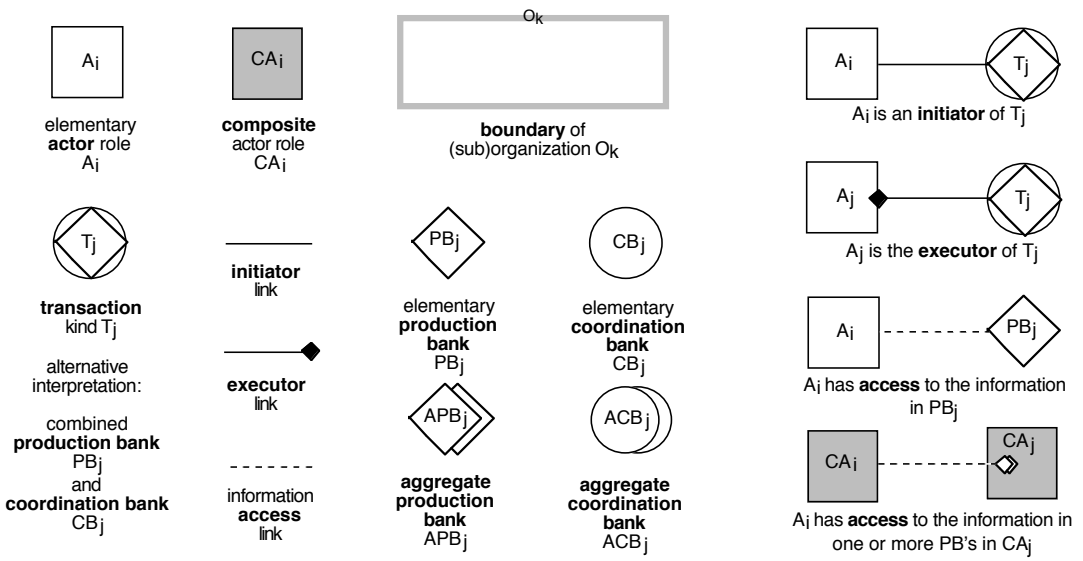
The SM is represented by means of the *State Space Diagram* (SSD).

Action Model

The *Action Model* (AM) of an organization contains an action rule for every agendum kind for every identified actor role. An action rule specifies the (production and/or coordination) acts that must be taken, and the corresponding conditions (in the production world and/or the coordination world) that must hold, on the occurrence of an instance of the agendum kind (which is an event in the coordination world).

The AM is represented by means of *Action Rule Specifications* (ARS).

Legend of the Organization Construction Diagram



 ©2009 Jan L.G. Dietz

Legend of the TRT and the BCT

The **Transaction Result Table** (TRT) is a table of the identified transaction kinds. The structure of each entry is:

<transaction number> <transaction name> <result fact number> <result fact formulation>, where:

<transaction number>	::= T<numerical code>	Example: T01
<transaction name>	::= <nominative expression>	Example: rental start
<result fact number>	::= R<numerical code>	Example: R01
<result fact formulation>	::= <verbal expression in past tense>	Example: [rental] has been started

Note that in the result fact explanation, variables are denoted by [<entity kind>]

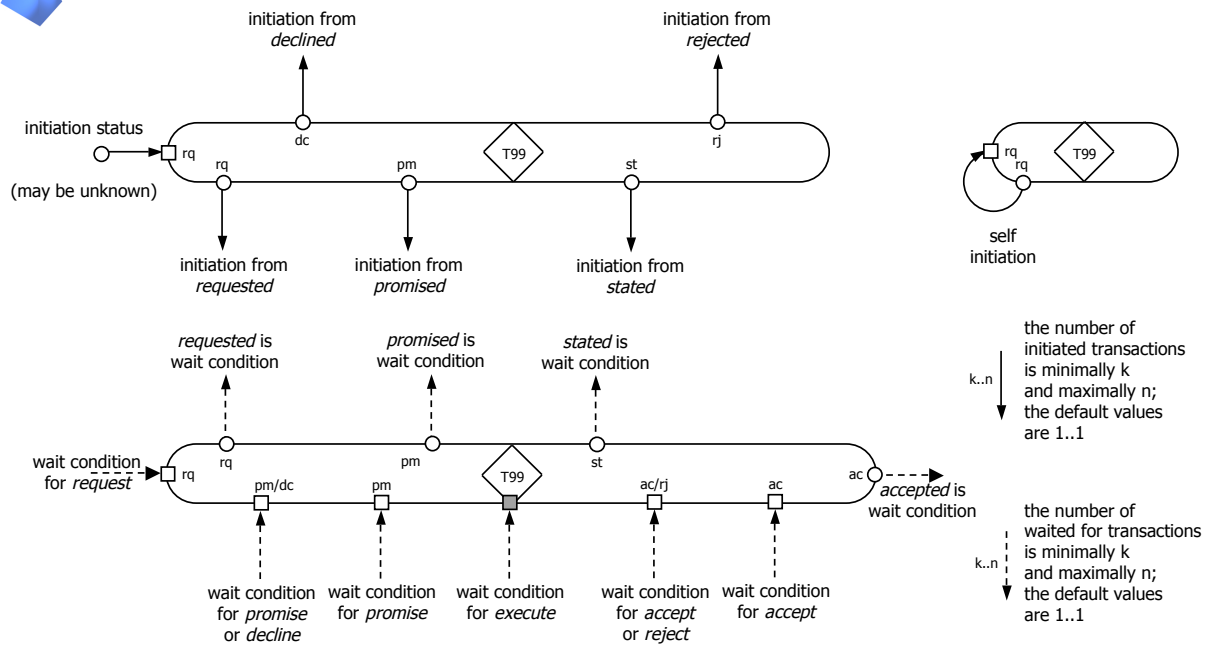
The **Bank Contents Table** (BCT) is a table that shows the fact kinds of which instances are contained in the listed production banks. The structure of each entry is:

<production bank number> <fact kind reference>, where:

<production bank number>	::= PB<numerical code>	Example: PB01
<fact kind reference>	::= <name of fact kind extension>	Example: RENTAL
	| <formulation of fact kind intension>	Example: the renter of [rental] is [person]

 ©2009 Jan L.G. Dietz

Legend of the Process Structure Diagram (1)



The 'sausage' is an 'extracted' transaction symbol. The diamond remains unchanged.
There is an invisible, non-proportional, time line from left to right. So, process steps occur in the order they have in the (standard) transaction pattern.

 ©2009 Jan L.G. Dietz

Legend of the Process Structure Diagram (2)

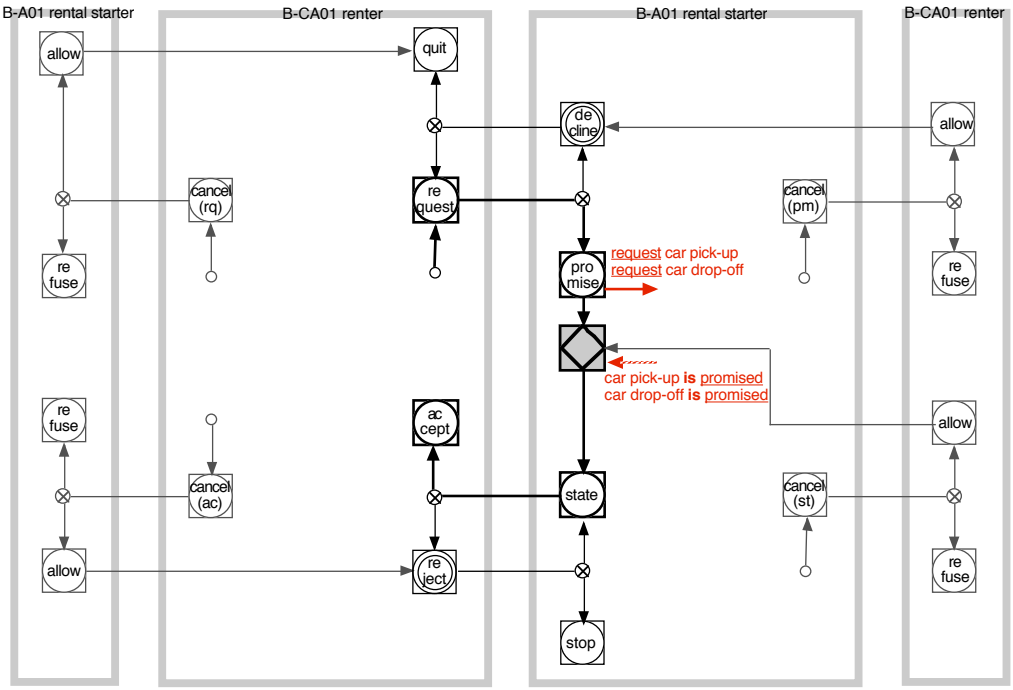
Every transaction status may, except the being executed, serve as a point from which another transaction is initiated, although only a few are mentioned in the PSD above.

Likewise, every transaction status, except the being executed, may serve as a point from which a wait condition holds, although only a few are mentioned in the PSD above.

Moreover, there may be a wait condition for every transaction step, although only a few are mentioned in the PSD above.

 ©2009 Jan L.G. Dietz

Legend of the Transaction Pattern Diagram (1)



 ©2009 Jan L.G. Dietz

Legend of the Transaction Pattern Diagram (2)

A box represents an act (C-act or P-act), a disk represents a C-fact, and a diamond represents the P-fact of the transaction.

An act with it resulting fact is called a transaction step. They are represented by the combination of a box and a disk or diamond.

The grey-lined large rectangles represent the responsibility areas of the initiator and the executor. In the example, the initiator is B-CA01 and the executor is B-A01.

The symbol for the production step is colored grey to emphasize that it is performed by the executor in isolation.
The meaning of a solid arrow from a disk or diamond to a box, is that the responsible actor will (try to) perform the act when the transaction status that is represented by the disk or diamond, has been reached. An external status (C-fact) is represented by a small disk.

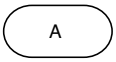
A double disk represents a discussion status.

A small disk with a cross in it represents an exclusive or. It means that the responsible actor either performs the one act or the other.

For clarification, inter transaction links may be added to the TPD. In the example TPD these are represented in red.

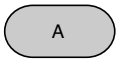
 ©2009 Jan L.G. Dietz

Legend of the State Space Diagram (1)

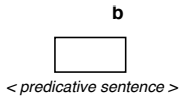


notation of the *object class* A

A is the extension of
the unary fact kind **a**



external object class A

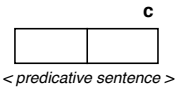


notation of the *unary fact kind* **b**

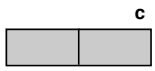
the *predicative sentence* explains **b**
it contains one variable, for its single role

external unary
fact kind **b**

notation of the *binary fact kind* **c**

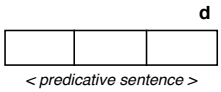


the *predicative sentence* explains **c**
it contains two variables, for every role one



external binary
fact kind **c**

notation of the *ternary fact kind* **d**

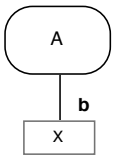


the *predicative sentence* explains **d**
it contains three variables, for every role one

external ternary
fact kind **d**

 ©2009 Jan L.G. Dietz

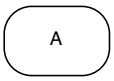
Legend of the State Space Diagram (2)



declaration of the unary fact kind **b**

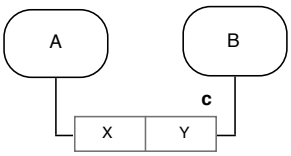
notation of the ***reference law***
for the unary fact kind **b**: □(**b**(x) ⇒ **a**(x))

A is called the *domain* of (the single role of) **b**



declaration of the *primal* unary fact kind **a**

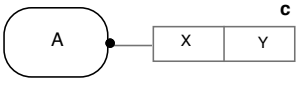
A is called a *category*



declaration of the binary fact kind **c**

notation of the ***reference laws***
for the binary fact kind **c**: □(**c**(x,y) ⇒ **a**(x) & **b**(y))

A is called the *domain* of the left role of **c**
B is called the *domain* of the right role Y of **c**



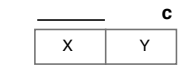
notation of a ***dependency law*** for fact kind **a**: □(**a**(x) ⇒ **c**(x, -))

since also holds: □(**c**(x, -) ⇒ **a**(x)) (reference rule),
a(x) and **c**(x, y) are called *existentially dependent*

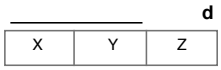
⇒ : *strict implication* □ : *necessity* ◇ : *possibility* ~ : *negation*

 ©2009 Jan L.G. Dietz

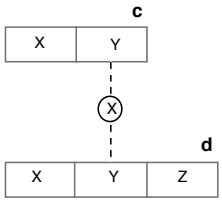
Legend of the State Space Diagram (3)



notation of a unary ***unicity law*** :
~◇ (**c**(x, y) & **c**(x, z) & y≠z)



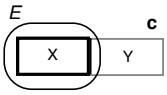
notation of a binary ***unicity law*** :
~◇ (**d**(x, y, z) & **d**(x, y, w) & z≠w)



notation of a mutual ***exclusion law*** :
~◇ (**c**(-, y) ⇒ **d**(-, y, -))
~◇ (**d**(-, y, -) ⇒ **c**(-, y))

 ©2009 Jan L.G. Dietz

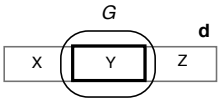
Legend of the State Space Diagram (4)



notation of the extensional definition of the *unary fact kind* **e**

E = { x I ∃y : **c**(x, y) }

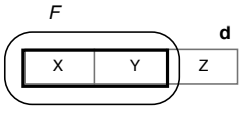
e is a *specialization* of the domain of role X



notation of the extensional definition of the *unary tact kind* **g**

G = { y I ∃x, z : **d**(x, y, z) }

g is a *specialization* of the domain of role Y



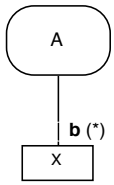
notation of the extensional definition of the *binary fact kind* **f**

F = { (x, y) I ∃z : **d**(x, y, z) }

f is a *specialization* of the cartesian product of the domain of role X and the domain of role Y

 ©2009 Jan L.G. Dietz

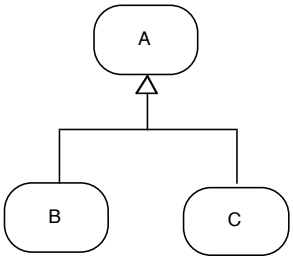
Legend of the State Space Diagram (5)



notation of the intensional definition
of the *fact kind* **b**

the generic *definition formula* is:
b(x) ≡ **a**(x) & < predicate over x >

b is called a ***specialization*** of **a**



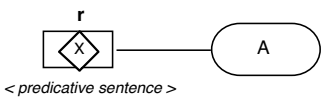
notation of the extensional definition of
the *fact kind* **a**

A = B ∪ C

a is called the ***generalization*** of **b** and **c**

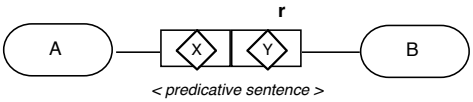
 ©2009 Jan L.G. Dietz

Legend of the State Space Diagram (6)



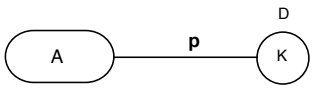
notation of the declaration of the *unary*
transaction result kind **r**, with domain A

the *predicative sentence* explains **r**
it has a placeholder for objects x∈A



notation of the declaration of the *binary*
transaction result kind **r**, with domains A and B

the *predicative sentence* explains **r**
it has a placeholder for objects x∈A and one for objects y∈B



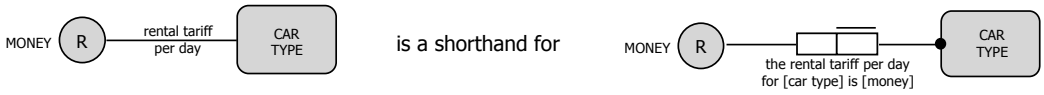
notation of the declaration of the *property kind* **p**,
a mapping from the object class A to the scale dimension D of
kind K

p(x)=y for every x ∈A, with y ∈D

 ©2009 Jan L.G. Dietz

Legend of the State Space Diagram (7)

Note1



In other words, the relationship between CAR TYPE and MONEY is a mathematical function: every member of the domain (object class CAR TYPE) maps to a member of the range (scale dimension MONEY).

Note2

The next general scale dimensions are considered to exist (between brackets the shorthand definition):

TIME (T)	Scales: hr-min-s, D-W-M-Y (in several calendars)
MONEY	Scales: dollar, euro etc.
MASS	Scales: kg, pound
LENGTH (L)	Scales: m, foot, yard
AREA (L^2)	Scales: m^2, acre
VOLUME (L^3)	Scales: m^3, pint, gallon
VELOCITY (L/T)	Scales: m/s
TEMPERATURE	Scales: ^oC, ^oF

After each dimension a number of possible scales are mentioned. On the ontological level, however, one does not care about a particular scale; only the scale dimension is important. Note that both points in time (time periods) and time durations are expressed in the TIME dimension.
Note that a specific scale kind belongs to each of the dimensions (Ordinal, Interval, Ratio or Absolute).

 ©2009 Jan L.G. Dietz

Legend of the Action Rule Specification

basic action rule syntax

when < transaction kind > **of** < object identifier(s) > **is** < transaction status >
if < boolean logical expression >
then < transaction kind > **of** < object identifier(s) > **must be** < perfective form of C-act >
else < transaction kind > **of** < object identifier(s) > **must be** < perfective form of C-act >

Note1: < transaction status > = requested, promised, executed, stated, accepted, etc.

Note2: < perfective form of C-act > = requested, promised, executed, stated, accepted, etc.

Note3: *action rules* are the operational form of the *laws* in the Process Model and the State Model.

Note4. The syntax and the formal semantics of the action rules need to be elaborated yet.

 ©2009 Jan L.G. Dietz