Semantic Model for Lead Management

Open Business Concepts

Author:Dominique VAUQUIER

Version: 1.0
Status: Reviewed by the "Services & Data
modeling" initiative workgroup – AXA Group
Publication: 12/10/2009
Copyright: Creative Commons

Rev	Date	Auteur	Description
1.0	2009-10-12	dvau	Final version taking heed of the workgroup feedback (2009 September 21 workshop)

Table of content

Table of figures

Table of tables

1 Introduction

Definition: "The semantic model formalizes business fundamentals."
Organizational choices revealed by business processes are excluded. The semantic model is not an IT model: it represents business knowledge and - to be more precise - the invariant part of business knowledge, isolated from and independent of practices and operating manuals.
Therefore, the semantic model has a universal vocation.

For further information, please refer to the "MDS-11" document ("The semantic model").

2 Package Index

Semantic aspect

Reality

Geography

Temporality

Ontology

Phenomenology

Catalogue

Portfolio

Production

Information

3   "Semantic aspect" Package

from Package <u>*Open Business Repository*</u>

Definition: "The semantic model formalizes business fundamentals."
Organizational choices revealed by business processes are excluded. The semantic model is not an IT model: it represents business knowledge and - to be more precise - the invariant part of business knowledge, isolated from and independent of practices and operating manuals.
Therefore, the semantic model has a universal vocation.

For further information, please refer to the "MDS-11" document ("The semantic model").
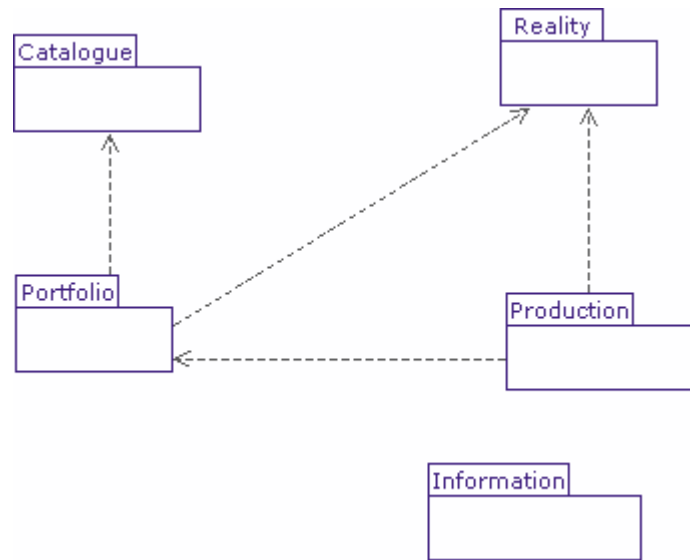
Figure 1 Object domains

On the scale of the enterprise, the semantic model becomes the enterprise reference model. It covers all of the domains and all the objects. It contains hundreds of classes, so it becomes indispensable to organize and classify its content.
As soon as the question of the structure of the model is addressed the architect must pay particular attention to the classification criteria retained. The functional domain can be used as a criterion for the organizational aspect but it is not suitable for the semantic aspect. In fact, the same business objects are involved in activities in several functional domains. It is therefore necessary to introduce a new notion specifically for the semantic aspect that is capable of correctly organizing the semantic objects.
This is achieved by breaking the semantic model down into "object domains". An object domain is obtained by regrouping the objects of the extended neighborhood of a principal object. It takes about a dozen principal objects to describe the activity of an enterprise. In our case: real object, person, product, contract, accident… Object domains are drawn around these objects. The precept says that an object, an item of information, an action on an object, a state transition… may only be represented in a single place in the model and be expressed in only one form.

The diagram below shows an example of a semantic model broken down into object domains. It shows, for example, the "Reality" domain containing the semantics of objects outside the enterprise. It is here, in effect, that part of the customer orientation is taken into account.

In this UML diagram, the object domains are represented by packages. The arrows show dependencies between domains. There are many dependencies. In the semantic model, this tight coupling can be tolerated up to a certain point, because the objective is to represent knowledge. The logical architecture diagram, however, will transform the object domains into logical constituents, but reduce the coupling by reducing the dependencies.

For a justification of each dependency, please refer to the documentation of each object domain.

Name	Summary
Reality	The "Reality" domain covers all objects and concepts we find in the real world, at least in the enterprise environment. The quality of this domain relies on its universality. Indeed, the model of this domain is such a generic one that it should be recognized by every organization, in any kind of business. Sharing this representation is a condition for communication between various actors (companies inside a group, partners along a supply chain, etc.). The main objects of this domain include: - "Party", the root class for persons and organizations that may assume one or many roles in interaction with the enterprise; - "Asset", with many specific classes that describe tangible assets (real estate, vehicles, machines, goods…); - Geography (addresses…); - Temporality. This domain is very rich and needs to be subdivided with subdomains.
Catalogue	The catalogue describes the products and services that the company delivers. The root is the generic class "Offer", with two branches: "Product" and "Service". The maximum of properties are factorized into these generic classes.
Portfolio	The portfolio makes the connection between objects of the reality – mainly parties – and the enterprise offering. The contract notion establishes the conjunction.
Production	Based on our knowledge of real objects and all along the relation established through contracts, the company follows the events that occur, emits information or invoices, and delivers services… This activity and the information it produces, live at a different pace, compared to the information compiled in the previous domains. That justifies defining a separate object domain. "Production" encompasses this day-to-day information. The objects inside this domain are arranged around these main objects: - Premium, - Claim, - Operation (e.g. survey, assessment, service delivery), - Payment.
Information	This package is not an object domain - strictly speaking - but a set of facilities, including the definition of general types.

Table 1 Owned Packages of "Semantic aspect" Package

4 "Reality" Package

from Package Open Busines Repository.<u>*Semantic aspect*</u>

The "Reality" domain covers all objects and concepts we find in the real world, at least in the enterprise environment. The quality of this domain relies on its universality. Indeed, the model of this domain is such a generic one that it should be recognized by every organization, in any kind of business. Sharing this representation is a condition for communication between various actors (companies inside a group, partners along a supply chain, etc.).

The main objects of this domain include:
- "Party", the root class for persons and organizations that may assume one or many roles in interaction with the enterprise;
- "Asset", with many specific classes that describe tangible assets (real estate, vehicles, machines, goods…);
- Geography (addresses…);
- Temporality.
This domain is very rich and needs to be subdivided with subdomains.

Figure 2 Content of the "Reality" domain

This diagram illustrates the content of the "Reality" domain (but does not show every class and type). This object domain contains the description of the Enterprise environment. Thus, it describes basic objects and notions that are quite universal. Maintaining the universality of the description will allow for sharing, for instance with partners in other activity sectors.

There is no dependency of this domain on any other object domain, since it comprises the most fundamental content: the representation of external reality itself.
There is a dependency to the "Information" package for use of very general types.

Modeling decision:

It is worth noticing that this position will differ in the logical architecture, which takes into account other considerations related to the IT system.

Name	Summary
Geography	This object domain contains all notions that pertain to geography, location, physical conditions for reaching someone or something.
Temporality	This package puts together all the notions related to time. The classes are definitively generic and can apply to many contexts. Generally, there is no need to specialize the classes of this package. The way of using them inside an object domain is to link the classes of this domain to the classes related to temporality. These new associations express what the date or calendar is intended for.
Ontology	"Ontology" is to be interpreted as closely as possible to its original meaning: the science of being. This object domain encompasses the description of concrete objects, which the enterprise deals with. It includes a classification of beings. The model describes these beings without any assumption regarding the business and how to interact with them, so that this description is quite universal.
Phenomenology	Other candidate name: "actuality". Part of the reality that manifests itself through phenomena, events, actions... As opposed to the "static" part that is represented in the "Ontology" domain. This object domain gathers the notions connected to events (something that is happening). Modeling decision: It is important to separate these sorts of objects from the "beings", since their life spans and behaviors greatly differ. Events may be recorded for a shorter time than parties... They are processed through quicker actions, within a short term horizon.

Table 2 Owned Packages of "Reality" Package

5 "Geography" Package

from Package Open Busines Repository.Semantic aspect.Reality

This object domain contains all notions that pertain to geography, location, physical conditions for reaching someone or something.

Figure 3 Content of the "Geography " domain

The core notion of the "Geography" domain is Site.

This class is a node but carries little information itself. Indeed, the expression of an address depends on the language, while site is independent of it.

The means of communication - e.g. phone & e-mail - are also described in this part of the model.

Name	Summary
Site	This class refers to the many ways of locating an actor. It gives the entry point for finding the various kinds of details, relative to an actor. NB: instances of Site are designated as "Mobile". No address is attached to these instances. The phone numbers attached to such instances are mobile phone numbers. Modeling decision: This way of dealing with mobility avoids another association between Party and Phone Number. Such an association would have led to potential redundancy due to having two paths from Party to Phone Number.
Address	This class lists the representative elements that compose the address of a site.
Country	This codification serves the location purpose but can be used in other contexts (such as product authorization).
Phone Number	The class controls and stores the phone numbers, depending on the national format.

Table 3 Owned Classes of "Geography" Package

Name	Values	Description
Location Nature		This codification lists the possible uses that qualify an address.

Name	Values	Description
Phone Nature	conversation fax telex additional remote transmission	

Table 4 Owned Enumerations of "Geography" Package

5.1 "Site" Class

from Package Open Busines Repository.Semantic aspect.Reality.<u>*Geography*</u>

This class refers to the many ways of locating an actor. It gives the entry point for finding the various kinds of details, relative to an actor.

NB: instances of Site are designated as "Mobile". No address is attached to these instances. The phone numbers attached to such instances are mobile phone numbers.

Modeling decision:
This way of dealing with mobility avoids another association between Party and Phone Number. Such an association would have led to potential redundancy due to having two paths from Party to Phone Number.

Name	Description
resides-> : [0..*] <u>Party</u>	A party, either a person or an entity, can be reached in one or several places (locations). The location can be qualified thanks to the qualifier "location nature". For a given nature, there may be many valid locations. The list of locations is ordered, so that the sites are displayed according to preferences.
-> : [0..*] <u>Reachability</u>	At a given period, a party may be reached in one or many places. The list of possible locations for reaching a party is ordered.
-> : [0..1] <u>Address</u>	
->number : [0..*] <u>Phone Number</u>	For a given phone nature, a site may have several lines. A phone number is always attached to a single site. The case of mobile phone is solved by creating a "mobile" site, instance of the Site class.
-> : [0..*] <u>Being</u>	A being can be reached in one or several places (locations). The location can be qualified thanks to qualifier: "location nature".

Table 5 Associations of "Site" Class
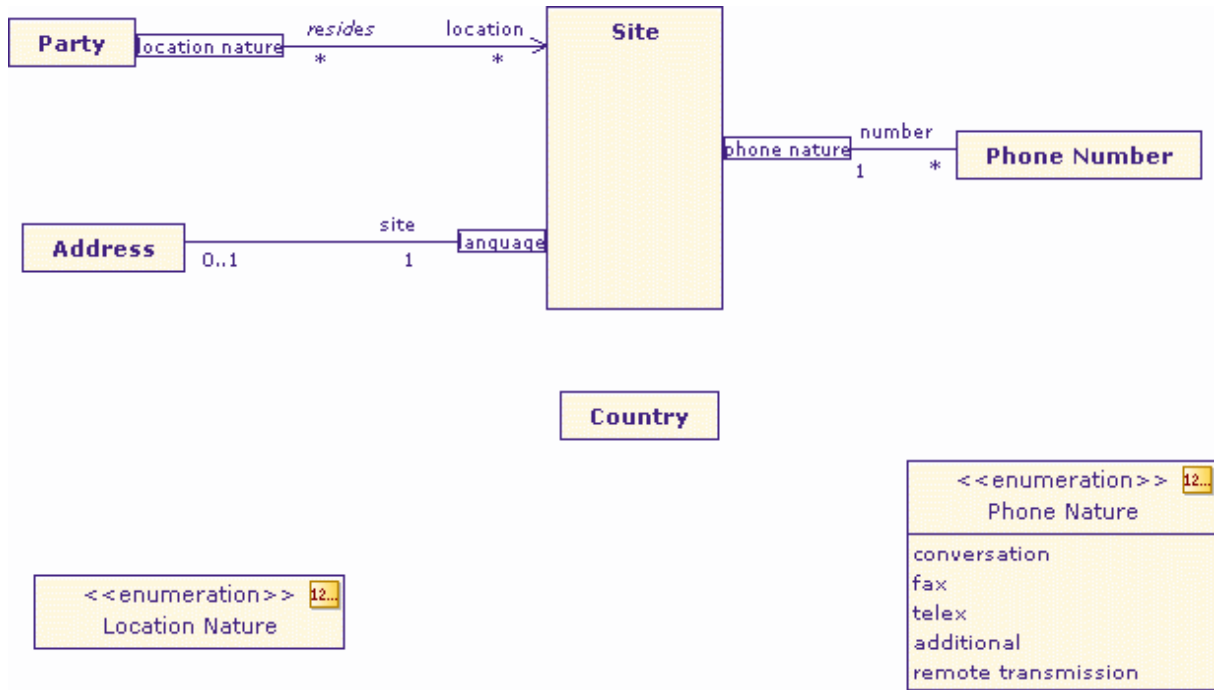
Figure 4 Semantics of Site

This proposed solution for recording the addresses takes into account:
•   the possibility for an actor to be located in many places or by many means;
•   the eventuality of expressing an address in several languages (in Belgian, for instance).

The associations own a qualifier (mentioned in the rectangular box). Please read the diagrams as following:
•   For a given actor, a Site where he resides (or can be reached) is designated, knowing a value of the location nature (home, office, invoice, delivery...).
•   The postal Address of a Site depends on the language in which it is expressed, with the possibility of having many languages for the same place.

5.2   "Address" Class

> from Package *Open Busines Repository.Semantic aspect.Reality.Geography*

This class lists the representative elements that compose the address of a site.

Name	Description
->site : [1..1] Site	

Table 6 Associations of "Address" Class

5.3 "Country" Class

from Package Open Busines Repository.Semantic aspect.Reality. *Geography*

This codification serves the location purpose but can be used in other contexts (such as product authorization).

5.4 "Phone Number" Class

from Package Open Busines Repository.Semantic aspect.Reality. *Geography*

The class controls and stores the phone numbers, depending on the national format.

Name	Description
-> : [1..1] Site	For a given phone nature, a site may have several lines.

Table 7 Associations of "Phone Number" Class

6 "Temporality" Package

from Package Open Busines Repository.Semantic aspect.Reality

This package puts together all the notions related to time.

The classes are definitively generic and can apply to many contexts.

Generally, there is no need to specialize the classes of this package. The way of using them inside an object domain is to link the classes of this domain to the classes related to temporality. These new associations express what the date or calendar is intended for.

Name	Summary
Period	Period of time. A period may be delimited by a starting date and an end date. It may also be infinite when one or both of the terms are not specified. Representing the period as an object rather than storing two dates allows for manipulating it "per se". For instance, holidays may be instantiated and the instances may apply to every actor, in every schedule. When a change occurs, only these instances are affected and the change propagates to every schedule.
Date	
Time	A specific instant of the day (hour, minute, second).
Moment	A moment is a point on the time line. It may be precisely defined, referring to a date and hour. It may also be instantiated without mentioning a date, for further scheduling.

Table 8 Owned Classes of "Temporality" Package

Name	Values	Description
Day	Monday Tuesday Wednesday Thursday Friday Saturday Sunday	The list of days in the week.

Table 9 Owned Enumerations of "Temporality" Package

6.1 "Period" Class

from Package Open Busines Repository.Semantic aspect.Reality.Temporality

Period of time.

A period may be delimited by a starting date and an end date. It may also be infinite when one or both of the terms are not specified.

Representing the period as an object rather than storing two dates allows for manipulating it "per se". For instance, holidays may be instantiated and the instances may apply to every actor, in every

schedule. When a change occurs, only these instances are affected and the change propagates to every schedule.

Name	Description
->scheduled event : [0..*] Event	
-> : [0..*] Party	
-> : [0..*] Opportunity	Some opportunities may present a limited time window, due to the situation of the party or to the conditions of the offer. This validity period is used when planning actions. The minimal cardinality is '0' since period instances may exist in other contexts that lead management. Modeling decision: The orientation of the association obeys the dependency between the packages. Minimum cardinality = '0' => It is possible that an opportunity exists without a limited period of validity. Maximum cardinality = '1' => In most cases, the opportunity is valid from a given date, to another one. But the semantic of period allows more complex situations. Please refer to the documentation of the class Period.
->start : [0..1] Moment	A period may be linked to a moment or not. If there is no start, that means that the period covers the past till the end moment.
->end : [0..1] Moment	A period may be linked to a moment or not. If there is no end, that means that the period applies from the start -if specified- and forever.
->opening time : [0..1] Time	For a given day of the week, this association may specify the opening time.
->closing time : [0..1] Time	For a given day of the week, this association may specify the closing time.
combines->secondary period : [0..*] Period	Thanks to the reflexive association "combines", it is possible to specify complex periods with blank periods inside. The qualifier of the association specifies the relation between the main period and one of the secondary period (inclusion versus exclusion).
combines->main period : [0..*] Period	Thanks to the reflexive association "combines", it is possible to specify complex periods with blank periods inside. The qualifier of the association specifies the relation between the main period and one of the secondary period (inclusion versus exclusion).

Table 10 Associations of "Period" Class
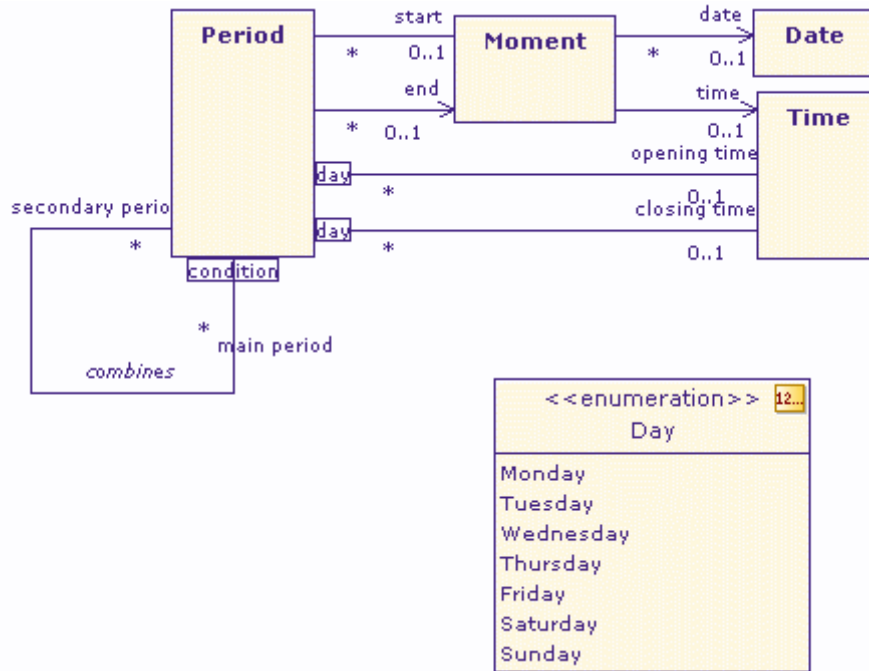
Figure 5 Semantics of Period

A period can be defined as an infinite period (always valid) when the end date is not mentioned.

The condition class allows for registering constraints between periods, such as subtraction (a period with blank periods inside) or an addition of several periods.

The associations to Time class allow for recording the time for a period, for example to declare the opening time.

6.2 "Date" Class

from Package Open Busines Repository.Semantic aspect.Reality.Temporality

Name	Description
-> : [0..*] Moment	

Table 11 Associations of "Date" Class

6.3 "Time" Class

from Package Open Busines Repository.Semantic aspect.Reality.Temporality

A specific instant of the day (hour, minute, second).

Name	Description
-> : [0..*] Period	For a given day of the week, this association may specify the opening time.

Name	Description
-> : **[0..*]** **Period**	For a given day of the week, this association may specify the closing time.
-> : **[undefined..undefined]** **Moment**	

Table 12 Associations of "Time" Class

6.4 "Moment" Class

from Package Open Busines Repository.Semantic aspect.Reality.*Temporality*

A moment is a point on the time line.

It may be precisely defined, referring to a date and hour.

It may also be instantiated without mentioning a date, for further scheduling.

Name	Description
-> : **[0..*]** **Period**	A period may be linked to a moment or not. If there is no end, that means that the period applies from the start -if specified- and forever.
-> : **[0..*]** **Period**	A period may be linked to a moment or not. If there is no start, that means that the period covers the past till the end moment.
->date : [0..1] **Date**	
->time : [0..1] **Time**	

Table 13 Associations of "Moment" Class

7 "Ontology" Package

from Package Open Busines Repository.Semantic aspect.Reality

"Ontology" is to be interpreted as closely as possible to its original meaning: the science of being. This object domain encompasses the description of concrete objects, which the enterprise deals with. It includes a classification of beings. The model describes these beings without any assumption regarding the business and how to interact with them, so that this description is quite universal.

Name	Summary
Party	A person or an organization (whatever the role it assumes). Modeling decision: This class describes the real objects which are individuals or entities (organizations, groups, companies...). As such, the semantics is focused on the intrinsic features and excludes characteristics that are involved in relation to contexts or to other objects. For instance, the notion of a customer is not included in the semantics of Party, since it implies an association with a provider. Therefore, "customer" is a relative notion and has no place inside the "Reality" domain. See the consequences in terms of architecture (ref. OPS-02). DVAU-20090728.
Being	Being is a very generic concept, used as a root class for a great part of the classes in the Reality domain. It encompasses: - belongings, material assets, - human beings (for instance, relatives), - organizations... Modeling decision: This superclass and the associations linked to it allow us to summarize many other concepts and relations in order to provide an overall view. As an example, the motivation of an Opportunity can be expressed through this mechanism (see the documentation of the Opportunity class).
Asset	
Person	
Organizational Entity	

Name	Summary
Relation	Relation between two parties. Comment: The relation allows for linking two parties of any sort. The nature of the relation is selected from a typology, which includes family links (relatives), professional relations (e.g. employment, economical partnership), friendship, associations.... Modeling decision: Relation is an associative class. This form expresses the fact that no relation exist without a couple of identified parties.
Reachability	How a party can be reached during a given period. Modeling decision: This associative class is necessarily located in the same place as the Party class because the association is oriented. As the associative class relies on the class Party, it must access it and, therefore, cannot be exiled in the "Geography" model.

Table 14 Owned Classes of "Ontology" Package

7.1 "Party" Class

from Package Open Busines Repository.Semantic aspect.Reality.Ontology

Inherits from: Being

A person or an organization (whatever the role it assumes).

Modeling decision:
This class describes the real objects which are individuals or entities (organizations, groups, companies...). As such, the semantics is focused on the intrinsic features and excludes characteristics that are involved in relation to contexts or to other objects. For instance, the notion of a customer is not included in the semantics of Party, since it implies an association with a provider. Therefore, "customer" is a relative notion and has no place inside the "Reality" domain.
See the consequences in terms of architecture (ref. OPS-02).
DVAU-20090728.

Name	Description
involves-> : [0..*] Contact	By definition, a contact is an event that brings together two or more parties or more. That explains the cardinality of the association, on the Party side.
purchases-> : [0..*] Offer	A party (individual or organization) purchases an offer (product or service). Doing so, the party assumes the role of client. Modeling decision: The association is orientated to take into account the interdependency between both domains, "Catalogue" and "Reality".
connects with->relationRole_B : [0..*] Party	

Name	Description
connects with->relationRole_A : [0..*] Party	
raises->lead record : [0..*] Opportunity	The situation or behavior of a party raises an opportunity. Depending on the party's motivation, the opportunity can be a sales opportunity or a mere contact opportunity, which may itself lead to a sales opportunity. "Lead" - here, the name of the role on the association – refers to the second meaning of the term: the party involved in the opportunity. Discussion on cardinality: - Minimal cardinality: An opportunity necessarily identifies at least one party. Without any party, the opportunity would have no meaning (who to contact? ...). -Maximal cardinality: In the standard situation: there is only one party concerned by the opportunity. The model allows us to connect several parties. For instance, we can think of an opportunity raised on family grounds (e.g. birthday) and that could lead to contacting both parents or even the children.
drives->assigned event : [0..*] Event	This association allows us to connect an event to a party. The latter is considered as responsible for the event. It is part of the notion of responsibility that only one party may be responsible for a given event. When a person belongs to an organization, both of the person and the organization may be considered as responsible. This case is dealt with in the object diagram "Responsibility of a person or his/her organization". Some events may not be linked to a party. For example: external or temporal event.
assumes->duty : [0..*] Being	A party assumes a responsibility regarding a being.
owns->belongings : [0..*] Asset	A party - either an individual or an organization - owns assets. The association allows us to describe the material situation of a party. It can be used to keep track of the belongings whether they are insured or not. Among the assets that a party owns, those which are insured are identified thanks to another association. The proprietor of an asset may not be known. Hence the minimal cardinality. This allows for storing information regarding assets that are used for other purposes; for instance, assets that a party assumes responsibility of. In some cases, an asset may have many proprietors (e.g. a house owned by husband and wife). Hence the maximal cardinality.
/is in charge of->onus : [0..*] Being	A party may be considered as being in charge of a being. This association is not a direct one: it summarizes at least two other associations from Party: "assumes" and "owns". It provides an easy means for drawing up a recap of the party situation.

Name	Description
resides->location : [0..*] <u>Site</u>	A party, either a person or an entity, can be reached in one or several places (locations). The location can be qualified thanks to the qualifier "location nature".
->period : [0..1] <u>Period</u>	The cardinality allows for sharing the same Period instance with many actors. That can be useful for example in the case of opening hours of a team or agency...

Table 15 Associations of "Party" Class
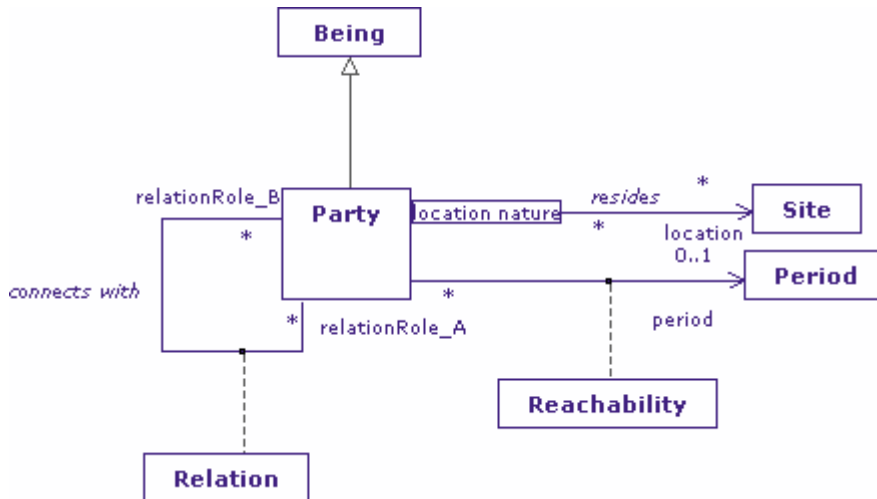
Figure 6 Party semantics

7.2 "Being" Class

from Package Open Busines Repository.Semantic aspect.Reality.<u>Ontology</u>

Being is a very generic concept, used as a root class for a great part of the classes in the Reality domain.

It encompasses:

- belongings, material assets,

- human beings (for instance, relatives),

- organizations...

Modeling decision:

This superclass and the associations linked to it allow us to summarize many other concepts and relations in order to provide an overall view.

As an example, the motivation of an Opportunity can be expressed through this mechanism (see the documentation of the Opportunity class).

Name	Description
level of knowledge : [1..1] Level of Knowledge	This attribute qualifies the level of certainty we have regarding the information of the being. See the documentation of the enumerated type.

Table 16 Attributes of "Being" Class

Name	Description
covers-> : [0..*] **Contract**	A contract covers
assumes->responsible : [0..*] **Party**	A party assumes a responsibility regarding a being.
/is in charge of->in charge : [0..*] Party	A party may be considered as being in charge of a being. This association is not a direct one: it summarizes at least two other associations from Party: "assumes" and "owns". It provides an easy means for drawing up a recap of the party situation.
motivates-> : [0..*] **Opportunity**	A being "motivates" an opportunity, that is that it provides the origin and ground for an opportunity. The association can be read in reverse order as: an opportunity "targets" a being.
->location : [0..*] Site	A being can be reached in one or several places (locations). The location can be qualified thanks to qualifier: "location nature".

Table 17 Associations of "Being" Class

Figure 7 Typology of beings

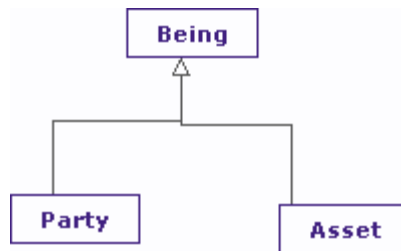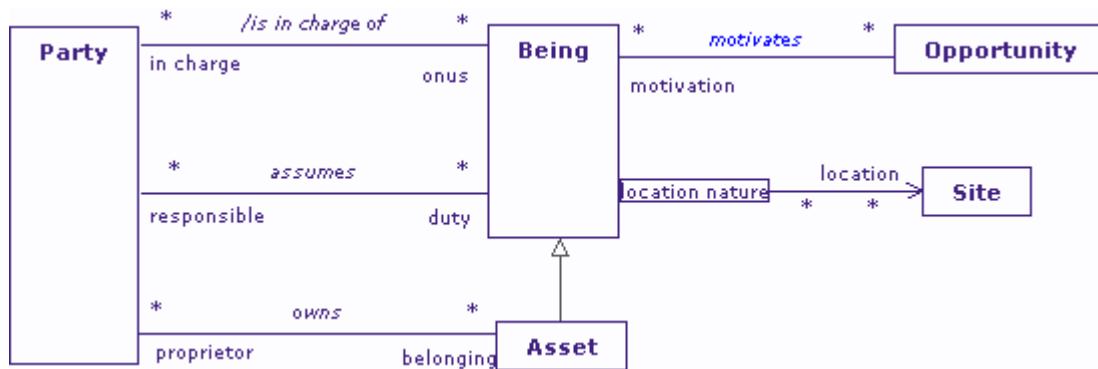*Figure 8 Semantics of Being*

7.3 "Asset" Class

from Package Open Busines Repository.Semantic aspect.Reality.Ontology

Inherits from: Being

Name	Description

Name	Description
owns->proprietor : [0..*] <u>Party</u>	A party - either an individual or an organization - owns assets. The association allows us to describe the material situation of a party. It can be used to keep track of the belongings whether they are insured or not. Among the assets that a party owns, those which are insured are identified thanks to another association.

Table 18 Associations of "Asset" Class

7.4 "Person" Class

from Package Open Busines Repository.Semantic aspect.Reality.<u>Ontology</u>

7.5 "Organizational Entity" Class

from Package Open Busines Repository.Semantic aspect.Reality.<u>Ontology</u>

7.6 "Relation" Class

from Package Open Busines Repository.Semantic aspect.Reality.<u>Ontology</u>

Relation between two parties.

Comment:
The relation allows for linking two parties of any sort. The nature of the relation is selected from a typology, which includes family links (relatives), professional relations (e.g. employment, economical partnership), friendship, associations....

Modeling decision:
Relation is an associative class. This form expresses the fact that no relation exist without a couple of identified parties.

7.7 "Reachability" Class

from Package Open Busines Repository.Semantic aspect.Reality.<u>Ontology</u>

How a party can be reached during a given period.

Modeling decision:
This associative class is necessarily located in the same place as the Party class because the association is oriented. As the associative class relies on the class Party, it must access it and, therefore, cannot be exiled in the "Geography" model.

Name	Description
->possible location : [0..*] Site	At a given period, a party may be reached in one or many places.

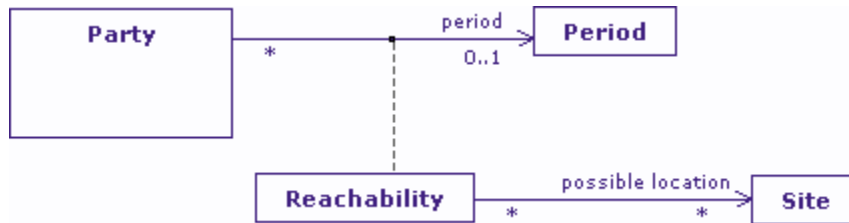*Table 19 Associations of "Reachability" Class*

Figure 9 Reachibility

The reachability is specified for one actor at a given period.

The period can be:

- at any time (no beginning, no end);

- complex (a combination of aggregated periods);

- opening hours.

For more detail, please refer to the semantics of Period.

8 "Phenomenology" Package

from Package Open Busines Repository.Semantic aspect.*Reality*

Other candidate name: "actuality".

Part of the reality that manifests itself through phenomena, events, actions...
As opposed to the "static" part that is represented in the "Ontology" domain.

This object domain gathers the notions connected to events (something that is happening).

Modeling decision:
It is important to separate these sorts of objects from the "beings", since their life spans and behaviors greatly differ. Events may be recorded for a shorter time than parties... They are processed through quicker actions, within a short term horizon.

Name	Summary
Event	Something that happens, happened or may happen. The class covers any event of any kind that may occur in the future or which has happened in the past. An event may be identified even without any date: an action is defined but not yet scheduled. Therefore, there is no date defined on the class. Scheduling an event results in linking it to a period.
Opportunity	An opportunity is a situation that makes it possible to do or achieve something, generally an action, a proposal or a sale.
Action	
Request	
Contact	A contact is an event bringing together at least two parties.

Table 20 Owned Classes of "Phenomenology" Package

8.1 "Event" Class

from Package Open Busines Repository.Semantic aspect.Reality.*Phenomenology*

Something that happens, happened or may happen.
The class covers any event of any kind that may occur in the future or which has happened in the past.

An event may be identified even without any date: an action is defined but not yet scheduled.
Therefore, there is no date defined on the class. Scheduling an event results in linking it to a period.

Name	Description
object : [1..1] string	

Name	Description
communication channel : [1..1] Medium	The communication channel informs on the medium used or to be used for actualizing the event.

Table 21 Attributes of "Event" Class

Name	Description
reveals->generated interest : [0..*] Opportunity	An event may reveal (be source of) an opportunity. The association allows to keep the memory of what provoke the opportunity.
drives->responsible : [0..1] Party	This association allows us to connect an event to a party. The latter is considered as responsible for the event.
->period : [0..*] Period	
follows->next : [0..*] Event	The "follows" association allows us to build up chains of events and to keep traces of the history (of actions, contacts...). The model allows for identifying many events as sources for a given one. For example, an appointment may result from the occurrence of several events like a family birthday, a new product to be proposed, a campaign... So, the chain of events is not a tree but a network. It may be worth visualizing this determination when dealing with an event.
follows->previous : [0..*] Event	The "follows" association allows us to build up chains of events and to keep traces of the history (of actions, contacts...). An event may trigger many other events.

Table 22 Associations of "Event" Class
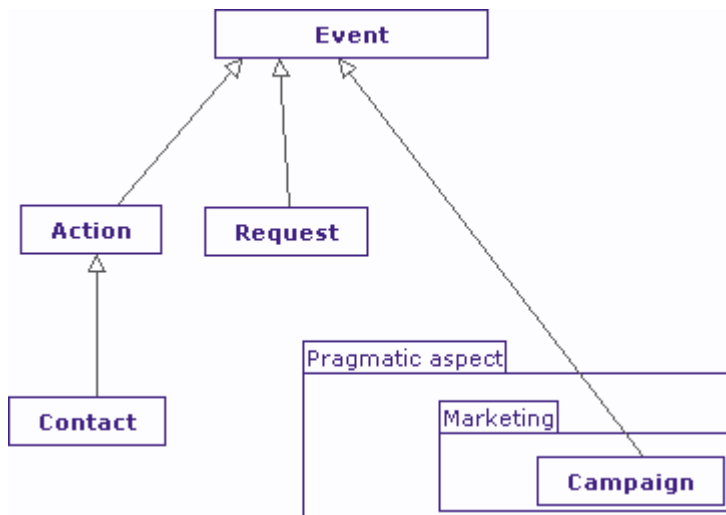
Figure 10 Event typology

All the notions represented on this figure share a common semantic. Every property that the Event class bears makes sense for the subclasses. For instance, the temporality of the event is rendered by its association toward Period and is of use on more specific concepts like action and campaign.

The typology can be extended. Each time a subclass is added on this inheritance tree we must check that every property of Event applies to the new concept.

The diagram shows the Campaign class, which is located in another domain inside another aspect. Indeed the campaign notion pertains to marketing. Nevertheless, it is possible and recommended that such a specialized concept benefits from a more generic one.
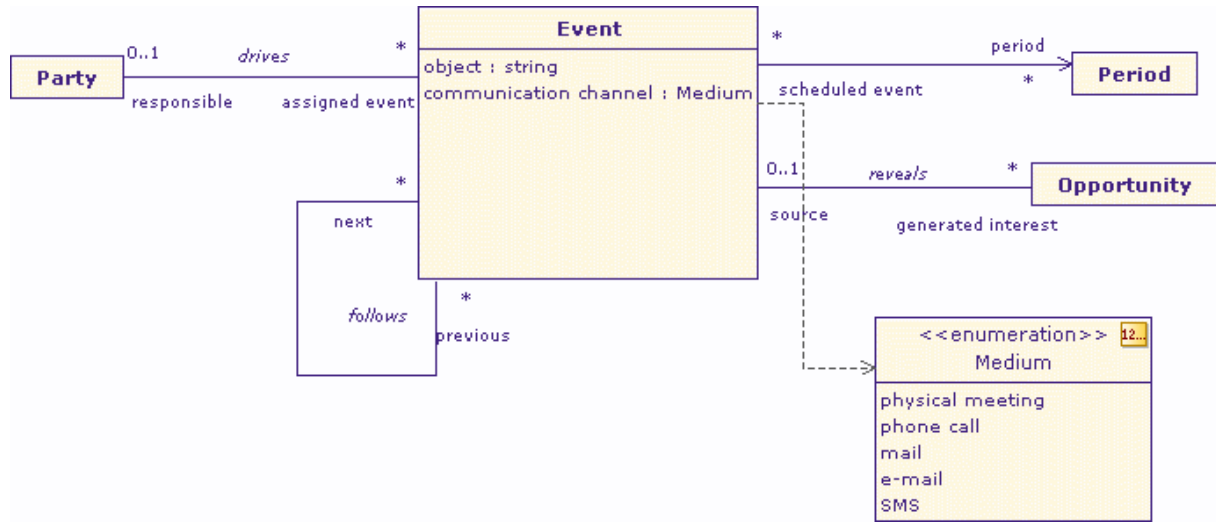
Figure 11 Event semantics

Event is a generic concept, covering whatever types of events as well as contacts, actions, operations, campaigns...
By connecting other objects to an event, it is possible to articulate its meaning.
An event can be defined before any scheduling decision. It is the reason why there is no date attribute in the class. The association with Period affords the ability to schedule. The period semantics allow many combinations.
For the sake of lead management, Event is connected to Opportunity.
The "follows" reflexive association is for keeping the chains of events that makes up history (of a customer relation, of a project, of a portfolio, etc.).
The nature of an event - i.e. the communication channel used - characterizes the means.
The event may be assigned to a party

Figure 12 Event examples

This object diagram is based on the following scenario:
1 The Marketing team launches a campaign.
2 The campaign leads to a contact: a prospect calls an agent or visits a web site or sends back a form.
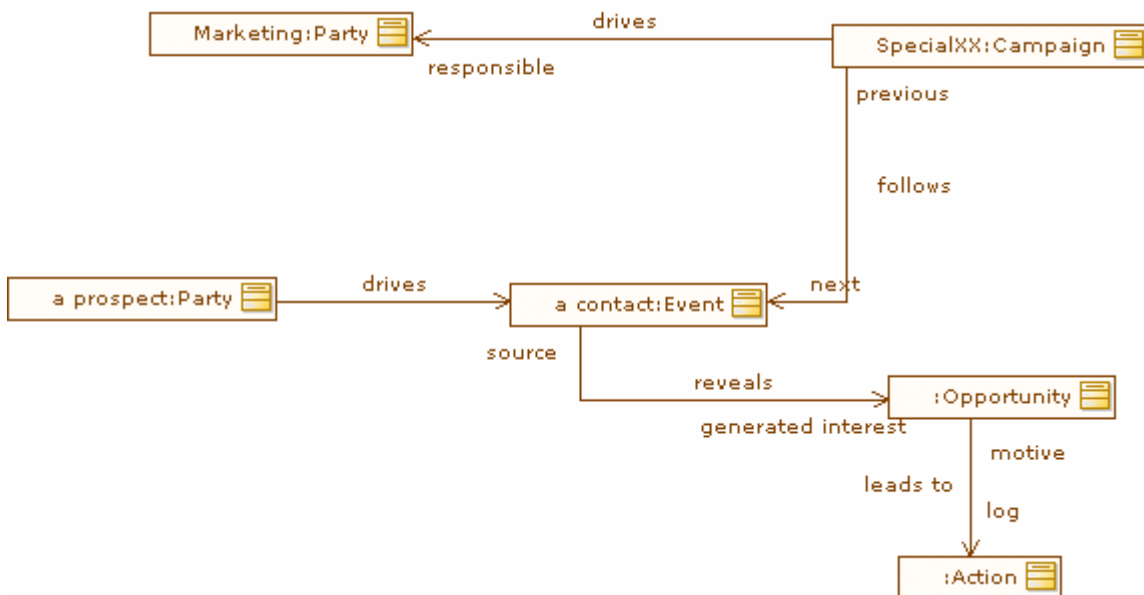3 Through the contact, enough information are gained for establishing an opportunity.
4 The lead management process continues with actions...

The model allows for creating all necessary objects and for connecting them via associations, so that such a scenario can be covered and the information captured.
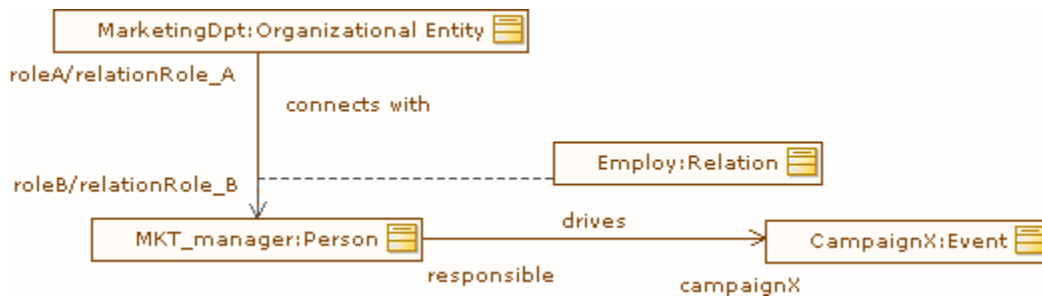
Figure 13 Responsibility of individuals and organizations

Organizational Entity and Person are subclasses of Party.
At one stage, an entity - e.g. Marketing department - may be considered as collectively responsible of a planned event. Then, an individual is tasked with the organization of the event. At that time, only the link between the person and the event is established. It is always possible to retrieve the events under the entity control, by navigating across the associations:
- "connects with";
- "drives".

State Machine "Automate for Event class"

At this level of genericity, a couple of states can be defined. Most of them are related to temporality and we have to take into account the fact that there is no date attribute on the Event class (see definition of Event).

As Event is a superclass, the content and logic of its automate must be consistent with the automates of the subclasses (Action, Contact, Campaign...).
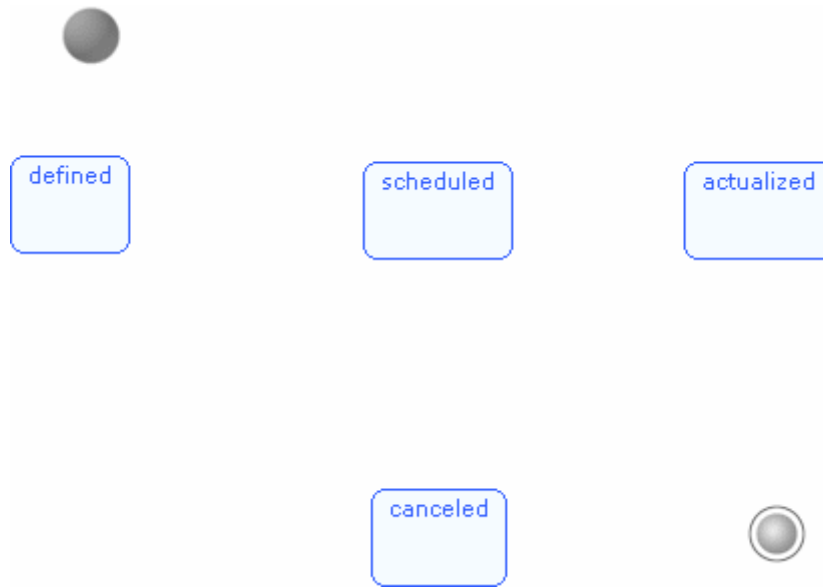
Figure 14 Life cycle of an Event

This state diagram is a draft. It only shows candidate states.

State Name	State Description
defined	
scheduled	
actualized	
canceled	
FinalState	
postponed	
obsolete	

Table 23 Automate for Event class

8.2   "Opportunity" Class

from Package Open Busines Repository.Semantic aspect.Reality. Phenomenology

An opportunity is a situation that makes it possible to do or achieve something, generally an action, a proposal or a sale.

Name	Description
Amount consolidated opportunity cost (IN during Period,IN by Party)	Potential lost value for the scope defined by the parameters.

Name	Description
qualify ()	This operation automatically calculates the potential value of the opportunity. It values the attributes: - potential; - opportunity cost; - success probability (if undetermined; if not, that means that it has been valued by a human actor. In this case, the value is kept). The level of knowledge is set at "uncertain". It will change when actions are taken.
estimate ()	The operation calculates the value of the "potential" attribute. The algorithm explores the motivations of the opportunity (see the diagram "Motivation of an Opportunity" and its comments). First of all, when there are offers linked to the opportunity as "potential interests", the potential is the sum of the prices. Assumptions needed for establishing the prices are based on the knowledge of the situation (starting with the party information). Secondly, when there is no "potential interest" expressed in terms of identified offer, an estimation can be provided based upon the beings that are connected to the opportunity (role "motivation" on the association "motivates"). In case there is neither "motivation" nor "potential interest", the estimation ends up with a null value.
assign (IN to Party)	Allows for linking the opportunity with the party in charge of it. Modeling decision: This operation can only be placed on this class. Indeed, due to the dependency between the sub-domains "Phenomenology" and "Ontology", Party could have no knowledge of Opportunity.
remove ()	

Table 24 Operations of "Opportunity" Class

Name	Description
comment over motivation : [1..1] Text	
potential : [1..1] Amount	The potential of an Opportunity is the financial value that may be expected in case of a successful conclusion. The amount of effort for processing the opportunity is to be balanced with its potential. NB: the value of "potential" can be set by use of the "estimate" operation. It can also be set by manually forcing it, when the person in charge gains information.
level of knowledge : [1..1] Level of Knowledge	This attribute qualifies the level of certainty we have regarding the information of the opportunity. See the documentation of the enumerated type. This attribute is used in lead management: a) for establishing priorities of actions; b) for anticipating the potential return of the activity.
opportunity cost : [1..1] Amount	Potential value that will be given up as a result of not dealing with the opportunity.

Name	Description
success probability : [1..1] integer	The likelihood for the opportunity to succeed and lead to a contract. Expressed as a percentage.
is of no value : [1..1] boolean	This derived attribute returns "true" if: Opportunity.potential * Opportunity.probability < $Opportunity.valueThreshold.
value threshold : [1..1] string	Set the value below which no action needs to be taken. This is a class attribute, meaning that this value is for the class. It is used as a paramater in some operations.

Table 25 Attributes of "Opportunity" Class

Name	Description
concerns->potential interest : [0..*] Offer	An opportunity may concern one or several offers. Modeling decision: The name of the association reads from Oppotunity to Offer but the association is orientated from Offer to Opportunity. Indeed, Offer is located inside the Catalogue domain, which depends on the Reality domain (see the General Architecture Dossier for further discussion; ref. OPS-02). A given offer may be of interest through many opportunities (we hope so).
reveals->source : [0..1] Event	An event may reveal (be source of) an opportunity. The association allows to keep the memory of what provoke the opportunity.
raises->lead : [1..*] Party	The situation or behavior of a party raises an opportunity. Depending on the party's motivation, the opportunity can be a sales opportunity or a mere contact opportunity, which may itself lead to a sales opportunity.
entails->effect : [0..*] Opportunity	Under certain circumstances, an opportunity may lead to another opportunity. For instance, an opportunity has been generated by a datamining operation. When qualifying it, the agent may discover other features of the party situation and take advantage for proposing other offers. They will not do it necessarily at the same time. So, they store the secondary opportunity for further use. Modeling decision: Check that this association is not redundant with the reflexive association on the Event class.
entails->cause : [0..*] Opportunity	Under certain circumstances, an opportunity may lead to another opportunity. For instance, an opportunity has been generated by a datamining operation. When qualifying it, the agent may discover other features of the party situation and take advantage for proposing other offers. They will not do it necessarily at the same time. So, they store the secondary opportunity for further use. Modeling decision: Check that this association is not redundant with the reflexive association on the Event class.

Name	Description
leads to->log : [0..*] **Action**	An opportunity - if qualified and followed - may lead to contacts. A contact may be independent from any stored opportunity. Hence the minimal cardinality. In some cases, several opportunities may be gathered and lead to a single contact.
->validity period : [0..1] **Period**	Some opportunities may present a limited time window, due to the situation of the party or to the conditions of the offer. This validity period is used when planning actions. The minimal cardinality is '0' since period instances may exist in other contexts that lead management. Modeling decision: The orientation of the association obeys the dependency between the packages. A given period - as an object - may be linked to many opportunities. The case occurs when the validity period belongs to the conditions of the offer that generates the opportunities. Therefore, a change of these conditions is limited to updating the period instance.
motivates->motivation : [0..*] **Being**	A being "motivates" an opportunity, that is that it provides the origin and ground for an opportunity. The association can be read in reverse order as: an opportunity "targets" a being.

Table 26 Associations of "Opportunity" Class

Figure 15 Opportunity semantics
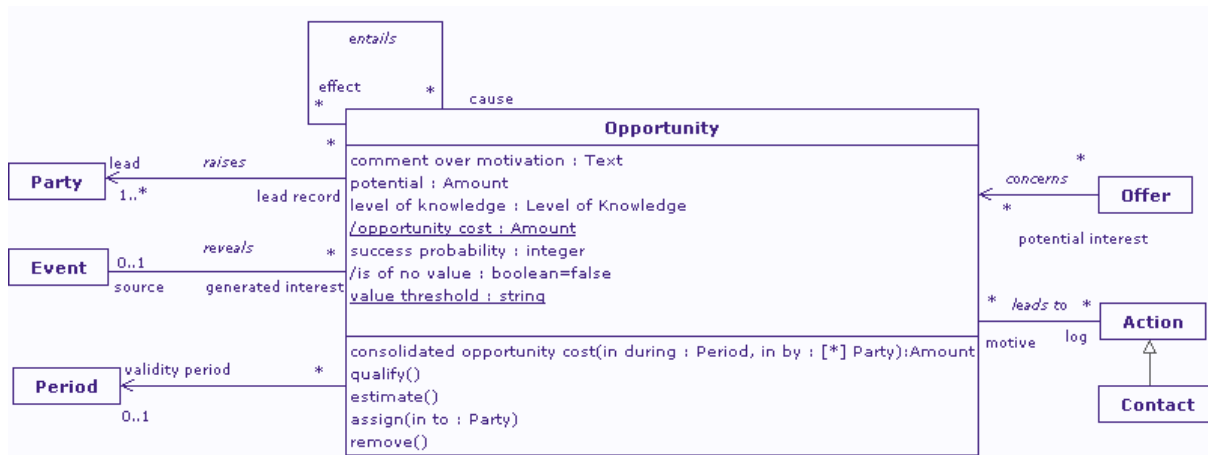
This class diagram shows the connections of the Opportunity notion.

An opportunity involves a party. In some cases, an offer already is identified. The history of the opportunity is kept thanks to the event notion. The lead management aims to take advantage of the opportunity and transform it into contacts. Many contacts and actions may occur before the conclusion of the opportunity.
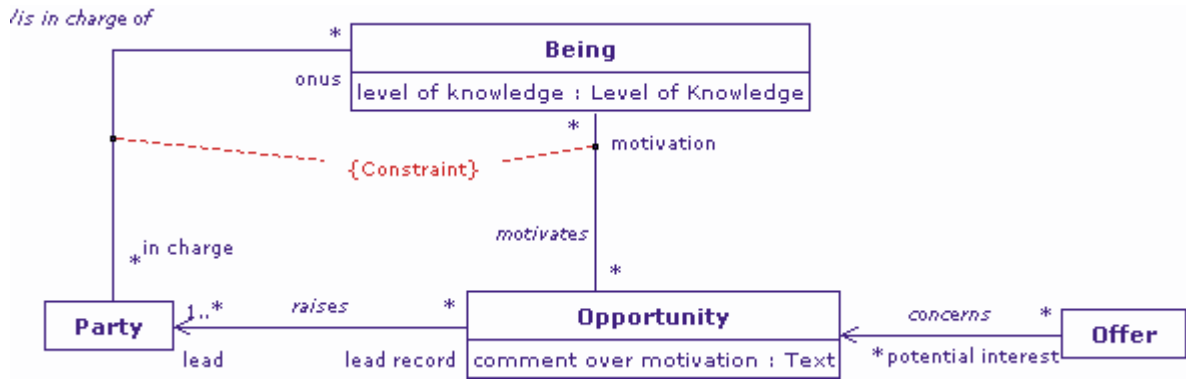
Figure 16 Motivations of an Opportunity

One of the key factors of a successful lead management lies in the attention granted to the motivations of the party involved in the opportunity.

To materialize this attention, the model allows these means:

a) a text attribute where a free comment can be stored;

b) a link to an identified offer or set of offers;

c) a link to an element of the party situation (belongings, responsibility...).

Depending on the origin of the opportunity, one of several of these means may be used in order to clarify the motivations.

Modeling decision:

A constraint is set so that the being that motivates an opportunity is necessarily linked to the party in charge.

The party that arouses the opportunity is not necessarily the one that is in charge of the targeted being. That is the reason why the constraint doesn't include the "arouses" association.
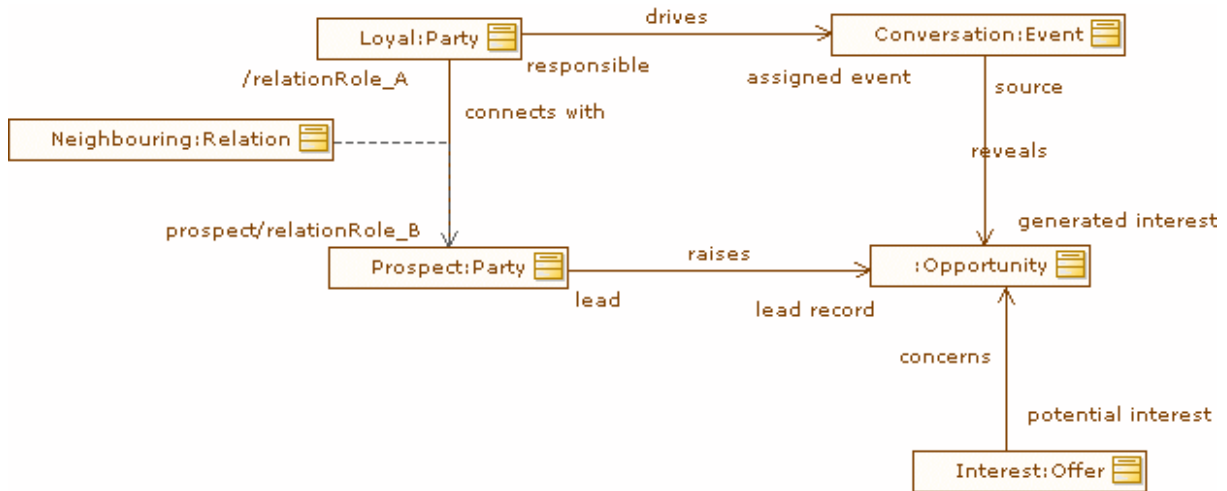
Figure 17 Generation of an Opportunity by a third party

This object diagram sets the stage for a scenario as follows:

1. During a conversation (instance of Event - it could be a Contact), a current customer (Mr. Loyal, represented by an instance of Party) evokes their neighbour who could be interested by an offer.

2. The sales person records the conversation as well as the information gained regarding the potential prospect (another instance of Party).

3. He/she creates an instance of Opportunity that will trigger the actions.

4. At a moment or another (possibly during the first conversation or later on through specific actions), the sales person is able to specify the prospect's interest.

5. The story of the opportunity can be followed thanks to its state machine: qualified, scheduled, successful...

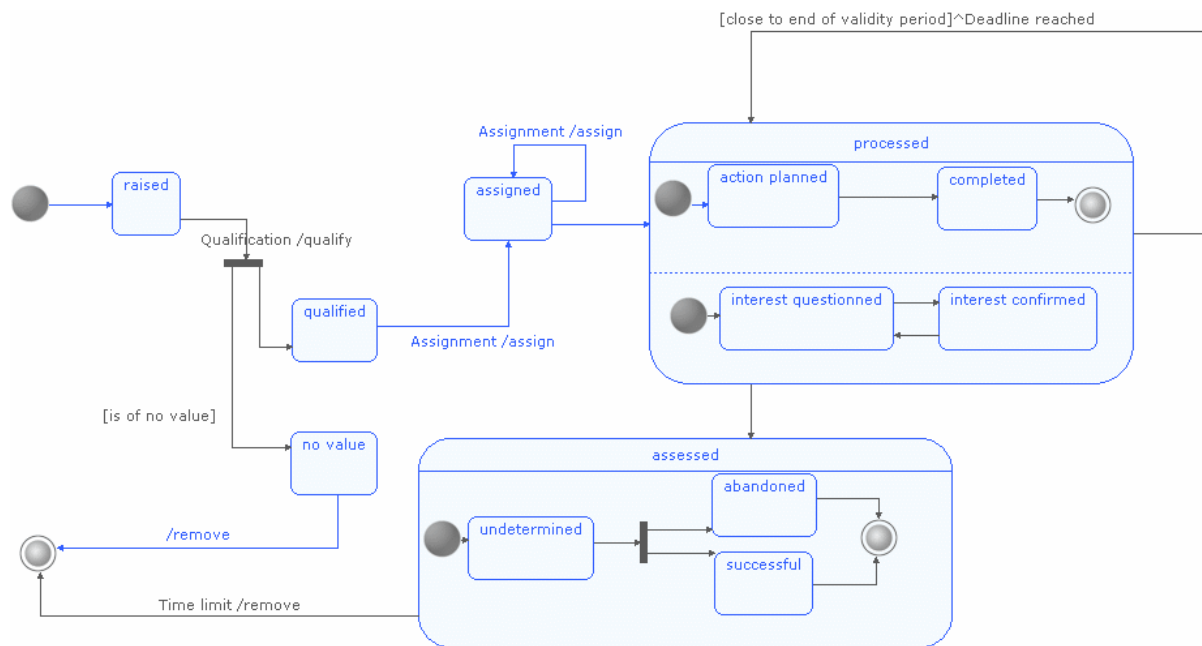### State Machine "Life cycle of an Opportunity"

Figure 18 Overview of the Opportunity life cycle

In the current version of the model, this state machine has to be considered only as a first attempt for describing the life cycle of an opportunity.

As a tool for expressing the business knowledge, the state diagram entails important consequences: it drives the behaviour of the object, thus conditioning the entire process of lead management.

Therefore, more time and a greater deal of attention has to be spent on the topic, so that the state machine takes into account the business fundamentals as well as the disturbances that may occur in the process.

State Name	State Description
raised	An instance of Opportunity has been created, whatever the value of this opportunity. At least one Party is connected to the Opportunity via the "arouses" association (see the cardinality of the association).

State Name	State Description
qualified	The value of the Opportunity has been assessed. Several factors may determine the potential value of the opportunity: - likelihood of signing a contract (based on the knowledge we have regarding the party's situation); - amount gained in case of a deal (use the price of the offers designated as "postential interest" (see association "concerns"); - possible directives of sales policy or marketing strategy. The attributes "potential" and "opportunity cost" are valued.
assigned	The opportunity has been assigned to a party (either a person or an entity: an agent, a team, an agency, a company...). The assignment may change in the course of the Opportunity process (for instance from a team or a manager to a subordinate).
processed	After assignment, the life cycle continues with actions.
action planned	One or several actions are created and linked to the opportunity.
completed	No need for other actions
FinalState	
interest confirmed	Thanks to new information gained through contacts, the level of knowledge can be raised to "confirmed" as well as the probability can be better appreciated.
interest questionned	Before contacting the lead party, there is still a doubt cast about the value of the opportunity.
no value	After qualification, the opportunity may be set apart as of no value.
FinalState	
assessed	The opportunity has been analyzed after a while and its sequels are determined.
successful	There is a positive result of any sort (not necessarily a sale - it can be a gain in confidence or in information or another opportunity...). When positionned in this stage, the opportunity is updated with the right potential and the level of knowledge is valued to "certain".
abandoned	The opportunity leads to no visible benefit.
undetermined	It is not possible yet to determine the result of the opportunity treatment. Possible causes include: - waiting for the prospect's answer; - waiting for the results of other events or opportunities that have been triggered in the course of the interaction...
FinalState	

Table 27 Life cycle of an Opportunity

8.3 "Action" Class

from Package Open Busines Repository.Semantic aspect.Reality.Phenomenology

Inherits from: Event

Name	Description
leads to->motive : [0..*] Opportunity	An opportunity - if qualified and followed - may lead to contacts.

8.4 "Request" Class

from Package Open Busines Repository.Semantic aspect.Reality.*Phenomenology*

Inherits from: <u>Event</u>

8.5 "Contact" Class

from Package Open Busines Repository.Semantic aspect.Reality.*Phenomenology*

Inherits from: <u>Action</u>

A contact is an event bringing together at least two parties.

Name	Description
involves->participant : [2..*] <u>Party</u>	By definition, a contact is an event that brings together two or more parties or more. That explains the cardinality of the association, on the Party side.

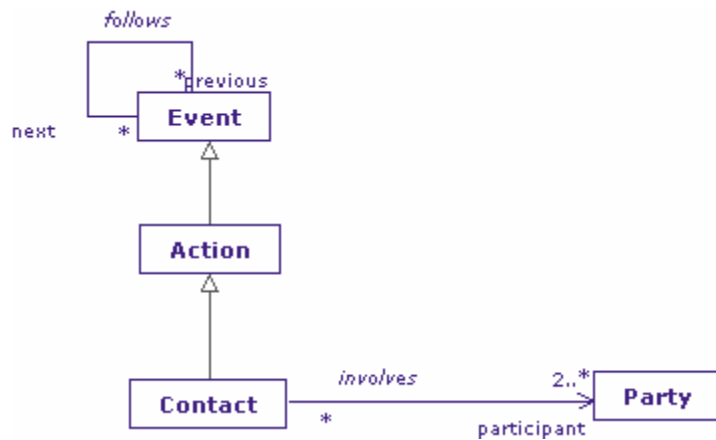*Table 29 Associations of "Contact" Class*

Figure 19 Semantics of Contact

The origin of a contact and previous actions are traced thanks to the association "follows" that the subclass Contact inherits.

The temporality of Contact is also inherited. It is thus possible to provide the story that leads to a contact and, from that, to recap the people who were involved.

About the cardinality of participant role, please refer to the documentation of the association "involves".

9 "Catalogue" Package

from Package Open Busines Repository.*Semantic aspect*

The catalogue describes the products and services that the company delivers.

The root is the generic class "Offer", with two branches: "Product" and "Service". The maximum of properties are factorized into these generic classes.
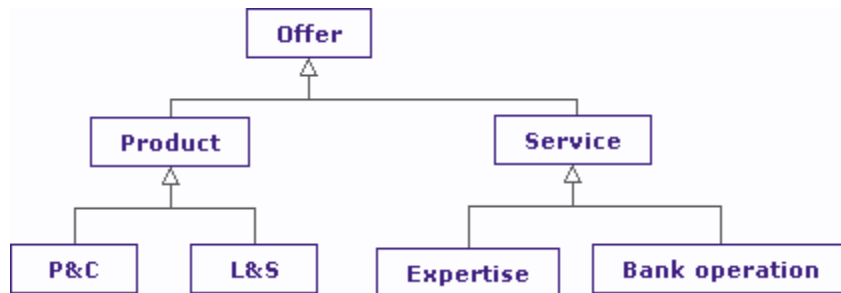

Figure 20 Content of the "Catalogue" domain

An important decision is the integration of all businesses (e.g. Life & Saving, Products & Casualties, and bank).

Doing so enables us to factorize many relations from and to this object domain.

This decision applies to the description of the business knowledge. It does not follow that the same decision will be relevant in matters of IT system. That will be discussed about the logical aspect and the software solution.
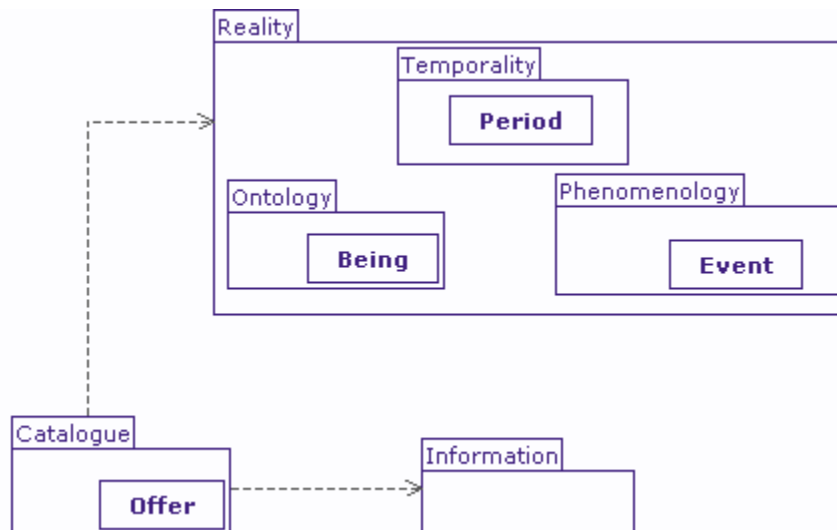

Figure 21 Postionning the "Catalogue" domain

The "Catalogue" domain refers to the "Reality". Indeed, the description of products mentions characteristics of real-life objects. For instance, a certain type of insurance product applies to a certain type of people or objects (beings). The type is part of the semantics of an object or concept and it must be located at the same place.

The description of an also Offer embeds notions of temporality and event. That reinforces the need for a dependency between both object domains.

There is no other dependency from the "Catalogue" domain.

"Catalogue" also refers to "the "Information" package where it finds notions like "Amount".

Modeling decision:
It is remarkable that this connection between "Catalogue" and "Reality" applies at a generic level: not from a given policy to a real object, but from a product to a type of object.
Based upon this remark, the dependency between "Catalogue" and "Reality" will be removed in the logical model.

Name	Summary
Offer	Offer is a generic concept that covers products and services. An offer can be anything a party can provide to another one, either it sells it or not. Comment: Offer allows to record the group catalogue but also part of the products and services that partners or parties in the supply may provide.
Product	
Service	
L&S	
P&C	
Bank operation	
Expertise	

Table 30 Owned Classes of "Catalogue" Package

9.1 "Offer" Class

from Package Open Busines Repository.Semantic aspect.<u>Catalogue</u>

Offer is a generic concept that covers products and services. An offer can be anything a party can provide to another one, either it sells it or not.

Comment:
Offer allows to record the group catalogue but also part of the products and services that partners or parties in the supply may provide.

Name	Description
purchases->client : [0..*] <u>**Party**</u>	A party (individual or organization) purchases an offer (product or service). Doing so, the party assumes the role of client. Modeling decision: The association is orientated to take into account the interdependency between both domains, "Catalogue" and "Reality".
concerns-> : [0..*] <u>**Opportunity**</u>	An opportunity may concern one or several offers. Modeling decision: The name of the association reads from Oppotunity to Offer but the association is orientated from Offer to Opportunity. Indeed, Offer is located inside the Catalogue domain, which depends on the Reality domain (see the General Architecture Dossier for further discussion; ref. OPS-02). An opportunity may concern one or several offers. It is also possible that we keep trace of an opportunity without being able to identify an offer. That is the reason why the minimal cradinality is '0'.

Table 31 Associations of "Offer" Class

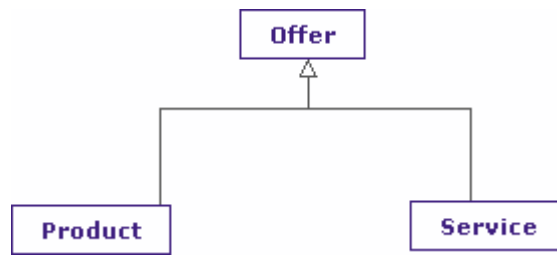*Figure 22 Offer semantics*

9.2 "Product" Class

from Package Open Busines Repository.Semantic aspect.<u>Catalogue</u>

Inherits from: <u>Offer</u>

9.3 "Service" Class

from Package Open Busines Repository.Semantic aspect.<u>Catalogue</u>

Inherits from: <u>Offer</u>

9.4 "L&S" Class

from Package Open Busines Repository.Semantic aspect.<u>Catalogue</u>

Inherits from: <u>Product</u>

9.5   "P&C" Class

from Package Open Busines Repository.*Semantic aspect.*<u>*Catalogue*</u>
Inherits from: <u>Product</u>

9.6   "Bank operation" Class

from Package Open Busines Repository.*Semantic aspect.*<u>*Catalogue*</u>
Inherits from: <u>Service</u>

9.7   "Expertise" Class

from Package Open Busines Repository.*Semantic aspect.*<u>*Catalogue*</u>
Inherits from: <u>Service</u>

10 "Portfolio" Package

from Package Open Busines Repository.Semantic aspect

The portfolio makes the connection between objects of the reality – mainly parties – and the enterprise offering.

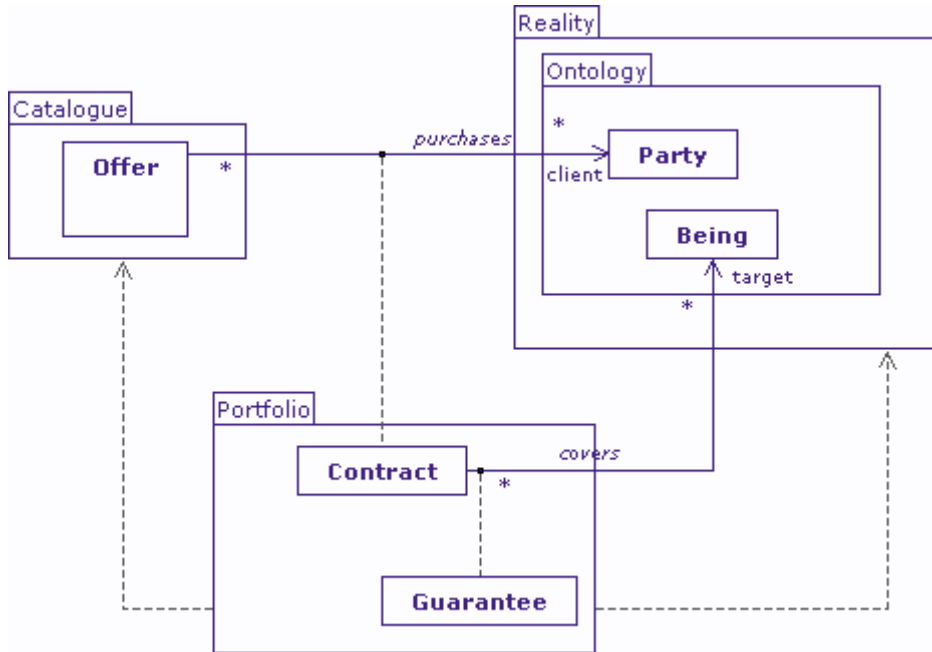The contract notion establishes the conjunction.

Figure 23 Justification of the dependencies from "Portfolio " domain

The dependencies from this domain reflect the relations that link its core notion of contract with the classes of "Catalogue" and "Reality".

The Contract class needs access to Offer and Party, since it refers to both classes for establishing a contract. It also needs to link with a being (a party or an asset).

Name	Summary
Contract	
Guarantee	

Table 32 Owned Classes of "Portfolio" Package

10.1 "Contract" Class

from Package Open Busines Repository.Semantic aspect.Portfolio

Name	Description
covers->target : [0..*] <u>Being</u>	A contract covers

Table 33 Associations of "Contract" Class

State Machine "Machine à état de protocole pour Contract"

State Name	State Description
proposed	NB: the modeling choice here is to use the notion of contract associated with its state rather than to introduce another notion, with intermediary existence, like Proposal. Refer to the documentation of the Contract class.
signed	
abandoned	

Table 34 Machine à état de protocole pour Contract

10.2 "Guarantee" Class

from Package Open Busines Repository.Semantic aspect.<u>Portfolio</u>

11 "Production" Package

from Package Open Busines Repository. *Semantic aspect*

Based on our knowledge of real objects and all along the relation established through contracts, the company follows the events that occur, emits information or invoices, and delivers services… This activity and the information it produces, live at a different pace, compared to the information compiled in the previous domains. That justifies defining a separate object domain. "Production" encompasses this day-to-day information. The objects inside this domain are arranged around these main objects:

- Premium,
- Claim,
- Operation (e.g. survey, assessment, service delivery),
- Payment.

12 "Information" Package

from Package Open Busines Repository.<u>*Semantic aspect*</u>

This package is not an object domain - strictly speaking - but a set of facilities, including the definition of general types.
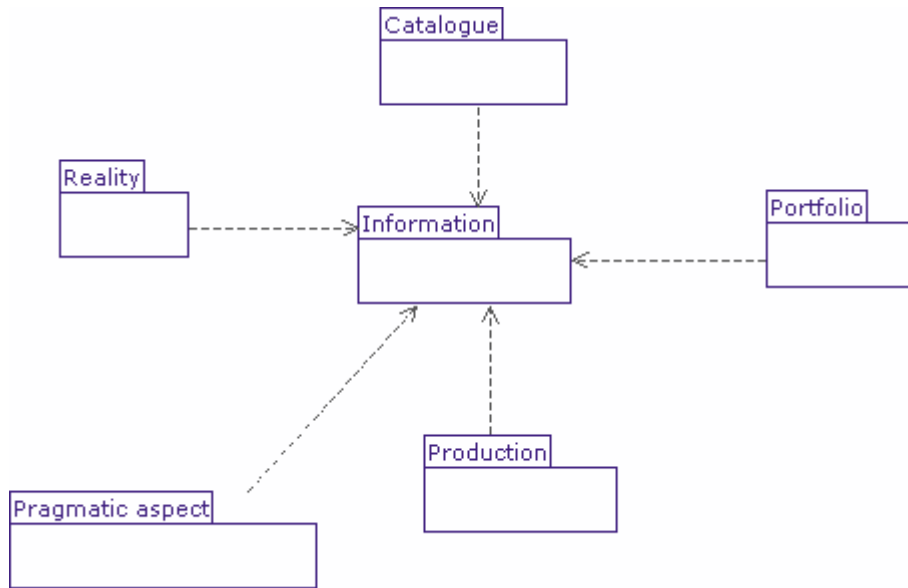

Figure 24 diagramme de classe de Information

The "Information" package provides the rest of the model with general types. That explains that every object domain maintains a dependency toward "Information".

Modeling decision:
These dependencies will be reviewed when it comes to logical architecture.

Name	Summary
<u>Text</u>	This type is for storing and manipulating texts. It includes basic operations like merging texts, evaluating the volume, checking vocabulary, etc.
<u>Amount</u>	This class encapsulates the whole sematics of financial amount. It includes: - the absolute value of the amount; - its currency; - operations for converting the amount into other currencies; - time information that are required for a proper conversion... The class is to be used over the entire business representation. Only for specific and documented reasons could basic types be introduced rather than using the Amount class.

Table 35 Owned DataTypes of "Information" Package

Name	Values	Description
Level of Knowledge	unknown uncertain confirmed certain assessed	This enumerated type allows to qualify the information related to the environment (about a party or its belongings and situation...). Qualifying information makes it possible to record information with a varying level of confidence into the same solution. Modeling decision: This type is used as the nature of several attributes through the model (e.g. Being class). A more sophisticated solution would be to handle the information qualification in the shape of an automate. This way, a change in the perceived quality of information could trigger other changes in the system. For now, the simplest solution is preferred.
Medium	physical meeting phone call fax mail e-mail SMS	The medium is the type of communication channel.
Language		This codification appears in a number of qualifyers and is part of the solution for handling multilinguism. Modeling decision: Inside the semantic model, the language notion is represented by an enumerated type. Nevertheless, since the list of languages cannot be established as a closed and stable list, it will be derived as a codification table, inside the logical model.

Table 36 Owned Enumerations of "Information" Package

12.1 "Text" DataType

from Package Open Busines Repository.Semantic aspect.Information

This type is for storing and manipulating texts.
It includes basic operations like merging texts, evaluating the volume, checking vocabulary, etc.

12.2 "Amount" DataType

from Package Open Busines Repository.Semantic aspect.Information

This class encapsulates the whole sematics of financial amount.
It includes:
- the absolute value of the amount;
- its currency;
- operations for converting the amount into other currencies;

- time information that are required for a proper conversion...

The class is to be used over the entire business representation.
Only for specific and documented reasons could basic types be introduced rather than using the Amount class.

13 Class index

- o [Action](#)
- o [Address](#)
- o [Asset](#)
- o [Bank operation](#)
- o [Being](#)
- o [Contact](#)
- o [Contract](#)
- o [Country](#)
- o [Date](#)
- o [Event](#)
- o [Expertise](#)
- o [Guarantee](#)
- o [L&S](#)
- o [Moment](#)
- o [Offer](#)
- o [Opportunity](#)
- o [Organizational Entity](#)
- o [P&C](#)
- o [Party](#)
- o [Period](#)
- o [Person](#)
- o [Phone Number](#)
- o [Product](#)
- o [Reachability](#)
- o [Relation](#)
- o [Request](#)
- o [Service](#)
- o [Site](#)
- o [Time](#)