Requirements Engineering Journal manuscript No.
(will be inserted by the editor)

A Cognitive Semantics for the Association Construct

Joerg Evermann[1]

Memorial University of Newfoundland
Faculty of Business Administration
310 Elizabeth Avenue, St. John's, Canada
+1 709 737 8527
e-mail: jevermann@mun.ca

The date of receipt and acceptance will be inserted by the editor

Abstract The Unified Modelling Language (UML), besides its traditional use in describing software artifacts, is increasingly being used for conceptual modelling, the activity of describing an application domain. For models to be clear and unambiguous, every construct of the modelling language must have well-defined semantics, which is its mapping to elements of the semantic domain. When used for conceptual modelling, the semantic domain of UML is the application domain, as perceived by the modeller. Modellers perceive and structure their perceptions using cognitive concepts. This paper proposes a mapping of the UML association construct to those concepts. Implications for the use of the association construct for conceptual modelling are derived, a UML profile for conceptual modelling is presented, along with the results of a case study using the semantics and profile.

Key words Object-oriented modelling – Associations – Natural language – Semantics – Cognition – Psychology

1 Introduction

Conceptual modelling is the description of an application domain, not of a software artifact, using formal or semi-formal modelling languages [1]. Good conceptual modelling, i.e. a clear and unambiguous description of the domain, reduces risks to system implementation [2], facilitates requirements engineering [3, 4], and reduces costly re-work later in the development process [5].

One prerequisite for good conceptual models is a well-defined semantics for the language that is used. This paper follows the notion of semantics of Harel and Rumpe: "A sound language definition must relate the syntactic expressions to the semantic domain elements so that each syntactic creature maps to its meaning" [6, p. 68]. "A language's semantics must provide the meaning of each expression,

(a) Association example

```java
class University {
    Student[]    attends;
    ...
}
class Student {
    University   attends;
    ...
}
```

(b) Corresponding Java implementation

Fig. 1 Example of UML – Java mapping

and that meaning must be an element in some well-defined and well-understood [semantic] domain" [6, p. 67].

Object-oriented modelling languages such as UML have originally been developed to describe software artifacts [7, 8]. For this purpose, their semantic domain is the set of programming language constructs. Here, the semantics of the language are generally well-defined and well-understood[1]. For example, the association construct, central to UML modelling, is mapped in C++ to pointers that point to objects or object arrays, and is mapped in Java to variables of reference types, as in the example in Fig. 1.

More recently, UML has been used for conceptual modelling of application domains [11, 12]. However, when used for conceptual modelling, the semantics of the association construct are unclear. For example, in a typical business application domain, one may find concepts such as 'Person', 'Business Process', 'Business Plan', 'Customer', 'Strategy Vision', 'Mission Statement', 'Order', 'Sale', etc. Here, it is not immediately clear what associations should and should not refer to. For example, does an association named 'order' refer to an order form, an order activity, or a particular state in the world? Does it refer to past or present order activities, to actual or projected order activities? The literature on associations is not helpful: An association is "the simplest form of a relationship" [13, p. 195], an "association represents the relationships between objects and classes" [14, p. 26], "An association specifies a semantic relationship that can occur between typed instances" [15, p. 39]. The concept of a relationship is no clearer than that of association. *Consequently, the central research problem that this paper addresses is to identify those concepts in an application domain that can serve as the semantic domain of UML associations, for purposes of conceptual modelling.*

While UML associations technically include meronymic relationships ("part-of") by virtue of different association end aggregation types, such as composition

[1] Some ambiguities and difficulties remain [9, 10].

and aggregation, these are excluded from the scope of this paper. Meronymic relationships are discussed in [16, 17, 18, 19, 20] (see also Section 7.2). Furthermore, while language syntax and semantics may have an effect on development methodology and processes, this paper focuses on the language aspect, leaving the methodology or process aspect for future research.

The remainder of the paper proceeds as follows. Section 2 presents the cognitive semantics for the association construct and proposes a UML profile to represent these semantics. This is followed in Sect. 3 by an identification of further cognitive concepts that describe important aspects of an application domain and that should be made explicit on associations. The UML profile is then applied to three examples in Sect. 4. An example from the literature is examined to show that all of its associations can be interpreted using the proposed semantics (Section 5). A case study using the profile is presented in Section 6. The paper closes with a discussion (Sect. 7) and an outlook to future research (Sect. 8).

2 A Cognitive Semantics for Associations

Recall that the semantics of a language construct are defined by its mapping to elements of the semantic domain [6]. For purposes of conceptual modelling, this domain is the application domain, *as perceived by the modeller* (or a group of modellers). Hence, in order to define the semantics of the association construct, we need to first identify the *perceived* elements of the domain. While previous studies [21, 22, 23, 24, 25] have employed ontologies such as that by Bunge [26] or SUMO [27], that research has been criticized for the apparently arbitrary adoption of the ontology [28, 29, 30, 31]. In this paper we recognize that perception and interpretation of reality is strongly shaped by our mental, cognitive concepts and structures [32, 33, 34]. Hence, these concepts constitute the semantic domain to which the association construct must be mapped.

A suitable basis for identifying cognitive concepts are linguistic structures such as syntax or grammar. Cross-linguistic and developmental psychology research has shown strong evidence of correspondence between general cognition and linguistic structures. Cross-linguistic research has shown that universal linguistic features express human cognitive concepts by showing that variations in linguistic features co-occur with variations in cognition. Such co-variation has been observed in color categorization [35, 36], cultural categories of shame [37], counterfactual reasoning [38], spatial reasoning [39, 40, 41, 42], gender systems [43], and for objects and events [44]. Developmental psychology examines whether cognitive structures impact language acquisition, or whether language, as it is acquired, shapes cognitive structures. Studies have shown strong correlations between nouns and objects [45, 46], adjectives and properties [47, 46, 48], object labeling and category structures [49, 50, 51, 52, 53, 54], proper names and properties [55], persistence and sortals [56, 57, 58], quantification and object solidity [59], count nouns and shapes [60, 61, 62]. Hence, when identifying cognitive structures for associations, we can begin by examining the natural language used with associations.

The association construct links classes of objects (classifiers) to express assertions or propositions about elements of the application domain. Associations such as "person *works for* company", "stock exchange *lists* company", or "person *owns* document", make assertions about people and companies, stock exchanges and companies, and people and documents, respectively. Embley notes that "relationships associate one object with another, similar to the way verbs and verb phrases relate one noun or noun phrase to another" [63, p. 18]. Similarly, Chen proposes: "A transitive verb in English corresponds to a relationship type in an ER diagram" [64, p. 130]. The correspondence between verbs and associations is also supported by the analysis of NIAM in [65, 66]. A recent empirical study found that all examined relationships were named by verbs or verb phrases [67]. Hence, associations should be mapped to those cognitive concepts that are expressed by verbs.

To identify the cognitive concepts expressed by verbs, we examine the two most influential frameworks in the psychology of language, Jackendoff's conceptual semantics (ConS) [68, 69, 70, 71], and Talmy's cognitive semantics (CogS) [72, 73, 74, 75]. Both are based on a broad, cross-linguistic empirical basis, and claim language independence and universality across languages, improving on previous ad-hoc mappings [76].

The fundamental cognitive distinction in ConS and CogS is between spatial and temporal cognitive concepts. Verbs express temporal concepts which can be either States or Events. Figure 2 summarizes our chain of reasoning: Associations are described by verbs, which in turn represent States and Events. This assumes that classes represent things in the application domain that participate in events (see Section 3.2)[2][3][4]. In the terminology of Harel and Rumpe [6], we define the semantic domain S for conceptual modelling to consist of human cognitive concepts. We define the semantic mapping

$$M : L_{UML} \rightarrow S$$

from UML language constructs L_{UML} to the semantic domain S such that

$$M(association) = \{Event,\ State\}$$

To allow conceptual modellers to express these semantics for the association construct, we define a UML profile "Cognitive Semantics for Conceptual Modelling", to be applied to the UML 2 meta-model (Fig. 3). In this profile, we define an abstract stereotype «CognitiveAssociation» which extends the association construct (the metaclass "Association"). The stereotypes «EventAssociation»

[2] We are interested in explaining the semantics of associations, rather than the representation of events. While it may happen that events are modelled as classes such as 'Shipping', 'Enrollment', etc., this is outside the scope of this paper. For further discussion see [77].

[3] This corresponds well with the intuitive analysis by [76].

[4] Classes may be mapped to the cognitive concepts of things, places, and paths. However, the full development of a semantic mapping for the UML class construct is beyond the scope of this paper. A comprehensive ontology based on cognitive linguistics is found in [29], but without the depth and focus on events and states presented here.

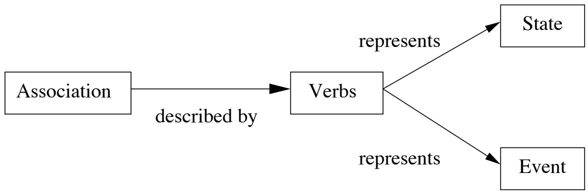


Fig. 2 Associations represent either states or events

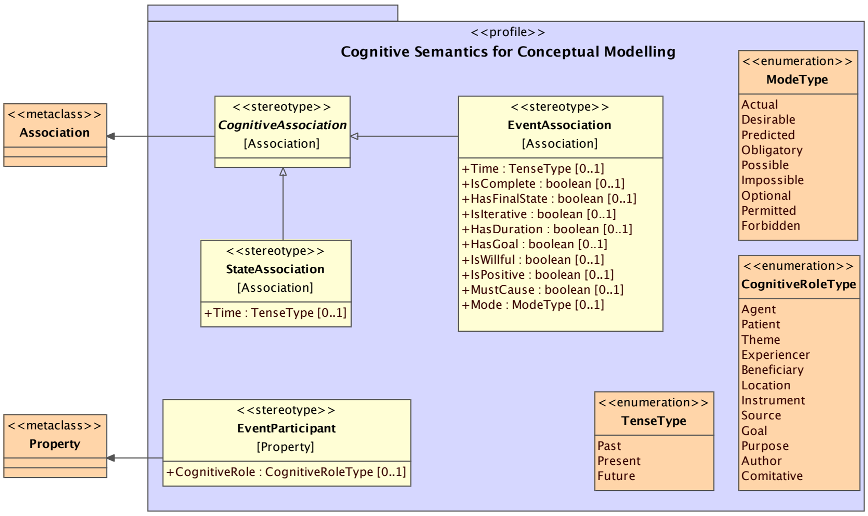


Fig. 3 UML 2.0 Profile for Cognitive Semantics

and «StateAssociation» are concrete sub-classes of the «CognitiveAssociation» stereotype. While some events are related to states, not all are. Therefore, such relationships are not modelled in Fig. 3 but are expressed as constraints on specific types of events. A further discussion of the profile in Fig. 3, including connections between states and events, follows in Sect. 3 below.

Harel and Rumpe suggest that "the description [of the semantic domain] can be in varying degrees of formality, from plain English to rigorous mathematics" [6, p. 67]. As no "rigorous mathematical" description of cognitive concepts exists, we turn to "plain English".

Events are defined as changes of objects, including instantaneous events or long processes, and subsume what may colloquially be called events, processes, actions, activities, etc., without making any distinction among these. Verbs for events express dynamic action or activity, as shown by the following examples:

– Customer has ordered product.
– Supplier will ship product.
– Student has enrolled in course.
– Product must be used in other product.

– Person should not supervise other person.

In contrast, a state expresses a static condition that holds between objects. Static conditions are those during which no change occurs and that are not associated with activity. While most object-oriented languages possess state constructs, these typically express conditions that hold *within* an object. In contrast, the state semantics of associations proposed here expresses conditions involving two or more objects. For example, in English they are commonly expressed by the verb "be" (or its more common form "is"). These include structural relationships, as shown by the following examples of States, none of which involve dynamic action or activity.

– Professor belongs to faculty.
– Part is contained in product.
– Warehouse consists of aisles.
– Office is on top of factory.
– Drill is next to lathe.
– Student is member of chess club.

The last example, while conceivably modelled by an ordinary association, should be modelled using specific constructs for aggregation or composition that UML provides. As indicated in Sect. 1, these constructs are outside the scope of this paper. In contrast, the following sentences, while superficially representing states, involve dynamics:

– New machine is reserved for factory.
– Captain is certified for aircraft.

"Reserving" and "certifying" are dynamic actions or activities. Hence, they can be expressed as events that have begun in the past and are ongoing (see also the discussion of progressive and telic events in Sect. 3):

– New machine has been reserved for factory.
– Captain has been certified for aircraft.

3 Implications of the Semantic Mapping

Because the mapping is made by means of verbs, this section identifies the cognitive concepts that natural languages mark on verbs. These are additional cognitive concepts that must be expressed by associations. Table 1 is a synthesis of cognitive concepts from cross-linguistic research. This research is based on cross-linguistic observations and the identified concepts are assumed universal. For example, two prominent works in linguistic semantics [81, 82] each examine more than 200 languages, from Aghem to Zapotec.

The concepts discussed in this section are represented as stereotype attributes (tags) in the proposed profile (Fig. 3). For easy implementation the concepts have been renamed from their original linguistic terminology. The minimum multiplicity for all stereotype attributes (tags) is zero to allow the modeller to indicate that

Tag/Attribute	Concept	Reference	Description
Time	Tense	CaseG, LS, LT, LFG	The relative temporal position of the activity
IsComplete	Aspect	CaseG, LS, LT, CogS, LFG	The state of completion of the activity
HasFinalState	Progressivity	CaseG, LT, LFG	Does the activity have a final state?
IsIterative	Iterativity	CaseG, LS, CogS	Is the activity repetitious or done once?
HasDuration	Punctuality	LS, CogS	Temporal interval of the activity
HasGoal	Telicity	LS, CogS	Does the activity have a goal?
Mode	Modality	CaseG, LS, LT	Permission, Ability, Obligation, Prediction, etc.
IsWillful	Volitionality	CaseG, CogS	Is the activity willful or accidental?
IsPositive	Opposition	CaseG, CogS	Is the affected thing positively or negatively affected?
MustCause	Success	CogS	Success criterion of the activity: To effect change or to prevent change

Table 1 Cognitive concepts related to events with references to source (CaseG = Case Grammar [78, 79], LS = Linguistic Semantics [78, 81, 82], LT = Linguistic Typology [84, 83], CogS = Cognitive Semantics [72, 73, 74, 75], LFG = Lexical Functional Grammar [80])

no distinction along a cognitive dimension is made. For example, if the `Time` attribute is omitted, every instance of the association may represent an event that happens in either past, present, or future, and each instance can represent an event at a different time. This prevents an unnecessary increase in the number of associations in a model.

3.1 Cognitive Concepts for States and Events

Tense Tense indicates when an event occurs or when the condition of a particular state holds. Most natural languages distinguish at least three tenses: past, present, and future. Some distinguish more. In English this distinction is marked on the verb itself (English uses the auxiliary 'will' for future tense). For example, we can distinguish between "products were delivered to customer" and "products are delivered to customer". Similarly, for states, we can distinguish between "customer is a VIP customer" and "customer was a VIP customer".

Tense is represented by the attribute `Time` for the «StateAssociation» and «EventAssociation» stereotypes. `Time` takes values from the `TenseType` enumeration defined in the profile.

IMPLICATION While not requiring the stringent time-stamping of temporal databases, making these distinctions in the conceptual model can describe important aspects of the application domain. For example, there may be more than one association between two object classes, tagged with different values for the `Time` attribute, and indicating events or states at different times. It allows the modeller to distinguish for example orders that will be delivered, from orders that have been delivered.

Aspect Aspect indicates whether an event is completed (perfective) or not (imperfective). In English this distinction is marked on the verb itself, e.g. "has been" vs. "had been" vs. "is". Aspect allows us to distinguish among events such as "supplier had been delivering product", and "supplier has been delivering product".

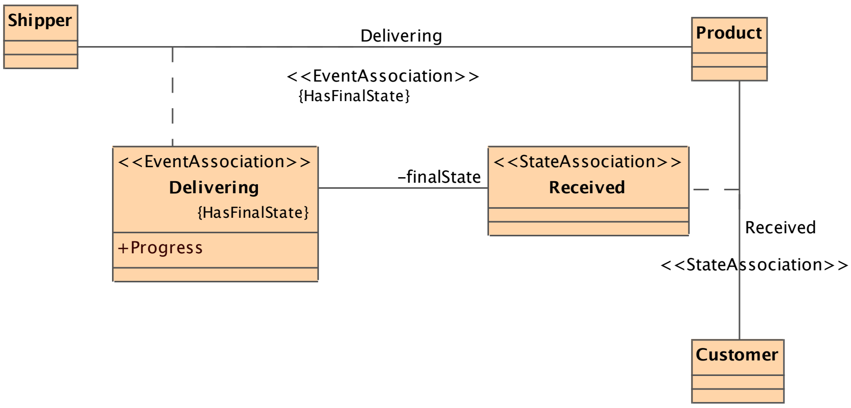
Aspect is represented by the boolean attribute `IsComplete` for the «EventAssociation» stereotype. It is not defined for «StateAssociations»; it makes little sense to speak of a state as ongoing or completed. The fact that a state condition held in the past is represented by tense.

IMPLICATION In the interests of a clear and unambiguous conceptual model, associations describing events with different aspects should be modelled as separate associations, tagged with different values for the `IsComplete` attribute. For example the association "supplier delivers goods" is tagged with `IsComplete = false` and the association "suppliers delivered goods" is tagged with `IsComplete = true`.

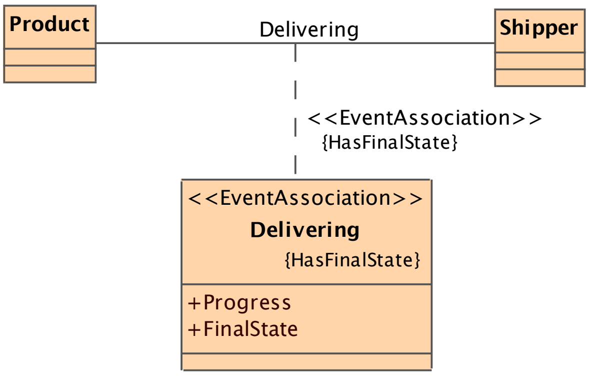
Progressivity This semantic distinction allows the differentiation between events that progress towards a final state (progressive), and those that do not (non-progressive). For example, a progressive event is "shipper delivers products", whose final state is reached when the products have arrived. On the other hand, the event "factory manufactures product type" is non-progressive, there is no final state. The factory will continue to manufacture a product type until the product type is discontinued.

Progressivity is represented by the boolean attribute `HasFinalState` for the «EventAssociation» stereotype. Progressivity is independent of tense (Time); progressive and non-progressive events may be in the past, present or future. Progressivity is independent of aspect (IsComplete); progressive and non-progressive events may be ongoing or completed. Progressivity is not applicable to states; it makes no sense of talking about the progress of a state or static condition.

IMPLICATION For progressive events, the model must include association class attributes that reflect the progress of the event and the final state of a progressive event must be represented. Two cases can be distinguished. (1) If the final state is within the scope of the conceptual model, it is itself represented as a «StateAssociation», as per the semantic mapping of associations. In this case, the «EventAssociation» is associated with the «StateAssociation» representing this



(a) Progressive event, final state in model scope



(b) Progressive event, final state not in model scope

Fig. 4 Example of progressive event

final state. For example (Fig. 4 (a)), the event "Delivering" (shipper delivers product) has a final state "Received" (products received by customer)[5]. (2) If the final state is not within the scope of the conceptual model, the modeller should represent it as an attribute of the «EventAssociation» (Fig. 4 (b)).

For a progressive event, the proposal merely requires that progress towards the final state be measurable, it does not require any particular form of measurement. For example, the "Progress" attribute on the event association "Delivering" in Fig. 4 could take on values such as "departed depot", "in transit", "final delivery", or "received".

These implications are expressed in the following OCL invariant:

[5] See also example 2 in Sect. 4 below.

```
context EventAssociation inv:
if self.HasFinalState then
self.baseAssociation.oclIsTypeOf(AssociationClass)
and
self.baseAssociation.ownedAttributes->exists(name='Progress')
and
( self.baseAssociation.ownedAttributes
          ->exists(name="FinalState')
  or
  self.baseAssociation.ownedAttributes->exists(a |
     a.association->exists(as |
     as.oclIsTypeOf(AssociationClass)
     and as->exists(extensionStateAssociation) ) )
```

Figure 4 shows an association between two association classes. It can be clearly identified as representing a state, namely the state of the world that a "Progressive events Has Final state". However, this is an assertion about the language itself, rather than about the domain. We do not stereotype associations representing such states.

Iterativity Iterativity indicates whether an event consists of a single action or is repeatedly performed. An example of an iterative event is that of "customer picks up orders on Wednesdays". In this case, the customer picks up the order *every* Wednesday. In contrast, a non-iterative event is the "customer picks up an order on Wednesday". This is an activity that happens only once, i.e. on a particular Wednesday. Iterativity is independent of tense (`Time`), aspect (`IsComplete`), and progressivity (`HasFinalState`). Iterativity is represented by the boolean attribute `IsIterative` for the «EventAssociation» stereotype.

IMPLICATION Iterative events have a frequency and a duration of the individual action. In the above example, e.g. weekly and 30 minutes. The modeller must include association class attributes expressing frequency and duration for iterative events (Fig. 5). The following OCL constraint expresses these implications.

```
context EventAssociation inv:
if self.IsIterative then
self.baseAssociation.oclIsTypeOf(AssociationClass)
and
self.baseAssociation.ownedAttributes
  ->exists(name='Frequency')
and
self.baseAssociation.ownedAttributes
  ->exists(name='IterationDuration')
```

Punctuality The concept of punctuality concerns the temporal distribution of an event. It is used to distinguish instantaneous from durative events, the latter having a non-zero duration. An example of an instantaneous event is the "product leaves assembly line", while an example of a durative event is the "product is being painted". Punctuality is independent of tense (Time) and progressivity (HasFinalState). However, iterative events (IsIterative) are always durative. While each

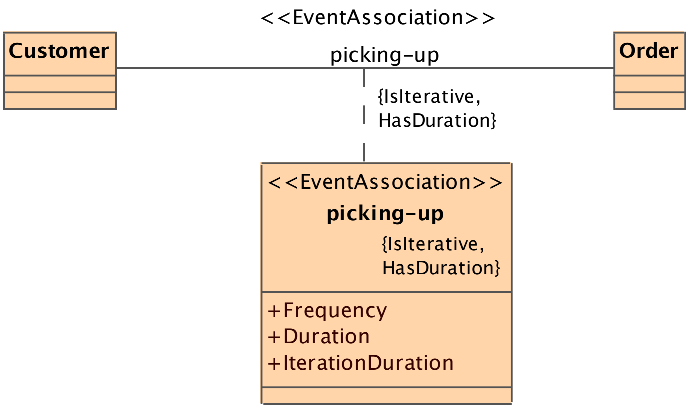


Fig. 5 Example of an iterative and durative event

individual occurrence may be instantaneous, the presence of iteration requires a temporal interval, i.e. a duration. Similarly, instantaneous events are always completed (aspect); it makes no sense to speak of an instantaneous event to be ongoing. Punctuality is represented by the boolean attribute `HasDuration` for the stereotype «EventAssociation».

IMPLICATION Durative events have by definition a duration. Hence, the model should include an association class attribute representing the duration. The example in Fig. 5 shows an event which is both iterative and durative, i.e. has frequency, duration of each iteration and overall duration of the event. Customers pick up orders for ten weeks (`duration`, the duration of the overall event that consists of many iteration instances) on a weekly schedule (`frequency`). Every pick-up takes one hour (`iteration duration`, the duration of each iteration instance of the overall event). The following OCL constraints express these implications:

```
context EventAssociation inv:
if self.HasDuration then
  self.baseAssociation.oclIsTypeOf(AssociationClass)
and
  self.baseAssociation.ownedAttributes
    ->exists(name='Duration')
context EventAssociation inv:
if not self.HasDuration then
  self.IsComplete=true and
  self.IsIterative=false
```

Telicity Telicity refers to the intentional goal-directedness of an event and concerns the intentions of the performer of an event. Telic events are always progressive (have final state), but not every progressive event is telic. For example, "inventory is shrinking" may occur through no directed action (e.g. spoilage), yet it possesses a final state (no inventory). On the other hand, the event "inventory is cleared out" is intended to achieve the final state of the activity (no inventory).

Telicity is independent of tense (Time), aspect (IsComplete), iterativity (IsIterative) and punctuality (HasDuration). Telicity is expressed by the boolean attribute `HasGoal` for the «EventAssociation» stereotype.

IMPLICATION Telic events have a goal which must be included in the model. As telic events are always progressive (have final state), the final state is assumed to be the intended goal state of the event. Hence, a separate goal state does not need to be modelled. The following OCL constraint expresses these implications.

```
context EventAssociation inv:
if self.HasGoal then
  self.HasFinalState
```

Modality Modality allows the modeller to describe whether an event does happen (actuality), should happen (desirability), may happen (optionality), will happen (prediction), must happen (obligation), can happen (possibility), cannot happen (impossibility), is allowed to happen (permitted), or is not allowed to happen (forbidden). Modality has been recognized as important for requirements specifications, and has been partially formalized in an IETF RFC [85]. Indicating modality for associations can clarify the intended semantics and distinguish whether, for example, "customer picks up orders" is something that actually happens (actuality), something that the customer should do (desirability), an option the customer may choose (optionality), an action the customer is known to do in the future (prediction), a constraint on the customer (obligation), an ability of the customer (ability), or something the company allows or forbids the customer to do (permission, prohibition). Modality is independent of all other cognitive concepts for events. Modality is represented by the `Mode` attribute for the stereotype «EventAssociation». `Mode` takes values from the `ModeType` enumeration defined within the profile. Explicitly marking modality on associations is important as it is not marked on English verbs.

IMPLICATION Modelling of modality may lead to multiple associations between classes. For example, in a human resource system there maybe multiple associations between the "Role" class and the "Person" class: "may fill", "must fill", "should fill", "can fill", etc.

Volitionality Volitionality indicates whether one of the participants in the event made the decision to execute the event; compare "the door opened for the truck" versus "the door was opened for the truck". In the first case, we understand that the door autonomously 'decided' to open, while in the second case the door was made to open. Note that both are telic, i.e have goals. Volitionality is expressed by the boolean attribute `IsWillful`, defined for the «EventAssociation» stereotype.

Volitionality is closely related to telicity (HasGoal) and progressivity (HasFinalState), as volitional (willful) events are always telic (have a goal), and therefore progressive (have a final state). However, the inverse does not hold. Volitionality implies that the agent with the goal participates in the event (i.e. the association), while this is not required for telicity. Volitionality is independent of tense (Time), aspect (IsComplete), iterativity (IsIterative), punctuality (HasDuration) and modality (Mode).

IMPLICATION In the case of volitional (willful) events, the modeller must indicate the agent that decides on the execution of the event. One of the participant classes in the association must be tagged with $CognitiveRole = Agent$ (see the discussion of event participants in Sect. 3.2 below).

```
context EventAssociation inv:
if self.IsWillful then
self.baseAssociation.memberEnd.class
  ->exists(c |
      c.extensionCognitiveClass and
      c.extensionCognitiveClass.CognitiveRole='Agent')
 and
self.IsTelic
```

Opposition Opposition specifies whether an action has a positive or negative effect on the affected object, and whether the affected object would be opposed to the event. For example, "the customer defrauds the business" has a negative effect. Clearly, the business is the affected object, the customer is the agent of the event. In contrast, "the customer refunds the money to the business" has a positive effect on the affected object. Opposition is independent of all other cognitive concepts defined for EventAssociations. Opposition is expressed by the boolean attribute `IsPositive` defined for the «EventAssociation» stereotype.

IMPLICATION Opposition has no modelling implications other than that it can be recorded when this information is helpful for the purposes of the conceptual model, and to disambiguate potentially confusing modelling situations.

Success Some languages distinguish grammatically whether an event should bring about or prevent a certain outcome. The concept of success, specifically the criterion for success, allows the modeller to make this distinction. For example, in one application the success criterion may be the prevention of the event "staff enters area", which is successful if the staff does not enter the area. In another application, the success criterion may be to bring about the event "staff enters area", which is successful if the staff does enter the area. Success is related to modality; for events that are forbidden, the success criterion is prevention. For events that are desirable, the success criterion is causation of the event. Success is represented by the boolean attribute `MustCause` for the «EventAssociation» stereotype. When the attribute value is true, successful outcome is the completion of the event. When the attribute value is false, successful outcome is the prevention of the event.

IMPLICATION Analogous to opposition, success has no modelling implications other than that it can be recorded when this information is useful for the purposes of the conceptual model, and to disambiguate potentially confusing modelling situations. Success is related to modality, as expressed by the following constraint:

```
context EventAssociation inv:
if self.Mode='Forbidden' then
  self.MustCause=false
if self.Mode='Desirable' then
  self.MustCause=true
```

Role	Reference	Description
Agent	CogS, CaseG, LS, ConS, [78, 82]	An active agent or thing
Patient	CogS, LS, ConS, [78, 82]	An agent or thing that something is done
Theme	CaseG, LS, ConS, [82]	The topic of the event
Experiencer	CogS, CaseG, LS, ConS, [78, 82]	An agent or thing that experiences an activity
Beneficiary	CogS, CaseG, LS, ConS, [78, 82]	An agent or thing that benefits or receives
Location	CogS, CaseG, LS, ConS, [78, 82]	A location
Instrument	CogS, LS, ConS, [78, 82]	The instrument by which the action is performed
Source	LS, ConS, [82]	A source location, thing or agent
Goal	CaseG, LS ConS, [82]	The goal of the action
Purpose	LS, ConS, [82]	The purpose of the action
Author	CogS, LS	The speaker or writer of a communicative action
Comitative	[82]	An agent or thing accompanying an action

Table 2 Cognitive concepts for event participants with references to source (CaseG = Case Grammar [78, 79], LS = Linguistic Semantics [78, 81, 82], CogS = Cognitive Semantics [72, 73, 74, 75], ConS = Conceptual Semantics [68, 86, 69, 70, 71])

3.2 Event Participants

Events are expressed by verbs, which in turn possess one or more arguments [78, 81, 82]. Just as verbal arguments play thematic roles, so the instances of classes participating in associations play thematic roles. Table 2 shows a synthesis of the cognitive concepts proposed in the linguistics literature. In the UML profile in Fig. 3 we have defined a stereotype «EventParticipant» which extends the metaclass `Property`. The stereotype «EventParticipant» defines a single optional attribute, which can take on a value of the `CognitiveRoletype` enumeration corresponding to the cognitive concepts in Table 2. The stereotype is not applicable to all properties, but only to those that are member ends of an association that represents an event:

```
context EventParticipant inv:
self.base.association.extension
  ->exists(oclIsTypeOF(EventAssociation))
```

IMPLICATIONS The experiencer, beneficiary, and locative roles are mutually exclusive [79]. Hence, we propose the following OCL constraints:

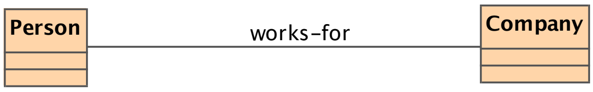
```
context Association inv:
 if self.memberEnd->exists(CognitiveRole='Experiencer')
 then not self.memberEnd->exists(CognitiveRole='Beneficiary')
 and not self.memberEnd->exists(CognitiveRole='Location')

context Association inv:
 if self.memberEnd->exists(CognitiveRole='Beneficiary')
 then not self.memberEnd->exists(CognitiveRole='Experiencer')
 and not self.memberEnd->exists(CognitiveRole='Location')

context Association inv:
 if self.memberEnd->exists(CognitiveRole='Location')
 then not self.memberEnd->exists(CognitiveRole='Beneficiary')
 and not self.memberEnd->exists(CognitiveRole='Experiencer')
```

4 Model Disambiguation Using The Natural Language Semantics Profile

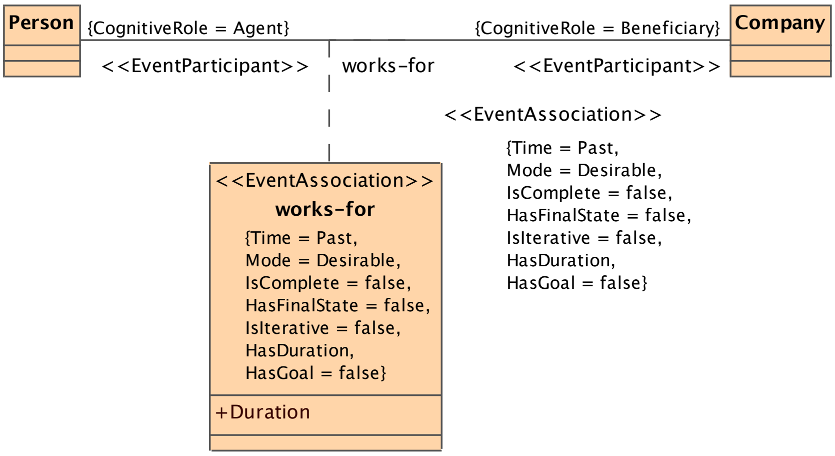
This section shows an example of how the proposed UML profile can clarify the meaning of ambiguous models. The example in Fig. 6 (a) is taken from [8], where not much is said about the application domain. The only semantic notion that is indicated is tense on the English verb, indicating that the event occurs or is occurring in the present.

Domain knowledge indicates, but does not completely determine, whether the association is intended to represent dynamic events, i.e. actual work being performed, or mere states of employment, i.e. contracts having been signed. Furthermore, we know nothing about whether the work being done is progressive, i.e. whether it terminates once a goal is reached (e.g. in the form of a fixed term or fixed outcome contract), or whether it is non-progressive, i.e. it continues indefinitely (e.g. in the form of a permanent, or habitual, employment). The work may be iterative (the person may be performing a single action every day), or it may be occurring once only (e.g. performing a specific task once, e.g. in the case of contract or project work). From domain knowledge we can assume that the work is durative, although punctual work cannot be ruled out (e.g. in the case of a simple action such as a courier delivering a document, which may be considered instantaneous). If the events are non-progressive, they cannot be telic, i.e. directed towards a specific goal. We can speculate on the modality, but without knowing the context of this model, we do not know whether the association describes job applicants (predicted), existing work relationships (actual), planned situations for a future project (desirable) or some other modality. Both the person and the company must play cognitive roles. As it is the person that carries out the activity, it plays the role of agent while the company plays the role of beneficiary, benefiting from the activity.

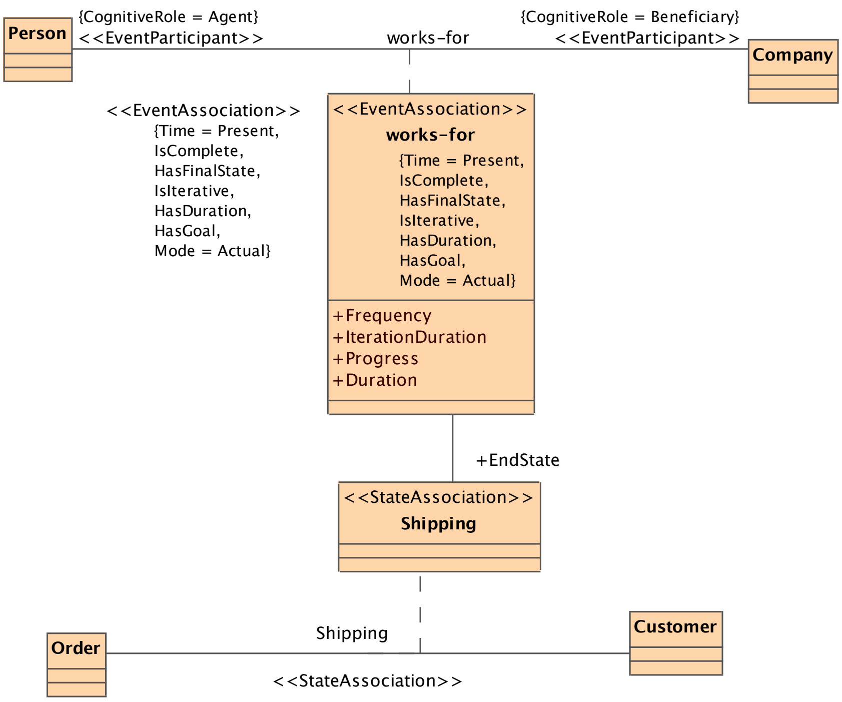
Using the proposed profile, we can model this domain with explicit semantics (Fig. 6 (b)). The new model clarifies the assumptions that we made. For example, it shows that the association represents dynamic activities («EventAssociation»). They occur in the present (`Time`) and are not completed (`IsComplete`).



(a) No semantic information



(b) Using the proposed profile



(c) Alternative interpretation

Fig. 6 Example association representing an event

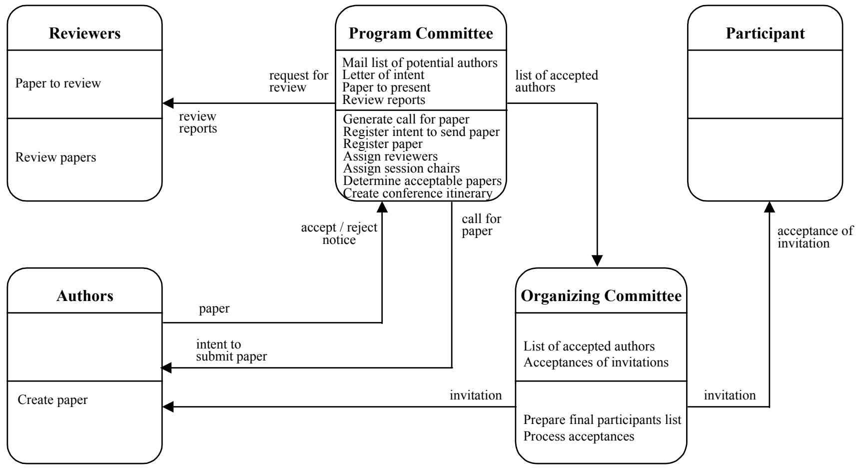


Fig. 7 Organizing a conference, from [87]

They are activities that do not end in a specific final state nor a specified goal (`HasFinalState`, `HasGoal`) and occur throughout an extended time interval (`HasDuration`). However, the activities are not yet actually occurring, they are presently desired (`Mode`).

A second interpretation is shown in Fig. 6 (c). Here, we assume that the events have a final state, a goal, and occur repeatedly, iteratively. Specifically, the person works for the company with the goal and final state to have orders shipped to the customer. Frequency and duration attributes are modelled to satisfy the constraints for iterative events. Clearly, the two interpretations are different. However, the original model only incompletely determined the interpretation and left the model ambiguous and the interpreter to make assumptions.

5 Example

This section applies the proposed cognitive semantics to an existing object-oriented model from prior conceptual modelling research [87]. Fig. 7 shows the domain of conference organization. This section is intended to show that the proposed semantic notions can be easily applied to existing models and, conversely, existing associations can be categorized using the proposed semantic notions. The associations in Fig. 7 are as follows:

Reviewers — Program Committee Without being aware of the intent of the model author, the following interpretation appears reasonable. As indicated by the association end names, this should be modelled as multiple event associations: The program committee (agent) invited reviewers (patient) (past, complete, has a final state, is not iterative, has a duration, has a goal, actual, and is willful). The final state is a state association between reviewer and program committee. A second event association between reviewers (agent or author), referee reports (patient) and the program committee (beneficiary) is needed (Reviewers will prepare reports for program committee, future, incomplete, has final state, not iterative, has a duration, has a goal, is willful, predicted or desirable). If the model in Fig. 7 was showing submitted or accepted papers as a class, these would be included as comitatives (reviews *about* papers).

Authors — Program Committee This association might conceivably relate to rejecting or accepting the paper submitted by the authors. Thus an event association between authors (agent), paper (patient) and program committee (location or beneficiary) is required to model submission (past, complete, final state, not iterative, has no duration, has goal, actual). The notifying event association might be modelled between program committee (agent), authors (location or beneficiary) and referee report (patient) (past, complete, final state, not iterative, has no duration, has no goal, actual). However, other interpretations, especially with respect to time and mode are also plausible. Other event associations include the authors (agents) registering their intent to submit a paper (patient) with the program committee (location or beneficiary), the program committee (agent) sending (or having sent) a call for papers (patient) to potential authors (location or beneficiary). However, without the original intention of the model developer, multiple interpretations remain open.

Program Committee — Organizing Committee This event association appears to indicate that the program committee (agent) will/has/is giving to the organizing committee (beneficiary or location) a list of accepted authors (patient). Again, without knowing the original intention of the modeller, multiple interpretations are possible.

Organizing Committee — Authors This event association indicates that organizing committees (agents) invite (or will/should/must invite?) authors (patients).

Organizing Committee — Participant This event association appears similar to the one between organizing committee and participant.

6 Case Study

This exploratory case study investigates the application of the proposed cognitive semantics. A case study can demonstrate the applicability and feasibility of the technique in a real-world project setting and can point towards areas where the technique is most helpful or where modifications might be needed. The case study's primary goal was to reflect on the process and the feasibility of applying the proposed profile. The case study should be read as a reflective experience report and sufficient evidence has been provided to allow the reader to judge the plausibility of the conclusions drawn from this case study.

6.1 Case Study Introduction

Industry partners for this study are a port operator ("PO") and a related warehousing company ("WH"). WH offers container loading, goods storage, and related services and works in close co-operation with PO. WH receives goods for shipment from its customers. The goods are delivered by shipping companies. Pallets of goods are unloaded and warehoused by WH, then loaded into 20ft or 40ft containers. These containers are then sealed and handed off to PO for loading onto cargo vessels.

PO provides all information technology services to WH. As part of an information systems project to improve warehouse management at WH, PO has developed an extension to their existing port management system that allows WH staff to manage inventory, movement of goods, container load planning and container packing. This project was chosen as the basis for the case study for three reasons. First, it was a small and well-scoped project in a domain that was well-understood by all stakeholders. Second, PO employs object-oriented system design methods, yielding a good fit with the technique proposed in this paper. Third, all project stakeholders were readily accessible. This study was conducted as an independent system analysis, carried out with similar scope as the one originally conducted by PO/WH for their new system.

6.2 Data Collection

Information about WH's operations were collected using interviews and observation. The initial interview, conducted at PO's offices, included the programmer and project manager at PO, and the owner of WH. The purpose of this interview was to find out how WH operates in terms of their business activities, such as their business processes, the entities involved in these processes, and the information that WH wishes to record and manage with the new information system. The initial interview was followed by observation of the operational activities of WH. For this, a research assistant spent a day at the WH warehouse and observed daily activities. To complement the observational and interview data collection, email clarifications were sought from project participants when required. A final meeting with the project stakeholders presented initial models to the WH/PO staff to ensure correctness, completeness and gather initial feedback on the perceived quality of the presented models. Based on that feedback, the models underwent another iteration of modifications. The models were not further validated, as the case study's primary goal was not to create necessarily correct models, but to investigate the feasibility of the proposed profile. Model correctness is important, but here it is a secondary goal.

6.3 Model Development

Based on the information gathered from the initial interview, preliminary UML models using the proposed UML profile were developed and iteratively refined.

As the focus of this proposal is on the association construct, the models contain only elements found in class diagrams. The major classes in the business domain were immediately obvious, as they represent physical things:

- Unit (of goods/products)
- Pallet
- Pallet load (of products)
- Truck load (of products)
- Returned units
- Carrier (trucking or shipping company)
- Container
- Ship
- Stack (place in warehouse)
- Stock (products)
- WH warehouse
- Operating personnel at WH

The criteria for identifying candidate events are those described in Section 2: "Events are defined as changes of objects, including instantaneous events or long processes, and subsume what may colloquially be called events, processes, actions, activities, etc." Any action/activity or event that was observed or communicated by stakeholders was a candidate event. It was also helpful to express these in natural language English sentences (see Sect. 6.3 below). The main activities that make up WH's daily operations were clear from the initial interviews. The model and UML diagrams were structured around these main activities by developing a separate diagram for each of these activities:

- Receiving of truck loads of products
- Storing loaded pallets
- Returning empty pallets
- Returning products
- Monitoring of warehouse operations
- Stock taking
- Placing containers on ship

In general, all roles of an association for which a participant could be identified were included in the model. For reasons of space we do not show all the diagrams or describe the process of modelling, but summarize the findings of the case study. Fig. 8 provides an example diagram from the case study that will be used for illustrating points in the following discussion.

6.4 Case Study Findings

During model construction using the proposed UML profile some issues arose that reflect on the feasibility of applying the proposed profile.

The first issue was confusion between (a) associations representing states and (b) the UML provided notions of generalization and specialization of classes. For

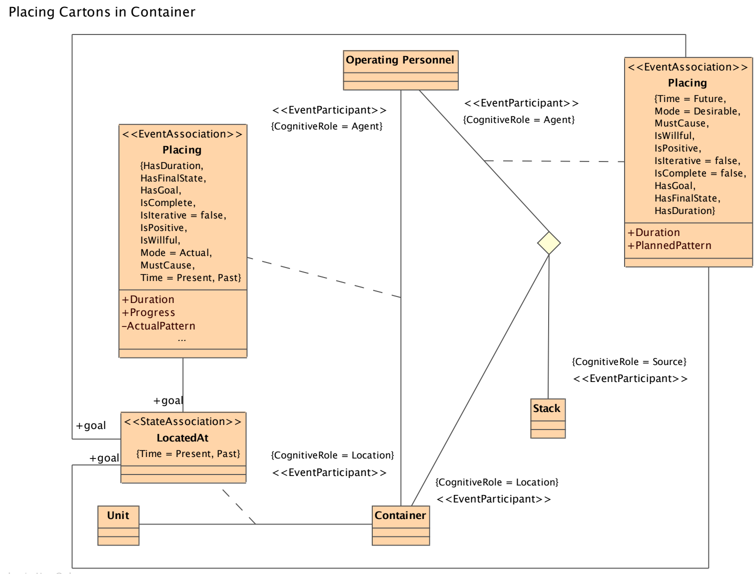


Fig. 8 Example diagram from case study

example, the fact that "a full pallet is an empty pallet with product" can be described either by an association that is stereotyped as a «StateAssociation», or it can be described using the generalization relationship in UML. This issue occurred in different contexts, such as "a full container is an empty container plus products". It was resolved by deciding to use the UML generalization construct: Using the generalization construct allows more specific and precise description of the application domain. For example, the UML generalization construct has the added semantics of inheritance, which is not possessed by all states. In fact, these added semantics were a second reason for keeping with the UML generalization concept, as this allows UML modelling tools to exploit these semantics for model analysis and code generation during later stages of the system development process.

A second issue requiring resolution was the nature of the events to be modelled. Initially, most of the events were analysed as being transitive and modelled using binary associations. For example, the event "Placing" was modelled as a binary «EventAssociation» between containers and units. On further reflection, the source of the units, the stack (location in warehouse) was included for a ternary «EventAssociation» (Fig. 8).

Debate over inclusion of operating personnel as participants in the associations required clarification from PO/WH. In the example shown in Fig. 8, the operating personnel is the agent of the event and should therefore participate in the association. However, the purpose of any IS analysis model is to lead to an implemented system and the conceptual model should not include unnecessary elements. Clarification was thus sought whether WH would wish to track individual activities of its personnel. Discussions revealed that this was not currently done. However, the system in development will have this capability and this capability is therefore desirable for WH to have. Hence, operating personnel should be included as association participant.

However, this decision has implications that go beyond technology, e.g. cost and ethics. For example, is it worthwhile planning and maintaining a schedule for such detailed activities as placing individual cartons? Is it acceptable for personnel to be supervised and controlled to this level of detail? While these issues cannot be easily resolved, the modelling decision was made to follow the lead of PO in their development of the new system and include the operating personnel as participants in the association, shown in Fig. 8.

A third issue arose out of the need to both plan events, as well as capture actually occurring events. Throughout the case study, this has frequently led to multiple associations with different tense and mode. For example, in Fig. 8 two associations are modelled for the activity of placing cartons in containers. The one on the left represents the actual activities being carried out, as shown by the `Mode` and `Time` tags. The association on the right represents future planned activities, as indicated by the `Desirable` value for the `Mode` tag and the `Future` value for the `Time` tag. Note that these associations have different sets of participants. For the planning of the container loading, it is important to identify and keep track of the location (stack) where the product is to be taken from. For the tracking of the actual activities, it is merely required to know who (operating personnel) is packing what container. The latter is a requirement identified by WH to satisfy traceability and accountability of staff, while the former is a requirement stemming from WH's inventory management and inventory planning.

While multiple associations lead to an apparent greater model complexity, they also allow a clear separation of e.g. plans versus actual events. Two recent studies [88, 89] show that the negative effects of apparent greater model complexity may be outweighed by the benefits of additional information or clearer representation. In these studies, increases in model complexity stemming from increased fidelity of domain representation did not lead to any disadvantages in terms of model readability.

A fourth issue that was identified during modelling is the requirement to include goals and/or final states for telic and/or progressive events. This requirement appeared sensible based on the linguistic considerations during the develoment of the technique. However, when applying the technique to the case, most of the goals or final states appear trivial, as exemplified in Fig. 8. For example, the goal and final state of the activity "Placing" is that state in which units of product are located in containers. This may be a consequence of this case study being particularly simple. More applications of the technique are required to evaluate whether outcome states are *always* trivially related to events, in which case they may well be omitted.

Fifth, another issue involved the use of multiple time tag values for associations. As part of the requirements gathering at WH, it became clear that information about past events would be maintained indefinitely, and that information about future events, planned or expected, would also be maintained when possible. Thus, as time progresses, this necessitates multiple associations with different tag values for each type of event. A similar issue arises as events also change from being incomplete to being complete as time passes, requiring separate associations with different values for the `IsComplete` tag. Using the proposed technique makes it clear that distinctions exist and is thus more faithful in representing the domain. For the software system, the use of multiple associations requires a mechanism to "move" instances of the association representing incomplete, present events to the association representing completed, past events. For example, this could be done by requiring operating personnel to log the completion of individual activities with the software system. While multiple associations do not necessarily impair the readability of the model [88, 89], the fact that this is a common situation led us to allow each association to carry multiple values for the `Time` tag. An example of such an association is also shown in Fig. 8, where the "Placing" «EventAssociation» on the left contains two `Time` tag values. Note that the "Placing" «EventAssociation» on the right, representing future events, contains not only a different `Time` tag value, but also differs in other ways from the association representing past and current events (different mode, different participants).

Sixth, an issue that came up repeatedly was the naming of the associations. The modeller (research assistant) originally named event associations using a specific tense and modality in natural English language, e.g. "operating personnel places product in container". However, these labels were often found to be inadequate or misleading. On further discussion of the requirements and the process with WH/PO staff, a different tense and modality emerged as more appropriate, e.g. "operating personnel should place product in container" or "operating personnel has placed or is placing product in container". To prevent these problems, the associations are labelled using the infinitive forms of verbs and all other grammatical markers modelled as tag values on the associations. This was found to be helpful and important to avoid conflicting interpretations of the diagram.

Related to this, it was found to be helpful to represent the associations and the semantic markers on them in natural English language (as far as this is possible: English does not mark all the possible semantic distinctions). For example, the English *approximations* of the two «EventAssociations» shown in Fig. 8 are:

- Operating personnel should be placing something from stack into container (until units of product are located in container).
- Operating personnel have been or are placing something in container (until units of product are located in container)

Expressing the associations using natural language enabled a checking of the modelled markers against an easier to understand description. It also served to reveal inadequate characterization of English language requirements. For example, a requirement might have been expressed as "Operation personnel perform stock takes". On closer examination, after employing the available semantic markers, this was found to be inadequate. Instead, the correct and precise requirements statement should have read, "Operating personnel should be performing stock takes", indicating the desirability, the future tense, and the durative nature of the events. The fact that only an approximate English expression could be found shows that English is often a limited language and may lead to incompletely understood requirements.

6.5 Case Study Conclusions

The use of the additional tags and stereotypes on elements led to elication of requirements beyond what would have been done for a model without this information. For example, the fact that an event has to be characterized as either telic or non-telic, and that for telic events a goal state has to be identified, requires additional information to construct a complete model. Another instance where additional information is required is the iterativity and durativity of events. For example, does a supplier deliver products regularly with a certain frequency and does the "delivering" comprise many such individual events, or is "delivering" an individual, non-repeated event?

The new modelling technique forced both the analyst (researcher) as well as other stakeholders to clarify and elaborate the business domain and the IS requirements. This yielded models that captured the application domain in a more precise way. It also highlighted questions about the desirability about certain distinctions. For example, modelling past and present events implies that past data is not archived out of the system, and that a software mechanism exists to change event instances from present to past as time progresses.

The use of the stereotypes and tags was found to be able to guide the modelling process. For example, when a progressive event was modelled, its final state must be identified. Similarly, when a state was modelled, it is important to identify the event that brings about this state and determine whether it is within the scope of the model.

On the other hand, the application of the profile also showed that these benefits can be realized only through a greater modelling effort. The number of associations in the model will increase, and so will the amount of information captured on each association. Some difficulties were encountered when deciding on how to model particular facts, e.g. deciding on modality and iterativity. More precise guidelines will need to be developed for this in future research.

7 Discussion

This paper has defined a semantic mapping for the UML association construct *when used for describing application domains*, rather than describing software artifacts. The focus is on improving the understandability and reducing the ambiguity of the conceptual model of the application domain.

The proposed modelling technique has generally led to an increase in the number of model elements. However, while this increase may appear to impact the readability, recent studies have shown that this is not in fact so, and that increases in apparent model complexity can be accompanied by concommitant increases in representational fidelity and domain understanding by model readers [88, 89]. The case study results support this suggestion, as the increase in model elements was directly related to additional information being modelled, for example distinctions between current and past events. However, more rigorous experimental studies on the proposed technique are needed to confirm this.

While we have specified certain dependencies among values of stereotype attributes, other dependencies may exist. However, we adopt a principle of caution of specifying these as constraints for the following reasons. (1) Such dependencies are not postulated by our reference literature. (2) Specifying such constraints would imply their universality. However, there may exist exceptions, and it is impossible to consider all possible modelling situations. (3) Even if no constraints are specified, the modeller remains free to express the specific set of concepts required in any situation, or to extend the proposed profile for specific application domains by adding domain-specific constraints.

An aspect is the "naturalness" of the specification. Other things being equal, a specification that appears more natural is to be preferred. However, this proposal is aimed at improving the specificity and comprehensiveness of a domain specification, so these other things are not equal. Many of the notions presented here are not typically part of a domain description, nor do they enter into the considerations for model construction (see the discussion in Sections 4 and 5 where we present examples with multiple interpretations, as the original specifications are ambiguous).

One reason for any perceived "unnaturalness" may be the fact that the English language does not grammaticize all the cognitive concepts presented here. For example, while tense ("was", "is", "will be", etc.), mode ("should", "might", "ought"), iterativity ("goes" versus "is going"), and beneficiary/locative ("to") are grammaticized in English, other cognitive concepts are not. However, the other concepts are also important, as other languages grammaticize them and native speakers of such other languages might think of some English notions as "unnatural". For example, Russian marks telicity on the verb and a Russian speaker may well feel that an English conceptual model without telic markings misses important domain aspects. Conversely, tense is marked on the verb in English but not in Chinese where it must be expressed using additional phrases; a Chinese speaker may well feel that tagging associations for tense is less important. Instead of dismissing concepts that are not marked in ones own native language as unimportant, they can offer insights into the application domain and its dynamics, which may otherwise remain hidden. The research on which this paper is based suggests that, while not all natural languages grammatically encode all distinctions, every distinction is grammatically encoded in some language, indicating its universal importance.

As the profile is defined using standard extension mechanisms, tool support is provided by all UML tools that support profiles[6]. While tool support for OCL constraints is not yet widespread, some OCL tools exist to enforce the constraints

[6] The profile is defined using the MagicDraw UML tool. All diagrams in this paper are created using this tool.

in a model[7]. However, these OCL tools are not integrated in major UML modelling tools.

7.1 Relation to UML Dynamics

UML provides a rich palette of behavioral constructs and the present proposal should not be read as to exclude or prohibit their use. This paper maps assocations to states and events, it does not make any claims about the inverse mapping. The semantics assigned to associations in this proposal, events and states, are closely related to that of other UML language constructs and it should be expected that associations will be used with these other constructs in the same model. UML provides state chart diagrams that describe behaviour of a classifier by means of states and transitions between states. In UML, a state machine is a type of behaviour and can thus be associated with a behavioured classifier, typically a class, but possibly a collaboration of objects, a use case, or an actor [15]. The state associations proposed here involve two or more objects so that they might correspond to states of state machines that specify collaboration behaviour. It is conceivable that a modeller might link state machine states and state associations so that whenever a state association is modelled in a class diagram, the collaboration of instances of those classes should be specified by a state machine that contains a corresponding state. As progressive events have a final state, the incoming state transitions in a state machine might well correspond to the event associations in the class diagram.

UML also provides the action and activity concept. Both are used to specify behaviour of behavioured classifiers. As such, they are related to the event associations presented here. Again, as associations represent events involving two or more objects, actions and activities used in specifying the behaviour of collaborations of objects bear the closest relationship. UML further provides actions, signals, events, and receptions to specify interobject communication. A discussion of communication mechanisms is however beyond the scope of this discussion.

A related proposal, based on the high-level ontology of Bunge[26], also makes a connection between associations and the representation of domain dynamics in UML [24, 22, 90]. There, association classes represent sets of joint properties of objects that arise out of a single interaction. Associations in that proposal are equivalent to event associations here, if the event is ongoing. For events that are completed, the final state association in the present proposal may represent "properties" that arises out of an event. For example, an "Enrolling" event (association) between a student and a university may lead to a "Registered" state with properties of StudentNumber, TutionFees, etc. These properties are joint between students and university and arise out of the enrollment event. However, while similarities exist, the present proposal interprets associations as either states or events, rather than properties, and does not limit their relationships to other associations

[7] The Dresden OCL toolkit at `http://dresden-ocl.sourceforge.net`, the Octopus toolkit at `http://www.klasse.nl`, the OCLE toolkit at `http://lci.cs.ubbcluj.ro/ocle/`, and the OSLO project at `http://oslo-project.berlios.de/`.

or classes, as in [90]. There, association classes cannot be associated with other association classes. In contrast, it is clear that e.g. events can cause other events, and therefore relationships between association classes can exist and should be modelled. Thus, while the ontological proposal and the present profile share some concepts, such as events and states, there are significant differences.

Finally, another proposal that relates UML class diagrams to domain dynamics suggests specific ways of modelling events as classes, but does not offer the rich and deep event characteristics based on cognition that are presented here [77]. Instead, that research seems to suggest that when modelling events as classes, there is no little need to also use UML behavioral constructs.

In summary, while these are suggestions about relationships to other UML constructs, an in-depth analysis is beyond the objective of the paper, which is to examine the semantics of the association construct. Domain dynamics are modelled with considerably less frequency than class diagrams, so that class models are often the only models [11]. This means that, unless made explicit as in this proposal, domain dynamics are often not expressed at all, potentially leading to a lack of domain understanding. While this section has given some indications of possible relationships, more research, specific to this area, is required to establish consistent and useful modelling rules and constraints, especially given the rich and diverse set of UML constructs for describing dynamics.

7.2 Related Work

This paper has focussed on what UML calls ordinary associations. These exclude meronymic ("part-of") associations (indicated in UML by specifying `association.memberEnd.aggregation = "composite|aggregate"` for the member ends of an association). Meronymic relationships are discussed in [95], based on cognitive research [96, 97]. In the context of UML, aggregation and composition are discussed in [16, 17, 18, 19, 20].

In the literature, the semantics of the association construct are often implicitly assumed when associations are discussed as representing either (1) elements of the application domain [95, 98, 21, 22, 24], (2) elements of relational calculus and set theory [99, 100, 10, 101], or (3) elements of programming languages [7, 8, 102, 103, 104, 9]. These different implicit semantics correspond to conceptual modelling, database systems, and software engineering, respectively. In this paper, we focus on the interpretation of associations with respect to the conceptual modelling.

Many studies discuss pragmatic or implementation aspects of associations, without clarifying their semantics. For example, Kristensen [105] suggests a need for complex associations to support abstraction, and proposes a language construct and modelling method for this. Other research examines associations between more than two classes and their reducibility to binary associations, which depends on the multiplicities of the association ends [10, 98, 99, 100], participation constraints [10, 98, 99, 100], the presence of objects that represent the relationship [98] and constraints on insertion and deletion of instances [99, 100]. Stevens [9, 104] examines binary associations and their mapping to object-oriented programming languages, clarifying the semantics with respect to the software domain, but not with respect to the application domain.

Conceptual models describe the application domain. Hence, work in this area has frequently used ontologies, shared conceptualizations of a domain [106] to analyze the semantics of associations in conceptual modelling [21, 22, 23, 24, 25]. However, one concern about the ontologies being used is their lack of explicit foundation in empirical observation [28, 29, 30, 31, 94]. In contrast, the present paper is based on semantic frameworks that are explicitly rooted in empirical work. For example, Talmy's cognitive semantics work [74, 75] cites hundreds of empirical studies of dozens of natural languages, Whaley and Frawley's work on linguistic semantics is each based on empirical studies of more than 200 languages [82, 81] and Croft bases his linguistic universals also on hundreds of original studies of more than 200 languages [84, 83]. By comparison, the ontological work of Bunge has had comparatively little and indirect verification [91, 92, 93, 89] with some results not being very positive [94]. Moreover, the present proposal captures greater detail about associations and association participants than the ontologically-based work in [21, 22] and has concrete modelling implications not found in [23, 25]. For example, the present proposal provides a number of tags with which to characterize types of associatons and presents rules, in the form of constraints, on how to model associations.

A number of previous studies have recognized the importance of natural language grammar in IS or database development. Abbott [107] maps common nouns to abstract datatypes, and verbs, attributes, predicates or descriptive expressions to operators. Sykes [65] applies the concepts of subject, verb, object, complement and adverbial, borrowed from English grammar, to the NIAM development method. Also in connection with NIAM, Dunn and Orlowska [66] propose a method to parse English sentences into NIAM structures. English syntax is also the basis for a proposal by Chen [64], mapping nouns, verbs, adjectives, adverbs, gerunds, and clauses to elements of the ER modelling notation. Weigand [108] examines the use of functional grammar for knowledge representation, but does not explore in depth the semantics of specific grammatical categories. The approach by Rolland and Proix [76] is argued to be based on Fillmore's case grammar, although the actual concepts used by the authors bear little resemblance to Fillmore's theory, and are introduced without justification. A recent study on generating natural language from class diagrams found that all examined associations are labelled using verbs or verb phrases, while classes are exclusively labelled by nouns or noun phrases. Generating natural language from conceptual models can benefit from the present proposal, as stereotyping and tagging of associations determines the specific form of the verbs to generate.

The previous studies employing linguistics have three drawbacks. First, they employ syntactic or grammatical concepts, rather than semantic concepts. For example, a noun is a syntactic concept. Nouns are defined by the role they play in the construction of clauses, which are also syntactic concepts. Subject and object are grammatical concepts used to construct sentences, which are also grammatical concepts. Syntax and grammar are closely related: subject and object roles in a sentence are filled by nouns or noun phrases. Semantic concepts such as agent or patient are distinct, but again closely related. For example, the agent of an action is often expressed by the subject role in a sentence, which in turn is filled by a noun in the corresponding clause. Second, these studies are based on a single language, typically English, and neglect the majority of the more than 6000 spoken natural languages. Thus, we can be assured neither of their comprehensiveness (they may neglect grammatical or semantic notions that other languages express) nor of their universality (they may express notions idiosyncratic to the English language). Third, these studies offer no theoretical or empirical justification why syntax and grammar are suitable foundations for the semantics of such languages.

Notable exceptions are the discussions by Storey [109] and Siau [110], who independently develop classification of relationships based on cognitive principles of relation element theory. While a classification is a helpful step in elucidating the meaning of associations or relationships, it sidesteps the issue of defining their meaning by means of an explicit mapping to a semantic domain.

8 Conclusion

The paper presented a proposal for an explicit definition of the semantics of the UML association construct for conceptual modelling. Associations are not just "the simplest form of a relationship" [13]; we have argued, based on established literature, that associations represent states and events. Furthermore, we have argued, also from established literature, that cognitive concepts form the relevant semantic domain for conceptual modelling. These concepts are related to natural language syntax. Consequently, we have based our proposed semantics on an analysis of natural language. Based on the defined semantics, we have identified cognitive concepts that may be marked on associations. To this effect, we have defined a UML profile.

Identifying the cognitive concepts that are represented by associations serves to further the domain understanding that is the goal of conceptual modelling. Marking associations and association ends with cognitive concepts also reduces model ambiguity, and thus leads to clearer and more meaningful models. An example and a case study have shown that associations without the proposed profile have ambiguous meanings; a number of semantic distinctions are often implicit, and the model interpreter must derive them from domain or general background knowledge. When this domain or background knowledge is not shared between the modeller and model interpreter, the model may be interpreted incorrectly. The proposed profile makes explicit these otherwise implicit semantic distinctions and leads to less ambiguous models and more accurate model interpretation. The interpretation of the models is more likely to be consistent between different interpreters, as it relies less on shared domain or background knowledge.

Explicating the intended meaning of associations using a UML profile may also be helpful in increasingly multi-lingual development projects [111, 112], where not all natural languages of project participants mark all cognitive concepts.

While the case study has shown the feasibility of applying the proposed semantics, and has indicated that its benefits may be in a more thorough domain understanding and domain model, this may have come at the cost of increased modelling effort. Further research in this area will use experimental methods to assess the benefits and the modelling effort and quantify the trade-offs between them, using outcome measures such as accuracy, correctness, completeness, ease of use, ease of learning, time to model, time to interpret, etc. [113, 114, 115].

As language semantics may affect development methodologies and processes, further research is required on those methodological aspects that are affected by the proposed semantics, modelling profile and constraints.

References

1. Kung, C., Solvberg, A.: Activity modelling and behaviour modelling. In Olle, T., Sol, H., Verrijn-Stuart, A., eds.: Information System Design Methodologies: Improving the Practice. Addison-Wesley Publishing Co., Amsterdam (1986)
2. Offen, R.: Domain understanding is the key to successful system development. Requirements Engineering **7** (2002) 172–175
3. Iscoe, N., Williams, G.B., Arango, G.: Domain modeling for software engineering. In: Proceedings of the 13th International Conference on Software Engineering ICSE 91, Austin, TX. (1991) 340–343
4. Jackson, M.: The world and the machine. In: Proceedings of the 17th International Conference on Software Engineering ICSE 95, Seattle, WA. (1995) 283–292
5. Boehm, B.: Understanding and controlling software costs. IEEE Transactions on Software Engineering **14** (1988) 1462–1477
6. Harel, D., Rumpe, B.: Meaningful modeling: What's the semantics of "semantics"? IEEE Computer (2004) 64–72
7. Rumbaugh, J.: Relations as semantic constructs in an object-oriented language. In: Proceedings of the 1987 Conference on Object Oriented Programming Systems and Languages and Applications, Orlando, FL., ACM Press (1987) 466–481
8. Rumbaugh, J., et al.: Object Oriented Modeling and Design. Prentice Hall, Englewood Cliffs, NJ (1991)
9. Stevens, P.: On the interpretation of binary associations with the unified modelling language. Software and Systems Modelling **1** (2002) 68–79
10. Genova, G., Llorens, J., Martinez, P.: The meaning of multiplicity of n-ary associations in UML. Software and Systems Modelling **1** (2002) 86–97
11. Dobing, B., Parsons, J.: How the UML is used. Communications of the ACM **49** (2006)
12. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? Data and Knowledge Engineering **58** (2006) 358–380
13. IBM: Developing object-oriented software: An experience-based approach. Prentice Hall, Inc., Upper Saddle River, NJ (1997)
14. Bahrami, A.: Object oriented systems development. Irwin/McGraw-Hill, Boston, MA (1999)

15. Object Management Group: OMG Unified Modeling Language (OMG UML): Superstructure, V2.1.2. (2007) Document formal/2007-11-02.
16. Saksena, M., France, R.B., Larrondo-Petrie, M.M.: A characterization of aggregation. In: Proceedings of the 1998 Conference on Object-Oriented Information Systems OOIS 98. (1998) 11–19
17. Brunet, J.: An enhanced defintion of composition and its use for abstraction. In: Proceedings of the 1998 Conference on Object-Oriented Information Systems OOIS 98. (1998) 276–292
18. Barbier, F.: Systematic construction of UML associations and aggregations using cOlOr framework. In: Proceedings of the European Conference on Object-Oriented Programming ECOOP'98. (1998) 480–482
19. Henderson-Sellers, B., Barbier, F.: Black and white diamonds. In: Proceedings of the 1999 Conference on the Unified Modelling Language UML 99. (1999) 550–565
20. Shanks, G., Tansley, E., Weber, R.: Representing composites in conceptual modeling. Communications of the ACM **47** (2004) 77–80
21. Wand, Y., Storey, V.C., Weber, R.: An ontological analysis of the relationship construct in conceptual modeling. ACM Transactions on Database Systems **24** (1999) 494–528
22. Evermann, J.: The association construct in conceptual modelling - an analysis using the Bunge ontological model. In Pastor, O., e Cunha, J.F., eds.: Proceedings of the 17th International Conference on Advanced Information Systems Engineering, CAiSE 2005, Porto, Portugal, Springer Verlag (2005) 33–47
23. Opdahl, A., Henderson-Sellers, B.: Ontological evaluation of the UML using the Bunge-Wand-Weber model. Software and Systems Modeling **1** (2002) 43–67
24. Evermann, J., Wand, Y.: Towards ontologically based semantics for UML constructs. In Kunii, H., Jajodia, S., Solvberg, A., eds.: Proceedings of ER 2001. (2001) 354–367
25. Rosemann, M., Green, P.: Developing a meta-model for the Bunge-Wand-Weber (BWW) ontological constructs. Technical report, Department of Commerce, University of Queensland (2000)
26. Bunge, M.A.: Ontology I: The Furniture of the World. Volume 3 of Treatise On Basic Philosophy. D. Reidel Publishing Company, Dordrecht, Holland (1977)
27. Niles, I., Pease, A.: Towards a standard upper ontology. In: Proceedings of the 2nd International Conference on Formal Ontologies in Information Systems FOIS, Ogunquit, Maine 2001. (2001) 2–9
28. Veres, C., Hitchman, S.: Using psychology to understand conceptual modelling. In: Proceedings of the 2002 European Conference on Information Systems ECIS, Gdansk, Poland. (2002) 473–481
29. Evermann, J.: Towards a cognitive foundation for knowledge representation. Information Systems Journal **15** (2005) 147–178
30. Veres, C., Mansson, G.: Cognition and modelling: Foundations for research and practice. Journal of Information Technology Theory and Applications **7**

(2005) 93–104
31. Wyssusek, B., Klaus, H.: On the foundations of the ontological foundation of conceptual modeling grammars: The construction of the Bunge-Wand-Weber ontology. In: Proceedings of PHISE'05, Porto, Portugal, FEUP Porto (2005) 583–593
32. Lakoff, G.: Women, Fire, and Dangerous Things – What Categories reveal about the Mind. The University of Chicago Press, Chicagor, IL (1987)
33. Gopnik, A., Wellman, H.: The 'theory theory'. In Hirschfield, L., Gelman, S., eds.: Mapping the Mind: Domain Specificity in Culture and Cognition. Cambridge University Press, New York, NY (1994) 257–293
34. Murphy, G.L.: The Big Book of Concepts. The MIT Press, Cambridge, MA (2002)
35. Berlin, B., Kay, P.: Basic Color Terms: Their Universality and Evolution. University of California Press, Berkeley, CA (1969)
36. Davidoff, J., Davies, I., Roberson, D.: Colour categories in a stone-age tribe. Nature **398** (1999) 203–204
37. Frank, H., Harvey, O., Verdun, K.: American responses to five categories of shame in Chinese culture: A preliminary cross-cultural construct validation. Personality and Individual Differences **28** (2000) 887–896
38. Bloom, A.: The Linguistic Shaping of Thought: A Study in the Impact of Language on Thinking in China and the West. Erlbaum Associates, Hillsdale, NJ (1981)
39. Pederson, E., Danziger, E., Wilkins, D., Levinson, S., Kita, S., Senft, G.: Semantic typology and spatial conceptualization. Language **74** (1998) 557–589
40. Levinson, S.C., Kita, S., Haun, D.B., Rasch, B.H.: Returning the tables: Language affects spatial reasoning. Cognition **84** (2002) 155–188
41. Bowerman, M.: The origin of children's spatial semantic categories: Cognitive versus linguistic determinants. [116] 145–176
42. Bowerman, M., Choi, S.: Space under construction: Language-specific spatial categorization in first language acquisition. [117] 387–428
43. Boroditsky, L., Schmidt, L.A., Philips, W.: Sex, syntax, and semantics. [117] 61–79
44. Sera, M.D., Gathje, J., del Castillo Pintado, J.: Language and ontological knowledge: The contrast between objects and events made by Spanish and English speakers. Journal of Memory and Language **41** (1999) 303–326
45. Waxman, S.R., Markow, D.B.: Words as invitations to form categories: Evidence from 12- to 13-month-old infants. Cognitive Psychology **29** (1995) 257–302
46. Hall, D.G., Moore, C.E.: Red bluebirds and black greenflies: Preschoolers' understanding of the semantics of adjectives and count nouns. Journal of Experimental Child Psychology **67** (1997) 236–267
47. Hall, D.G., Waxman, S.R., Hurwitz, W.M.: How two- and four-year-old children interpret adjectives and count nouns. Child Development **64** (1993) 1651–1664

48. Waxman, S.R., Markow, D.B.: Object properties and object kind: Twenty-one-month-old infants' extension of novel adjectives. Child Development **69** (1998) 1313–1329
49. Graham, S.A., Baker, R.K., Poulin-Dubois, D.: Infants' expectations about object label reference. Canadian Journal of Experimental Psychology **52** (1998) 103–112
50. Hall, D.G., Waxman, S.R.: Assumptions about word meaning: Individuation and basic-level kinds. Child Development **64** (1993) 1550–1570
51. Imai, M., Gentner, D., Uchida, N.: Children's theories of word meaning: The role of shape similarity in early acquisition. Cognitive Development **9** (1994) 45–75
52. Michnick Golinkoff, R., Shuff-Bailey, M., Raquel, O., Ruan, W.: Young children extend novel words at the basic level: Evidence for the principle of categorical scope. Developmental Psychology **31** (1995) 494–507
53. Waxman, S.R., Namy, L.L.: Challenging the notion of a thematic preference in young children. Developmental Psychology **33** (1997) 555–567
54. Waxman, S.R.: Specifying the scope of 13-month-olds' expectations for novel words. Cognition **70** (1999) B35–B50
55. Hall, D.G.: Semantic constraints on word meaning: Proper names and adjectives. Child Development **65** (1994) 1299–1317
56. Hall, D.G.: Continuity and the persistence of objects: When the whole is greater than the sum of the parts. Cognitive Psychology **37** (1998) 28–59
57. Xu, F., Carey, S.: Infants' metaphysics: The case of numerical identity. Cognitive Psychology **30** (1996) 111–153
58. Xu, F.: Object individuation and object identity in infancy: The role of spatiotemporal information, object property information, and language. Acta Psychologica **102** (1999) 113–136
59. Samuelson, L.K., Smith, L.B.: Early noun vocabularies: Do ontology, category structure and syntax correspond? Cognition **73** (1999) 1–33
60. Hall, D.G.: Preschoolers' default assumptions about word meaning: Proper names designate unique individuals. Developmental Psychology **32** (1996) 177–186
61. Mueller Gathercole, V.C., Min, H.: Word meaning biases or language-specific effects? Evidence from English, Spanish, and Korean. First Language **17** (1997) 31–56
62. Mueller Gathercole, V.C., et al.: Ontological categories and functions: Acquisition of new names. Cognitive Development **10** (1995) 225–251
63. Embley, D.W.: Object-oriented systems analysis: a model-driven approach. Prentice Hall, Inc., Englewood Cliffs, NJ (1992)
64. Chen, P.P.: English sentence structure and entity-relationship diagrams. Information Sciences **29** (1983) 127–149
65. Sykes, J.: English grammar as a sentence model for conceptual modelling using NIAM. In Falkenberg, E., Hesse, W., Olive, A., eds.: Information System Concepts - Towards a consolidation of views. Chapman Hall (1995)
66. Dunn, L., Orlowska, M.: A natural language interpreter for the construction of conceptual schemas. In Steinholtz, B., Solvberg, A., Bergman, L.,

eds.: Proceedings of the 2nd Nordic Conference on Advanced Information Systems Engineering (CAiSE), Springer Verlag (1990) 175–194
67. Meziane, F., Athanasakis, N., Ananiadou, S.: Generating natural language sepcifications from UML class diagrams. Requirements Engineering **13** (2008)
68. Jackendoff, R.: Semantics and Cognition. The MIT Press, Cambridge, MA (1983)
69. Jackendoff, R.: Semantic Structures. The MIT Press, Cambridge, MA (1990)
70. Jackendoff, R.: Parts and boundaries. In Levin, B., Pinker, S., eds.: Lexical and Conceptual Semantics. Blackwell Publishers, Cambridge, MA (1992) 9–46
71. Jackendoff, R.: The Architecture of the Language Faculty. The MIT Press, Cambridge, MA (1997)
72. Talmy, L.: Lexicalization patterns: Semantic structure in lexical forms. In Shopen, T., ed.: Language Typology and Syntactic Description: Grammatical Categories and the Lexicon. Volume 3. Cambridge University Press, Cambridge, UK (1985) 57–149
73. Talmy, L.: The relation of grammar to cognition. In Rudzka-Ostyn, B., ed.: Topics in Cognitive Linguistics. John Bejamins Publishing Company, Amsterdam (1988) 165–205
74. Talmy, L.: Toward a Cognitive Semantics : Concept Structuring Systems. Volume 1. The MIT Press, Cambridge, MA (2000)
75. Talmy, L.: Toward a Cognitive Semantics : Typology and Process in Concept Structuring. Volume 2. The MIT Press, Cambridge, MA (2000)
76. Rolland, C., Proix, C.: Natural language approach to conceptual modeling. In Loucopoulos, P., Zicari, R., eds.: Conceptual Modeling, Databases and CASE: An Integrated View of Information Systems. Wiley (1992)
77. Olive, A., Raventos, R.: Modeling events as entities in object-oriented modeling languages. Data and Knowledge Engineering **58** (2006) 243–262
78. Palmer, F.R.: Grammatical Roles and Relations. Cambridge University Press, Cambridge, UK (1994)
79. Cook, W.A.: Case Grammar Applied. The Summer Institute of Linguistic and The University of Texas at Arlington, Dallas, TX (1998)
80. Falk, Y.N.: Lexical Functional Grammar. An Introduction to Parallel Constraint-Based syntax. CSLI Publications, Stanford, CA (2001)
81. Frawley, W.: Linguistic Semantics. Lawrence Erlbaum Associates, Mahwah, NJ (1992)
82. Whaley, L.J.: Introduction to Typology. The Unity and Diversity of Language. Sage Publications, Thousand Oaks (1997)
83. Croft, W.: Typology and Universals. 2nd edn. Cambridge University Press, Cambridge, UK (2003)
84. Croft, W.: Typology and Universals. Cambridge University Press, Cambridge, UK (1990)
85. Bradner, S.: Key words to use in RFCs to indicate requirement levels. Technical Report RFC 2119, Harvard University (1997)

86. Jackendoff, R.: Consciousness and the Compuational Mind. Then MIT Press, Cambridge, MA (1987)
87. Gemino, A.: Empirical Comparisons of Systems Analysis Modeling Techniques. PhD thesis, University of British Columbia, Canada (1999)
88. Bajaj, A.: The effect of the number of concepts on the readability of schemas: An empirical study with data models. Requirements Engineering **9** (2004) 261–270
89. Gemino, A., Wand, Y.: Complexity and clarity in conceptual modelling: Comparison of mandatory and optional properties. Data and Knowledge Engineering **55** (2005) 301–326
90. Evermann, J., Wand, Y.: Ontology based object-oriented domain modelling: Fundamental concepts. Requirements Engineering (2005) 146–160
91. Burton-Jones, A., Weber, R.: Properties do not have properties: Investigating a questionable conceptual modeling practice. In Batra, D., Parsons, J., Ramesh, V., eds.: Proceedings of the Second Annual Symposium on Research in Systems Analysis and Design, Miami, FL. (2003)
92. Burton-Jones, A., Meso, P.: How good are these UML diagrams? an empirical test of the Wand and Weber good decomposition model. In Applegate, L., Galliers, R., DeGross, J.I., eds.: Proceedings of the Internataionl Conference on Information systems, Barcelona, Dec. 2002. (2002)
93. Bodart, F., Patel, A., Sim, M., Weber, R.: Should optional properties be used in conceptual modelling? A theory and three empirical tests. Information Systems Research **12** (2001) 384–405
94. Fang, J., Evermann, J.: Evaluating ontologies – towards a cognitive measure of quality. In: Proceedings of the 2007 Workshop on Vocabularies, Ontologies and Rules in the Enterprise VORTE, Annapolis, MD, October 2007. (2007)
95. Storey, V.: Meronymic relationships. Journal of Database Administration **2** (1992) 22–35
96. Winston, M.E., Chaffin, R., Herrmann, D.: A taxonomy of part-whole relations. Cognitive Science **11** (1987) 417–444
97. Chaffin, R., Herrmann, D., Winston, M.: An empirical taxonomy of part-whole relations: Effects of part-whole type on relation identification. Language and Cognitive Processes **3** (1988) 17–48
98. Flynn, D., Knight, D., Laender, A.: Multiple relationships: An analysis of their semantics and their modelling. In Falkenberg, E.D., Hesse, W., Olive, A., eds.: Information System Concepts: Towards a consolidation of views. IFIP/Chapman & Hall, London (1995)
99. Jones, T.H., Song, I.Y.: Binary equivalents of ternary relationships in entity-relationship modeling: A logical decomposition approach. Journal of Database Management **11** (2000) 12–19
100. Jones, T.H., Song, I.Y.: Ternary relationships: Semantic requirements and logically correct alternatives. In Siau, K., ed.: Advanced Topics in Database Research. Idea Group Publishing, Hershey, PA (2002) 17–33
101. Dey, D., Storey, V.C., Barron, T.M.: Improving database design through the analysis of relationships. ACM Transactions on Database Systems **24** (1999)

453–486
102. Booch, G.: Object oriented design with applications. Benjamin/Cummings, Redwood City, CA (1991)
103. Coad, P., Yourdon, E.: Object-Oriented Analysis. Yourdon Press, Englewood Cliffs, NJ (1990)
104. Stevens, P.: On associations in the unified modelling language. In Gogolla, M., Kobryn, C., eds.: Proceedings of the 4th International Conference on the Unified Modeling Language UML 2001, October 1-5 2001, Toronto, ON. (2001)
105. Kristensen, B.B.: Complex associations: Abstractions in object-oriented modeling. In: Proceedings of OOPSLA'94, Portland, OR. (1994) 272–286
106. Gruber, T.R.: A translation approach to portable ontology specification. Knowledge Acquisition **5** (1993) 199–220
107. Abbott, R.J.: Program design by informal English descriptions. Communications of the ACM **26** (1983) 882–894
108. Weigand, H.: Assessing functional grammar for knowledge representation. Data and Knowledge Engineering **8** (1992) 191–203
109. Storey, V.: Understanding semantic relationships. The VLDB Journal **2** (1993) 455–488
110. Siau, K.: Informational and computational equivalence in comparing information modeling methods. Journal of Database Management **15** (2004) 73–86
111. Ebert, C., De Neve, P.: Surviving global software development. IEEE Software **18** (2001) 62–69
112. Carmel, E., Agarwal, R.: Tactical approaches for alleviating distance in global software development. IEEE Software **18** (2001) 22–29
113. Topi, H., Ramesh, V.: Human factors research on data modeling: A review of prior research, an extended framework and future research directions. Journal of Database Management **13** (2002) 3–19
114. Gemino, A., Wand, Y.: A framework for empirical evaluation of conceptual modeling techniques. Requirements Engineering **9** (2004) 248–260
115. Aranda, J., Erns, N., Horkoff, J., Easterbrook, S.: A framework for empirical evaluation of model comprehensibility. In: Proceedings of the International Workshop on Modeling in Software Engineering, Washington, DC. (2007)
116. Gumperz, J.J., Levinson, S.C., eds.: Rethinking linguistic relativity. Cambridge University Press, Cambridge, UK (1996)
117. Gentner, D., Goldin-Meadow, S., eds.: Language in Mind: Advances in the Study of Language and Thought. The MIT Press, Cambridge, MA (2003)