# Dealing with Uncertainty in Information Modelling

No Author Given

No Institute Given

**Abstract.** We present ongoing research concerning a communication-based approach to information modelling. The general goal of our research is to understand and support (contextualized) *modelling dialogues* rather than the models that result from these dialogues or the modelling languages in which the models are expressed. We take the point of view that information modelling dialogues are subject to the same kinds of uncertainty that occur in any communication between human agents. This uncertainty is for a large part due to the contextualized nature of information models. By focusing on dialogues and guiding them through strategies for dealing with uncertainty, we hope to achieve better, properly contextualized, information models. We present an analysis of uncertainty in information modelling, and give an example of a viable approach to one particular type of uncertainty reduction in information modelling. We work towards a functional design for an interactive modelling environment for testing our theories.

*information structures, domain analysis*

## 1 Introduction

Our view on information modelling is strongly *communication based*: an information model of a domain is very much seen as a model (an 'information grammar') of the language of the people communicating in that domain. It is ultimately an instrument for sharing knowledge between individuals [1]. We base our work mostly on the ORM method (also known as NIAM) [2, 3], yet in principle our approach should also be viable for other formal (and even informal) information modelling methods (for example, predicate logic, or the UML [4]). We emphatically do not focus on modelling languages, but on the modelling *process*. Hence, our approach aspires to be *modelling language independent* in the long run. We are primarily interested in the creation of *formal* information models, yet do not expect domain experts to be able to directly produce formalizations. This implies that we base our modelling approach on statements in, or closely resembling, natural language [5, 6]. Formal models are derived from the resulting set of statements.

Our communication-based approach implies that one or more conversations take place between people involved in the modelling activity. Traditionally, the two types of participant distinguished in such conversations (or *dialogues*) are the *domain expert* and the *system analyst* (we will return to this below). To our best knowledge, current information modelling literature focuses on either what information models should look like, or what modelling language they should be expressed in. Even work on derivation of models from natural language (computer-supported or not [7]) involves (translation between) representations, not on any sort of modelling dialogue.

A traditional, positivist view on information modelling discards communicative uncertainty and expects a complete and unambiguous model simply to be available 'after model creation', based on the idea that once you express a model in a formal language, everything about it is clear. We embrace a more subjectivist, situational view on information modelling, i.e. we view information models as heavily *contextualized*.

Importantly, the contextualized nature of information models [8] puts central *the actual people producing the model*, and *the interaction between them*. A way of respecting this is to indeed focus on making the right people interact in the right way to produce the desired information models.

An inherent aspect of human-to-human communication is that there commonly is, at least temporarily, some *uncertainty* on behalf of the 'hearing' party about what the 'speaking party' means or wants with the utterances she produces, and vice versa. This uncertainty is closely related to the contextualized nature of communication. It also holds for modelling dialogues, that can be seen as specialized *information gathering conversations*, aiming for *uncertainty reduction*. We propose to view uncertainty reduction as a crucial drive behind formal information modelling. In this paper, we are interested primarily in uncertainty aspects of the information modelling process, and how modelers might deal with them.

In our research at large, we are currently working towards the creation of an operational test environment in which we can observe and try out modelling strategies as take place in and steer modelling processes. This should enable us to validate and improve existing theories about information modelling and the modelling process, and eventually to guide real life modelling processes, making them more effective, more accessible to laymen, and more efficient. To achieve this, we are developing a coherent theoretical framework concerning the communicative modelling process, and on this basis are creating a concrete, implementable structure for the support of real modelling dialogues. This paper, however, involves exploration of an important part of the theoretical framework.

## 2 Uncertainty in information modelling

Analogous to [9], the following awareness levels of information need can be distinguished:

**The visceral level** The searcher subconsciously experiences that something is missing. We assume that the searcher at this stage is capable of recognizing (at least) some characteristics of what could satisfy this need.

**The conscious level** The searcher is aware of this need, and can judge the relevance of available information. The searcher may start to actively search for ways to satisfy the need.

**The formalized level** The searcher has some formulation (either implicit or explicit) of the information need. In case of an implicit formulation, a searcher can judge the relevance of a description of that need.

**The compromised level** For a number of searchers (or tasks), a compromise is reached as to the best product composition from the actual assortment.

Information modelling as seen as a communicative activity (an information gathering process) can be expected to involve, at some point, each of these levels. Consequently, the levels correspond to levels of reduced uncertainty in view of the wider modelling goals. A similar assumption is put forward in [10], indicating that system specification starts out from 'vague' ideas that are then further refined and developed. Within a modelling dialogue, then, the informal specification evolves from an incomplete and 'vague' domain description to a formal and precise specification of domain knowledge, thereby going from the visceral level up to the compromised level.

In addition, it may be uncertain what the status is of a model (or sentences in the model description) in terms of its level of acceptance by various participants. The level of agreement that is to be reached again relates to the modelling goals at large. Proper et all. [1] distinguish between three levels (related to communication amongst participants in the general information system development process):

**Aware** – An actor may become aware of (possible) knowledge by way of it being shared by another actor (possibly from outside the community), or by creating it herself.

**Agreed** – After the knowledge has been shared, an actor can make up her own mind about it, and decide wether or not to *agree* with the knowledge shared.

**Committed** – Actors who agree to a specific knowledge topic may decide to commit to this knowledge. In other words, they may decide to adapt their future behavior in accordance to this knowledge.

Identifying and resolving 'vagueness' in informal specification is a major part of the refinement task of the system analyst. Two main *types of uncertainty* can be relevantly identified [11], each raising a different class of questions within the modelling conversation:

**Epistemic uncertainty** This uncertainty exists in the mind of the individual expert, and reflects the incomplete knowledge a domain expert has of the domain. The uncertainty is a result of limited mental resources and limited time to investigate the domain [12].

**Linguistic uncertainty** This is uncertainty introduced in communication between participants, occurring when an expression in common language has more than one possible interpretation. For flexible common languages, such as natural language, this may occur frequently. Very constrained languages, on the other hand, may prevent the occurrence of multiple interpretation, at the expense of limited expressive power.

Although precise methods for handling the various types of uncertainty depend on the situation, several general approaches can be distinguished. Lipshitz and Strauss [13] investigated how decision makers handle uncertainty; they found that four general ways occurred:

**Reduction of uncertainty** Collect additional information, e.g. by asking.

**Assumption based reasoning** Fill gaps in knowledge by making plausible assumptions.

**Weighing pros and cons** of various alternatives.

**Suppression** Ignore uncertainty, at least for a while.

We view these four approaches as basic *strategies* for dealing with uncertainty in information modelling.

During the modelling process (aiming to reach a level of certainty dictated by the modelling goals), epistemic uncertainty can be addressed by asking questions of completeness. Linguistic uncertainty can be addressed by asking questions of meaning. The intention behind the modelling process can be formulated as: participants trying to reach a state of minimal uncertainty amongst them, conform the situationally required level.

If controlled language is used (which in fact is a good strategy for reducing linguistic uncertainty), participants will at least be relatively certain about the structure (syntax) of the sentences produced/read during modelling. This does not hold for the mapping of phrases or elements within those sentences onto concepts in the modelling technique used: the system analyst will still be uncertain as to whether all required elements have been mapped, and whether they have been mapped correctly. Dedicated modelling strategies will have to be used to answer such questions of meaning, but if controlled language is used in answering them, the leap from natural language to formal language is at least eased considerably.

We can combine the various distinctions above as follows. Reducing *epistemic* uncertainty (expressing and sharing knowledge about a domain) is the core goal of the modelling process. However, in order to do so, reduction of *linguistic uncertainty* is a crucial sub-goal. The degree of linguistic precision required strongly depends on the general level of uncertainty aimed at within a modelling effort. In particular, if the information need is to be satisfied at the *formal* level, linguistic uncertainty must be quite low (one might say non-existent, but in view of our contextualized, communication-based perspective on modelling, this is extremely hard to achieve). Finally, the required level of *agreement* about (parts of) the model also influences the intensity and method by which communication is to take place. This is in turn related to demands set for the required level of uncertainty reduction, and so on. Finally, modelling *strategies* will have to be chosen, roughly based on the four basic strategies for dealing with uncertainty listed above, and further specialized to cover the various levels and types of agreement and uncertainty.

The question then is: what strategies (questions to ask and ways of asking them) result in the required reduction of uncertainty, in view of the particular goals set? While we cannot currently answer this question in great depth and detail, we nevertheless propose a solution direction, and provide an example of one particular type of uncertainty reduction in information modelling.

## 3 Dealing with uncertainty in dialogues: an example

To make our approach more concrete, we will now discuss how one particular kind of uncertainty, *non-specificity*, may be dealt with in modelling dialogues. However, in order to do this, we first have to explain our generic way of modelling dialogues.

We assume that knowledge transfer between actors in the modelling dialogue is performed using the following *dialogue actions*:

| | |
|---|---|
| Propose$(a, s)$ | Actor $a$ proposes statement $s$. It does not become part of the common model until every other actor has accepted it. |
| Withdraw$(a, s)$ | Actor $a$ withdraws statement $s$. Withdrawal is the opposite of proposal. |
| Accept$(a, s)$ | Actor $a$ accepts statement $s$ as a valid statement; it may eventually become part of its internal model $M_a$. A statement can only be accepted after it has been proposed. |
| Reject$(a, s)$ | Actor $a$ rejects statement $s$, because $a$ finds $s$ unacceptable for further consideration. Rejection is the counterpart of acceptance. |
| Ask$(a, q)$ | Actor $a$ asks question $q$, to be answered by some actor. Queries can be withdrawn or answered. |
| Answer$(a, q, s)$ | Actor $a$ answers question $q$ with statement $s$; an answer functions as a special proposal. |

Based on these actions, a dialogue grammar can be composed; for details, see [14]. In addition, as a dialogue progresses, statements follow a particular life cycle:

after they have been proposed, they can be accepted, rejected, or withdrawn by actors. A *dialogue state* $DS$ is a structure that represents the state of the statements and questions in a dialogue:

$$DS = \langle A, S, Ac, Q \rangle$$

Here, $A$ is the set of actors participating in the dialogue; $S$ is the set of *active* statements, i.e., statements that have been proposed but not withdrawn; $Q$ is the set of questions that have been asked but not yet answered. $Ac$ is a total function that administrates the *acceptance state* for each combination of statement and actor:

$$Ac : S \times A \rightarrow \{u, a, r\}$$

The acceptance state '$u$' stands for undecided, '$a$' for accepted and '$r$' for rejected.

The dialogue state $DS$ can be derived incrementally from the subsequent actions in a *Dialog* as follows. Let $DS_1 = \langle A, S_1, Ac_1, Q_1 \rangle$ be the dialogue state upon performing a dialogue action. The state afterward is referred to as $DS_2 = \langle A_2, S_2, Ac, Q_2 \rangle$.

| | |
|---|---|
| $Propose(x, s)$ | $S_2 = S_1 \cup \{s\}; Ac_2 = Ac_1 \cup \{(s, y, u) \mid y \in A\}$ |
| $Withdraw(x, s)$ | $S_2 = S_1 \setminus \{s\}$ |
| $Accept(x, s)$ | $Ac_2 = Ac_1 \cup \{(s, x, a)\}$ |
| $Reject(x, s)$ | $Ac_2 = Ac_1 \setminus \{(s, x, r)\}$ |
| $Ask(x, q)$ | $Q_2 = Q_1 \cup \{q\}$ |
| $Answer(x, q, s)$ | $Q_2 = Q_1 \setminus \{q\}; Propose(x, s)$ |

After proposal, a statement remains a *proposed* statement until either (1) some actor rejects it (causing its state to change into *rejected*), or (2) all actors have accepted the statement, after which it has state *accepted*. Note that the dialogue state does not record rejected statements.

When the modelling dialogue is completed, the set $\left\{ s \in S \mid \forall_{a \in A} \left[ Ac(s, a) = a \right] \right\}$ of accepted statements essentially *is* the formal model: its statements are expressed in a controlled format that is understood by all participants. Controlled language thus may be seen as an intermediate modelling technique. As it is also understandable for a system analyst, this controlled language has a sufficiently sound formal basis.

A dialogue process is defined by the *rules of communication* that are agreed upon by all participating actors before the dialogue starts. The rules determine all relevant aspects that have to be agreed on before a useful dialogue can take place, such as:

- which dialogue actions can be used in which situation
- what is the agreed common language, i.e., the form (syntax) in which statements may be expressed

– what are the assumptions on validity of statements
– what types of uncertainty are allowed
– etc.

The communication rules for the basic dialogue process discussed below are based on (a strict interpretation of) the ORM method [2]. The process poses strict constraints on the communication between and tasks of domain expert and system analyst:

1. the controlled language has a strict format; there is no uncertainty about how to interpret statements in this controlled language
   (a) the syntax is well-defined
   (b) the *basic semantics* of sentences is well defined. This indicates the straightforward relation of syntactical elements to 'concepts' (rather than the 'real meaning' that is only available in the mind of the domain expert).
   (c) *no distortion*: expressed statements are (required to be) valid
   (d) *no irrelevance*: statements are relevant for resulting model
   (e) *specific*: at all stages of the dialogue, all information needed for a correct formal model is available.
   (f) *unambiguous*: only a single interpretation possible (results from well-defined syntax and basic semantics).
   (g) only *absence* is allowed during the dialogue (missing statements); the validation feedback-cycle works towards completeness of the model itself.
   (h) no other incompleteness is allowed
2. the domain expert is responsible for
   (a) providing a complete and valid mental model of the domain
   (b) checking a verbalized model for validity and correctness
3. the system analyst is responsible for the mapping of a set of statements onto the underlying modelling concepts

Because of the constraints on the dialogue in this basic approach, the system analyst has no other responsibility than creating a formal model from the set of accepted statements. We expect this allows in principle for full computer assistance of the system analyst's task (i.e. replacing the analyst by an automated 'module').

The following dialogue is a valid example in the basic setting as described :

```
DE> person with name John lives in city with name Nijmegen
  propose(de, person with name John lives in city with name Nijmegen)
  accept(de,  person with name John lives in city with name Nijmegen)
SA> accept
  accept(sa, person with name John lives in city with name Nijmegen)
```

From this dialogue, the system analyst is capable of creating a formal ORM model, represented by the following statements (see [14] for a detailed explanation):

```
orm-entity-type(person)    orm-entity-type(city)
orm-label-type(name)       orm-label-type(name)
orm-entity(John)           orm-entity(Nijmegen)
orm-label(John)            orm-label(Nijmegen)
orm-type(John,name)        orm-type(Nijmegen,name)
orm-type(John,person)      orm-type(Nijmegen,city)

orm-relation-type(lives-in)
orm-role-type(lives-in, 1, person) (next section)
orm-role-type(lives-in, 2, city)
orm-relation(r1)
orm-type(r1, lives-in)
orm-role(r1, 1, John)
orm-role(r1, 2, Nijmegen)
```

In the basic dialogue process, no uncertainty regarding the interpretation of statements is allowed in any form. However, we will also present an extension of the basic dialogue process, such that non-specificity is allowed in statements.

In the basic dialogue process, the domain expert is responsible for expressing domain knowledge in a very strict format. As such a strict format is not natural for people to specify in, we relax the constraints on the format, while preserving a formal dialogue grammar that allows computer support for the system analyst.

A severe constraint is the *non-specificity* constraint: "each statement has to provide enough information to derive a formal model". This includes the demand that types are explicitly given alongside instances. This forces the dialogue statements into a format that is very unnatural and impractical for the average domain expert.

*Example 1.* Compare the first specific sentence with the second, nonspecific sentence:

```
1) person with name John lives in city with name Nijmegen
2) John lives in Nijmegen
```

Clearly, the second statement is more natural and easier to specify. However, the types of the entities 'John' and 'Nijmegen' are not specified, making it impossible to incorporate statement 2 in a formal ORM model.

The new skills of the system analyst allow (and require) a more complex dialogue. Specific statements are handled the same way as in the basic dialogue; however, non-specific statements need a special dialogue strategy in order to be handled.

Note that system analysts now also must have the skill to formulate a question that, when answered, solves a non-specificity. In addition, the system analyst must be able to create a plausible assumption on what information is missing from nonspecific statements.

The domain expert also has new responsibilities:

1. Answering questions posed by the system analyst, and
2. Judging assumptions made by the system analyst
   As example, we relax the format in the following way:

```
S  -> E a E          # sentence construct
E  -> S | SN | N     # entity specifier
SN -> ET p LT L      # standard name
N  -> L              # name
```

The basic semantics of this language is equal to that of the basic format, except that entities need not be fully qualified anymore: an entity may be represented by either a standard name or a simple name (a label). In the latter case, no type information about the entity is specified. Note that other communication rules remain in place. In particular, non-ambiguity of the language requires names to be unique.

The following examples show the various ways to deal with nonspecific statements, in line with the basic strategies for dealing with uncertainty as presented in section 2.

*Example 2.* A scenario demonstrating *asking for missing information*:

```
DE> (s1) John lives in city with name Nijmegen
SA> qualify John?
DE> (s2) John is person with name John
  Accept(SA, {s1, s2})
```

*Example 3.* A scenario demonstrating *assuming information that is missing*:

```
DE> (s1) John lives in city with name Nijmegen
  assume(sa, "(s2) John is a thing")
  accept(sa, {s1, s2})
```

*Example 4.* A scenario demonstrating *temporarily ignoring missing information*:

```
DE> (s1) John lives in city with name Nijmegen
DE> (s2) person with name John works in city with name Groningen
  accept(sa, {s1, s2})
```

These simple but realistic examples of dealing with uncertainty in modelling dialogues conclude our paper. In our ongoing research, we intend to expand our theoretical exploration of modelling dialogues and strategies. We work towards the creation of an experimental environment for communication-based modelling, in order to enable empirical validation and improvement of our theoretical framework. We are also in the process of applying some available, robust natural language processing techniques to aid the mapping of controlled language sets of accepted statements onto formal structures. In the long run, we aim it computer-supported guiding and recording of modelling dialogues for various kinds of modelling and various formalisms.

## References


1. Proper, H., Verrijn-Stuart, A., Hoppenbrouwers, S.: Towards utility-based selection of architecture-modelling concepts. Technical Report NIII-R0417, Institute for Information and Computing Sciences, Radboud University, Nijmegen, The Netherlands, EU (2004) To be presented as an invited talk at: The Second Asia-Pacific Conference on Conceptual Modelling Newcastle, Australia, January 30 – February 4, 2005.
2. Halpin, T.: Information Modeling and Relational Databases, From Conceptual Analysis to Logical Design. Morgan Kaufman, San Mateo, California, USA (2001)
3. Verheijen, G., Bekkum, J.v.: NIAM: an Information Analysis Method. In Olle, T., Sol, H., Verrijn-Stuart, A., eds.: Information Systems Design Methodologies: A Comparative Review. North-Holland/IFIP WG8.1, Amsterdam, The Netherlands, EU (1982) 537–590
4. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modelling Language User Guide. Addison-Wesley, Reading, Massachusetts, USA (1999)
5. Frederiks, P.: Object-Oriented Modeling based on Information Grammars. PhD thesis, Radboud University, Nijmegen, The Netherlands, EU (1997)
6. Hofstede, A.t., Proper, H., Weide, T.v.d.: Formal definition of a conceptual language for the description and manipulation of information models. Information Systems **18** (1993) 489–523
7. Hoppenbrouwers, J., Vos, B.v.d., Hoppenbrouwers, S.: Nl structures and conceptual modelling: Grammalizing for KISS. Data & Knowledge Engineering **23** (1997) 79–92
8. Hoppenbrouwers, S.: Freezing Language; Conceptualisation processes in ICT supported organisations. PhD thesis, Radboud University, Nijmegen, The Netherlands, EU (2003)
9. Taylor, R.S.: Question-negotiation and information seeking in libraries. College and Research Libraries (1968) 178–194
10. Veldhuijzen van Zanten, G., Hoppenbrouwers, S., Proper, H.: System Development as a Rational Communicative Process. In Callaos, N., Farsi, D., Eshagian-Wilner, M., Hanratty, T., Rish, N., eds.: Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics. Volume XVI., Orlando, Florida, USA (2003) 126–130
11. Regan, H., Hope, B., Ferson, S.: Analysis and portrayal of uncertainty in a food web exposure model. Human and Ecological Risk Assessment **8** (2002) 1757–1777
12. Anderson, J.R.: Cognitive psychology and its implications. W.H. Freeman and Company (1990)
13. Lipshitz, R., Strauss, O.: Coping with uncertainty: a naturalistic decision-making analysis. Organizational Behaviour and Human Decision Processes **2** (1997) 152–154
14. Bosman, S., van der Weide, T. Technical report, Institute for Information and Computing Sciences, Radboud University, (Nijmegen, The Netherlands, EU)