PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is an author’s version which may differ from the publisher’s version.

For additional information about this publication click this link. http://hdl.handle.net/2066/111113

Please be advised that this information was generated on 2017-08-03 and may be subject to change.
Conceptualisation Support for Language Development

S. Hoppenbrouwers, H. Weigand

Nijmegen Institute for Computing and Information Sciences/

NIII-R0314 May 2003
Conceptualisation Support for Language Development

Stijn Hoppenbrouwers,
Nijmegen University (KUN), The Netherlands,
stijnh@cs.kun.nl

Hans Weigand,
Tilburg University, The Netherlands,
H.Weigand@uvt.nl

ABSTRACT
Conceptualisation, or the development of Conceptual Models, can be performed for several reasons. One important goal nowadays is to support communication systems. A communication system aims at supporting communication in or between organizations and is typically supported by IT. It is based on some language that must be developed and agreed upon first, and must be adapted or redeveloped when changes in the environment occur. Conceptualisation as part of language development becomes more and more a continuous effort. This paper introduces a phased model for the conceptualisation process. It discusses which techniques are useful in which phase and which ones can be applied in a continuous conceptualisation effort.

Keywords: conceptualisation, closed languages, meta-communication

1. Introduction

One of the fundamental questions in the field of Information Systems is what happens when representations of natural language (usually words) are built into automated systems. For example, consider a database system used by different people that uses words such as: salary, employee status, etc., or a web service catalogue that uses keywords to index web services all over the Internet. What happens in all these cases that there is some conceptualisation process that results in an explicit specification of concepts. The resulting specification is built into the system. This built-in specification is sometimes hardwired in the code, more often it is adaptable, but in all cases it takes quite some effort to change it. For that reason, we use the term “freezing language” as a metaphor.

Freezing a language is done at some particular moment, but the frozen language is supposed to be used for quite some time, possibly in various situations. Therefore language needs to be frozen with care. First of all, the frozen language must be efficient. Whether the language is used by people, or by systems, or by users through systems, it should be possible to communicate some message with as little effort as possible. The frozen language should also be clear. The terms should not be confusing or give cause to misunderstandings. These two basic requirements – clarity and efficiency – are sometimes opposed. For example, it can be efficient to abbreviate names of departments to two or three letters. Although this can be perfectly clear for insiders, it can be confusing for people less acquainted with the organization, and totally obscure to systems that try to interpret these names automatically.
A basic problem with frozen language is its closed character. The fundamental difference with natural language as used in speech or writing lies in the natural capacity of human beings to adapt their language use to occurring situations as they go. This we call the adaptability of language. The difference becomes clear when a break-down occurs. In such a case, humans can engage in a conversation concerning the language used, for example: “eh, what do you mean with X?” followed by some exchange aiming at the repair of the conversation. Such communication about communication is what we call linguistic meta-communication. To be more precise, the adaptability of language consists of two elements: the possibility to engage in a linguistic meta-communication (which is always there in natural discourse, but often not possible or hard in situations of frozen language), and the ease of adapting the frozen language (either in the form of a temporary “fix” or a real defreezing/freezing process).

Why is the closed character of frozen language a problem? The problem is that the world changes. New people enter the discourse environment. The system has to interoperate with other systems; legislation changes and because of that, terms get a different meaning; And so on. Sooner or later, the frozen language does not do its job anymore. The problems begin when people start misinterpreting certain data, or do not feel confident anymore about the “closed” data items they can use.. Locally, people may think that they solve the issue by agreeing on some interpretation, but this may not correspond with the interpretation in other groups. If trouble increases, people may try to avoid the system, or “work around” it (for example, by using a “comments” field to clarify in “open language” what the intended meaning should be of closed term X). Evidently, the efficiency of the system goes down rapidly. If the problems continue, the effectiveness of the system may become an issue and the system may be abandoned completely.

Much more can be said about this problem, but in comparison to other problems in the field of ontologies or information systems, it is relatively easy to comprehend. Unfortunately, this does not mean that it is taken serious yet. For computer scientists, the problem is usually dealt with as an extensibility problem: the software should be easy to adapt, preferably at a logical level. However, this is only part of the solution. The problem is often not taken seriously because it seems so obvious that if the language is confusing, this confusion should be solved. However, in practice this turns out to be a non-trivial problem, as anyone working in a large organization, or dealing with communication between organizations, will recognize. The authors have performed an intensive case study in a large administrative organization, and discovered many language problems that gave rise to serious deficiencies in the organizational processes. Problems that were directly related to information systems with frozen languages.

A broad (but still initial) exploration of the problem is given in (Hoppenbrouwers, fc.). In this article, we focus on one aspect only: the conceptualisation process. Freezing a language involves conceptualisation. Traditionally, this conceptualisation is performed at design time (ex ante). To alleviate the problems of frozen language, it would be desirable for conceptualisation to be possible also at run time (ex post), that is, each time that some break-down occurs. In fact, this would put conceptualisation at a much more “natural” place then when it is performed out of use context, at design time. According to (Winograd & Flores, 1986), concepts emerge from a specific and limited need to conceptualise. In their view, concepts are usually just part of the common background where they remain implicit. Reflection, in the sense of “making sense of the world”, is triggered when the background is challenged somehow.

Conceptualisation, as the building of conceptual models, has been prominent in IS design for decades. However, over the years the motivation for building conceptual models has slightly
shifted. The first conceptual models were used mainly for the derivation of database schemata. The emphasis was on the identification of entity types and relationships, and the related constraints. For that reason, the *structure* of the conceptual model was most relevant, and not so much the conceptual content, nor its lexical form. When conceptual modelling became embedded in object-oriented analysis, this emphasis did not change very much, although the identification of *activities* was added to it. The situation was different in the field of Knowledge Engineering. In this field, sophisticated techniques where developed for describing conceptual meanings. Ontologies were introduced as “specifications of a conceptualisation” to which the stakeholders commit. Ontologies provide “agreed-upon meanings”, and as such have a coordinating role. It is assumed that knowledge systems become more stable when they are built on an ontology. Today, ontologies are proposed as a tool for enabling the Semantic Web (McGuinness et al, 2002) and interoperability between systems. Since ontologies are used for building systems, they typically are founded on formal logic.

In this paper, the emphasis is on the design of languages. Languages that play a role in the communication system of the organization. It is nice if conceptualisation can be used as a basis for database design, but most important is its usability for communication. For example, suppose that the organization makes use of six customer types. For a database designer, “customer type” will probably become a sorted attribute, and at some point he will need the six values to fill as a set or list. He needs to know whether customer type is unique. For the rest, he is not interested in the semantics of the terms, how far they are understood by different users, under what name, and who is the authority that decides on the definition of these values. However, these questions are crucial when we consider the usability of the conceptualisation within the communication system of the organization.

The structure of this paper is as follows. In section 2, we introduce a process view on conceptualisation. In section 3, we analyse each phase of the process in order to see which techniques can be used in that phase. We are particularly interested in the question whether the technique can also be applied in the case of ex post conceptualisation. We close with a general conclusion.

## 2. The conceptualisation process

Conceptualisation is a process in which several phases can be distinguished. From an extensive literature survey, the following phases presented themselves:

1. **obtain raw material.** This can be either creation of raw material, as in an interview, or selection of raw material (for example, collecting and selecting relevant documentation).
2. **obtain explicit representation.** The aim of this phase is to arrive at a representation of the Universe of Discourse (UoD), that is, the communication situation to be supported and its closed language.
3. **select relevant concepts.** This can be considered to be the core phase. Which concepts should be identified? Which identified concepts need to be included in the closed language?
4. **name and define relevant concepts.** In this phase, the focus is on how the identified concepts are expressed in the closed language (terminology). This includes for example the use of homonymy and synonymy. The definition of terms can be done in several ways.
5. **check quality** (validation). The quality of the conceptualisation can be checked in various ways: checking consistency and completeness between the phases (e.g. did all identified concepts get a name?), experimental validation (including the active search for gaps or
potential problems) as well as the application of general quality criteria such as efficiency, clarity and maintainability.

Phase 1 and 2 are sometimes intimately related, but need to be distinguished analytically. It is seldom the case that an explicit representation of the UoD is readily –at hand, or can be created from scratch. This is the reason why phase 2 needs to be preceded by a phase 1, in which the raw material is collected that is used as input for the conceptualisation. Which sources to select from the available ones, or which techniques to use to acquire sources, are not trivial questions.

When the focus is on language development, it is important to be aware of the context-sensitive nature of communication. A general conceptual model of the domain can be useful, but it is also necessary to consider how agents phrase and interpret their messages in specific contexts.

3. Analysis of the conceptualisation process

This section contains an analysis of each phase of the conceptualisation process, with a focus on the main techniques that can be used in this phase. Some techniques can be used in different phases, and some cover several phases at once. For that reason, we had to make sometimes arbitrary choices under which phase to discuss a technique. A systematic overview of all phase/technique combinations can be found in (Hoppenbrouwers, fc.).

3.1 Obtain raw material

Selecting Artefacts
It is a widely accepted practice to take existing material as input for conceptualisation. In most of cases, NL texts of some sort are simply assumed to somehow provide useful input to further intuitive analysis (e.g. Rumbaugh, 1991); in some cases, text analysis is seen as a crucial and standard phase of the process for which more elaborate, even prescriptive guidelines are provided (e.g. van Dijk et al., 1989; Nijssen, 1993; Kristen, 1994). We found only a few references in the literature to guidelines concerning the actual selection of raw material.

The KISS method for OO analysis very explicitly encourages the use of existing documentation as a basis for a conceptual model (Kristen, 1994:102). The Grammalizer, a CASE tool designed to support the KISS analytical process (Hoppenbrouwers, 1997) was specifically designed to (optionally) take in fairly large volumes of existing text (up to 500 pages). However, both some explorative experiments and practical experience have shown that in particular the use of large volumes of heterogeneous text as input for quite specific conceptualisation efforts is counterproductive. It renders too many concept forms originating in too many different contexts, and the effort to select even potentially relevant concept forms from the multitude stifles the analytic process. In this respect, the difference between corpus based, general lexical analysis and explicit conceptualisation for specialised communicational contexts needs to be emphasised.

Interestingly, the original KISS method focuses on purpose-made textual descriptions. Previously created documentation is mentioned in a complementary role (Kristen, 1994:102). Apparently, the promise of the Grammalizer's capability to take in large volumes of existing text has later given rise to increased emphasis on the use of existing texts as the basis for conceptual analysis.
An exception to the general lack of explicit guidance on text selection is (Nijssen, 1993) where it is emphatically stated that the input documents used must be familiar documents. A familiar document is phrased in “the user's jargon” (ibid, p.64-6). Nijssen indeed has good reasons to demand familiar language to be used: he very explicitly wants to avoid terminological confusion. Unfortunately, he recognises the problem without acknowledging its full complexity. The seemingly simple demand for familiarity of language is not easily met, especially if a system is to be used in various discourse environments or touches upon a number of terminologies. Nevertheless, Nijssen's guideline is generally a good one: make sure the analyst is aware of the right contextual meaning of the terms it includes.

Nijssen explicitly determines the use that may be made of selected raw material: the text is to be ‘read out’ by a person familiar with the domain. What is interesting about this is that the input text is not subject to any form of analysis on a grammatical basis (as in many other approaches). Instead, the texts are used to inspire the controlled, purpose-created production of normalised clauses (a prototypical Phase 2 activity). The level of ‘conceptual purity’ and control thus achieved is remarkable. The downside is that all depends on the reader's ability to correctly interpret the text.

Assessment: selection of raw material is not commonly supported by explicit guidelines. A disadvantage of the use of existing text as the basis of further explicit conceptualisation is that there often is little control over who ‘uttered’ them, and in which context. Many instances of conceptualisation demand more control and better focus than can be attained through raw material selection. Yet once a good, reliable source of raw material can be found, the contextual properties of which are well understood and controllable, selection of raw material becomes in fact a minor issue and may well surpass creation of raw material in efficiency and comprehensiveness.

Creating Texts
A very basic form of creating raw material is plain NL text writing. A number of techniques mentioned below fall in this category (e.g. case description, scenario creation, diary keeping), but are discussed separately.

Sometimes, restrictions are imposed on the use of language, for example restricted vocabulary use and the avoidance of complex grammar. For example, in (Dijk, 1989:89), authors are instructed to use simple sentences with one noun and no conjunctions (e.g. after, before), in present tense. More generally, authors are instructed to at least carefully consider whether the vocabulary they use is appropriate, i.e. fits the domain and purpose of the text. However, it is rarely made clear what is meant with ‘appropriate’ or ‘purpose’.

Assessment: text authoring is so general as a technique that it cannot be uniformly evaluated in detail. Since it draws on skills that most educated humans have at their disposal universally –that is, to a certain degree-, it is one that can be readily and successfully applied as long as it can be made clear to the performing agent what needs to be described, and how elaborately. But if a text is written for the purpose of capturing the envisaged UoD, we are already in phase 2 (see below).

Interviews
Perhaps the most common, intuitive way of gathering information in cases involving analyst-subject interaction is the interview: a conversation in which one partly asks questions and the other answers them. Good interviewing, however, is almost an art form, just as much as good writing. It has been observed, especially in ethnography, that interviews are perceived by the interviewee as an artificial setting, and that this may seriously influence its value in data acquisition. Interestingly, (Hammerstey, 1983:119) acknowledges this limitation, but emphasises that “the distinctiveness of
the interview setting must not be exaggerated, and it may be viewed as a resource, not simply as a problem. [...] To the extent that the aim in ethnography is not simply in the provision of a description of what occurred in a particular setting over a certain period of time, there may be positive advantages to be gained from subjecting people to verbal stimuli different from those prevalent in settings in which they normally operate.” (Loucopoulos, 1995:42-3) explicitly mentions that vocabulary poses a large problem in interviews.

Assessment: interviews in context of communication-oriented conceptualisation are quite useful, in particular in a relatively open approach that nevertheless keeps the focus on generally concept-related issues. Interviews are popular as a source of raw material, but the skill involved in conducting a good interview should not be underestimated.

Think Aloud Protocols (also called “think aloud interviews”) are a crossover between a structured interview and the observation of a task. The subject is asked to verbalise any thoughts concerning the task she performs, which is recorded or noted down by the interviewer. Depending on how the subject is instructed, the information she gives may vary from practically oriented comments on the tasks performed to remarks concerning rather abstract, even associative clues reflecting ‘what the subject is thinking’ during the activity in question.

The technique quite typically belongs to the realm of knowledge elicitation (Firlej, 1991; Verhoef, 1993). They are useful in “revealing the subject's problem solving strategies, validation of conceptual information gathered in less structured interviewing, reveal sequences of events, and reveal the accompanying knowledge that a particular task requires” (Firej, 1991:112).

Observation
Observation (Spradley, 1980) and related techniques take a special place in Phase 1 conceptualisation. It is typically only the observer who has the intention to perform conceptualisation, not the agents observed. From the point of view of the agents observed, the technique thus generally falls under Phase 1, whereas from the point of view of the observer it may fall under either Phase 1 or Phase 2, depending on how actively the observer's interpretation influences the results in terms of capturing some envisaged UoD.

The applicability of observation-related techniques in actual Phase 2 conceptualisation (not just Phase 1 gathering of material) depends on whether the situation in which ‘language to be frozen’ is to be used can in fact be ‘made observable’. If the situation to be described does not yet exist, it may be simulated (for example, by means of role playing or techniques borrowed from ‘Rapid Prototyping’; see Maguire, 1997, p.112).

The technique of observations is unique in its capability of finding concepts that have never been put into words and therefore are never found by the previous techniques that rely on written text or speech. Observation for the purpose of conceptualisation is aimed at the identification of objects in the domain, or of behavioural patterns that occur several times so that one can conclude that it is not just an accidental, one-off sequence of acts. The use of observations has two major drawbacks: firstly, the technique is very costly, and secondly, its relevance for communication is problematic. If the agents in the domain never verbalized this pattern or object, apparently it is never talked about. If so, it may be relevant for problem-solving, but probably not for communication. However, if the starting point is a breakdown situation where communication has turned out to be problematic, it might be necessary to uncover tacit concepts and then observation is the best technique.
Assessment: the use of observations is a very costly technique. It is not recommendable to apply it in every case. However, when communication has broken down, it can be an irreplaceable tool. Because of the costs, the focus should be defined as sharply as possible.

3.2 Capture the envisaged UoD

Case descriptions
Textual descriptions made for the purpose of further conceptual analysis often take the form of case descriptions. They typically concern a (possibly fictional) example of some relevant situation. The level of detail may of course vary depending on the goal. However, description inevitably includes the presentation of elements, their properties, relations between them, and (depending on whether a temporal factor is included), which actions they perform, how this involves other elements, and perhaps even some cause-and-effect description. In information analysis and functional systems engineering, case descriptions are generally—and perhaps too readily—taken as the typical input text of conceptual analysis.

The KISS method (Kristen, 1994), indeed showing an implicit preference for case descriptions as a useful form of textual input, provides some guidelines at least for the style and basic scope of the case description: “The style of the textual description preferably is narrative. The description must explain ‘what’ an employee does [...] as fully as possible with activities that actually happen, with nothing left out” [...] The eventual aim of the textual description is to provide an insight into ‘who or what performs which activity in the organisation on what objects’. Textual description is therefore continued until we have sufficient material to allow us to carry out a grammatical analysis.” (ibid, pp.102-3). Rumbaugh (1991) provides some related guidance: “The problem statement should state what is to be done and not how it is to be done” (ibid, p.150).

Assessment: case description is a generally applicable, straightforward form of text authoring; however, its functionality in context of communication-oriented conceptualisation has its limitations. If it is used as part of a continuous effort, the case description should not be viewed as one comprehensive text because in practice it is hard to keep such a text up to date. Rather the goal should be to maintain a collection of cases, and whenever a new unforeseen situation is met, this case description can be added to this collection, that need not be consistent. One way of doing this is by so-called diary keeping (Hammerstey, 1983:134).

Scenarios
Scenarios arguably are not unlike case descriptions, but they typically have a procedure-oriented focus. They are more intensely concerned with roles, tasks, and dynamic aspects of the case. This positions scenario writing closer to description of actual operations and therefore to actual communication.

Scenarios are often used to describe one particular kind of task. For example, they are often meant to describe a concrete activity of a user working with a computer system in order to achieve a particular goal (Rumbaugh, 1991:170). In his approach, scenarios are characterisations of users and their task in a specified context. Dynamic modelling is not by nature easily expressed by just a set of (grammatically and textually independent) factual sentences; a sequential story has to be told. Sometimes a scenario is specific enough to include fictional dialogues.

An advantage of scenarios in general is that they attempt to capture a more or less complete context in which all relevant tasks, and therefore also communication about them, are systematically carried out. If this approach is extended with a reality check as a result of the scenario being acted
out, then scenarios represent a very strong basis for the exploration of communication in specific contexts. In some cases, CASE tools offer some support for the structuring of the conceptualisation process along the lines of a scenario (Loucopoulos, 1995:49).

The RESPECT handbook (Maguire, 1997) also has much to say about scenarios. Its strongly user-oriented approach goes beyond text writing and includes the acting out of scenarios (ibid, pp.112-4). After the development team, relevant stakeholders, and the facilitator are gathered, intended users are identified, as well as their tasks and the general context. This provides the basis for the scenarios. Next, the user goals are functionally decomposed into operation needed to achieve them. Task time estimates and completion criteria are assigned to tasks as usability targets. The session is then started, and the acting out of the various tasks is recorded and transcribed.

Clearly, this approach carries much further than mere scenario writing. Its most interesting aspect lies in the fact that a scenario is not conceived as a text and literally written down, but roles and tasks are specified in detail and then acted out. The approach is relatively open and, if carried out well, will render a realistic scenario with some potential for pointing out communicational pitfalls. In addition, the approach renders far more than the scenario alone, but includes identification of tasks and roles. In general, the more elaborate way of dealing with scenarios as described in the RESPECT handbook falls under Phase 2 rather than Phase 1 conceptualisation and is more explicitly related to the technique of dialogue construction.

Generally speaking, scenario writing is a better basis for communication-oriented conceptualisation than its more static counterpart, case description. The process of scenario writing thus is considerably more fit for use in Phase 2 than case description. Obviously, scenarios are excellent input for Phase 3, provided their focus matches the focus of the general conceptualisation process at hand.

Assessment: scenario writing is a good, generally applicable technique. It is better fit for communication-oriented conceptualisation than case description, but slightly harder to implement, mostly because it needs a more particular focus and involves more detailed description.

A useful variant of scenario writing is the construction and description (possibly even recording) of dialogues. With this we mean the capturing of actual or fictional dialogues representative of dialogues actually occurring in the domain analysed. Provided that the people uttering or writing them are knowledgeable of the domain focus their effort correctly, the description of dialogues renders literal utterances as might later be supported by some information system. If the dialog involves closed language, they are either a solid way of testing whether the concepts selected ‘work’, or they may be the basis of further analysis (Phase 3-4) actually rendering a closed language specification.

Dialogue descriptions may be an integrated part of scenarios. In addition, various techniques involving role playing, but also mock-ups or prototypes of information systems and information system interfaces, may be used to create a realistic environment for acting out a dialogue (which then needs to be recorded). If dialogue construction is conducted in an advanced enough fashion, it may provide detailed insight into differences between conceptual frameworks as used by particular stakeholders, and the conditions in which particular concepts are used (or not). However, in order to acquire such detailed linguistic information, substantial skills, capacity, and support for observation and analysis need to be available. Dialogue description seems one of the most effective and to-the-point techniques for capturing actual utterances as may occur in either an existing or an envisaged discourse environment.
Assessment: very good basic technique for capturing actual samples of communication; more effective and to-the-point than objective, non-interfering observation, and more precise and true to reality than many other forms of data gathering. Excellent basis for experimental approaches to conceptualisation, but may require highly skilled facilitation if used in complex situations.

Seeking Conceptual Differences
Provided that some sort of case description, scenario, or dialogue description is available, it may be effective to confront people in a domain with fact descriptions or communicational utterances (presented perhaps even as a conceptual ‘reference model’) and to ask them whether they would ‘use the same words’, or what their own phrasing of the utterance would look like. This technique may be used simply to trace homonyms and synonyms, but it may also uncover deeper underlying differences between the conceptual frameworks that individual people wield in the domain. Of course, whether or not these differences should be taken along in the final conceptualisation (Phase 3-4) then remains to be decided upon.

The technique seems fit for continuous use (even for monitoring); if circumstances are favourable, people working in a domain may be instructed to report spotted conceptual differences at their own initiative (‘checking by using’, see Section 2.5; also diary keeping), possibly making use of some standard form or template.

Assessment: seeking conceptual differences is a general technique, which makes it hard to evaluate. Its comparative approach pinpoints matters that may be crucial in any environment that faces the combination or clash of terminologies or conceptual systems. When included in an operational process, is quite efficient because it enables the recording of breakdowns without further need for systematic analysis.

3.3 Select relevant concepts

Grammatical Analysis
Grammatical analysis as part of conceptualisation seems to be restricted mostly to the field of Conceptual Modelling (covering requirements engineering, data and information analysis, and certain approaches to knowledge engineering) and is described sometimes in classic grammatical terms, sometimes somewhat differently. References can be found throughout the literature, for example concerning Entity-Relationship diagrams (Chen, 1983), NIAM/ORM (Nijssen, 1993; Halpin, 1995), KISS (Kristen, 1994; Hoppenbrouwers, 1997; Frederiks, 1995), GRAMMARS (Dijk, 1989), OICSI (Rolland, 1992) (see Burg, 1997:141-4 for a good overview); also, in a less NL-oriented setting, in (Rumbaugh, 1991; Booch, 1994; Booch et al., 1998) and many other OO approaches.

Typically, a combination of syntax-based word categorisation and semantic categorisation is applied. Techniques for grammatical analysis generally aim for identification of potentially relevant objects (nouns) actions (verbs) and attributes (adjectives) that may be reflected in some language artefact (usually text), but the analysis may be far more advanced. Because conceptual representations of the kind we are concerned with are rather firmly related to words, it is no wonder that grammar-based techniques are generally aimed at identification of word categories such as nouns, verbs, adjectives etc., and with rather basic phrase structures that show words in a clear relation to each other, displaying subcategorisation restrictions (Radford, 1988). More advanced syntactic analysis is generally superfluous.
Experience in the Grammalizer Project (Hoppenbrouwer et al, 1997) has shown that analysts have considerable problems with accepting advanced grammatical analysis as a core procedure in conceptual modelling. Besides having trouble performing it correctly, they often refuse to carry it out systematically because it is “not their kind of work”. Automatic parsing provides some relief, but also introduces the risk of the analyst accepting its output without further considerations of quality and relevance. We strongly suspect that in practice, similar problems hamper grammatical analysis as part of other CM approaches as well.

Closely related to grammatical analysis, a source text can be subjected to semantic analysis (e.g. Brahethvik & Gulla, 2001). With this we mean not an in-depth analysis of the possible interpretations of the text, but (less ambitiously) the recognition of semantic types. Most typically, objects and actions are looked for. The close relation with grammatical analysis lies in the straightforward fact that in most cases, objects and actions are grammatically marked (typically, as nouns and verbs). However, many subtle semantic categories have been distinguished throughout the literature, and in many cases it is far from easy to accurately and conclusively derive semantic types –even the straightforward ones- exclusively on the basis of grammar.

Assessment: using grammatical analysis as an explicit technique is moderately useful in context of small texts (say, up to five sentences). It entails several aspects that may be a threat to conceptualisation as much as a benefit. However, as a more implicit technique and in combination with semantic category analysis, it is extremely useful. Heuristics for using grammatical marking in conceptual analysis are thus particularly valuable if they are internalised (i.e. acquired by the analyst on an intuitive level). If automated, the technique may cause conceptual analysis to strand in an explosion of potentially relevant concepts.

Procurement of Existing Collections or Standards
Especially if a domain involves the use of some sort of standardised terminology (or has strong links with such a domain), it may be a good idea to actively embrace standard concepts. What we mean by ‘standards’ is previously described sets of concepts, in other words, products of deliberate linguistic meta-communication. Clearly, identification of individual concepts has already been performed in order to produce some concept collection. The term ‘standard’ may be too strong in some cases; it is possible that a collection of concepts is not explicitly set as a standard, but just happens to be available and useful. We thus extend the general idea to any explicitly described coherent collection of concepts.

Assessment: procurement of existing conceptual frameworks can be a lifesaver as well as a nail in an organisation's coffin, depending on many factors. This is why for complex cases, procurement processes might justifiably become subject to a specialised research effort. If we simply focus on the option of ‘copying’ instead of building’, we can rather lamely conclude that this is indeed a good idea in some cases, as long as one is very sure of the stability of the conceptual situation, or is prepared to invest in further adaptation of the framework if the need arises.

3.4 Name and define relevant concepts

We have arrived at the most obvious kind of activity relating to explicit conceptualisation: that of the description of individual concepts. By means of Phase 3 conceptualisation, the envisaged UoD has been broken down to a set of concepts (a ‘frozen language’) that is a selected subset of the concepts occurring in some envisaged UoD and that is to be used in ICT-supported communication
in the domain to which the UoD belongs. The concepts selected in Phase 3 may be represented in various ways and by means of various meta-languages, but the descriptions are primarily given for reference purposes: they are ‘working descriptions’, not ‘final descriptions’. In Phase 4, the final description of the concepts in view of its future use in (meta-)communication is faced.

We assume every concept to exist of a form and a meaning. There are two main issues concerning concept form:

1. If several terminologies are concerned, either because they are part of inter-domain language use (various sub-languages within the general domain) or intra-domain language use (interaction with some ‘foreign’ language domain, possibly within a domain specialised in such interaction), specification of differences and relations between terminologies involved is crucial. This boils down to careful considerations of homonyms and synonyms across terminologies.

2. For individual concept forms, various practical considerations may come into play, mostly related to implementation of the information system: medium (character sets, fonts, graphics, technical compatibility between formats, etc.); data structure (data syntax, database instructions or programming languages, etc.; all related to `machine readability’); in addition, matters of ‘human readability’ (perception, cognitive ergonomics, unfortunate associations with other terms, etc.).

**Sorting Out Homonyms and Synonyms**

This technique is fairly simple in essence, and though the importance of recognising homonyms and synonyms has been widely recognised (thought usually as things to avoid), hardly anything is ever said about how one goes about doing it.

As we are discussing operational techniques, not theory, we take a pragmatic approach here: forms and meanings are ‘the same’ if they are considered sufficiently similar to be called so by a representative group of users, in a particular situation of use. In cases of doubt, simple testing of homonyms and synonyms is possible: see whether concepts can both be used in the same context (in terms of either form or meaning) without relevant consequences.

Two different activities concerning homonyms and synonyms can be recognised:

- Identification of homonyms or synonyms
- Deliberate creation/maintaining of homonyms or synonyms

Identification of existing homonyms and synonyms is a matter of understanding similarities between linguistic patterns, usually as occurring in different language domains. Comparison is the key activity. For both comparison of explicit form and meaning, complete representational similarity is not usually required for items to be considered ‘the same’. For example, differences in spelling or graphical representation (form) or differences in definition format or formulation (meaning) are not enough to declare a concept ‘different’. Consequently, identification of ‘sameness’ should mainly be carried out on the basis of active interpretation and evaluation by human agents, based on ‘intuitions of sameness’ rather than similarity in representation. This is not easy to automate, but existing lexical resources with homonym/synonym information such as WordNet (Miller, 1987) can be instrumental.

Once homonyms and synonyms have been recognised, they may or may not be deliberately maintained. In many CM-oriented approaches, homonyms and synonyms are seen as matters that
need to be ‘resolved’ in order to reach a high quality conceptual analysis. This is because homonyms and synonyms may cause ambiguity of representations, which in many design-oriented situation is indeed undesirable. However, the language situation in complex domains, in particular inter-domain communication, may require a quite different stance, requiring homonyms or synonyms to be tolerated, even embraced. Here are some simple questions that should be asked in every case:

1. Why would it be desirable to maintain a particular homonym or synonym?
2. Would it be effective to exchange one of the concept forms in the set of synonyms by another form in that set, thus resolving synonymy?
3. Would it be effective to choose a different word form for one of the two similar word forms in a set of homonyms, thus resolving homonymy?

Assessment: at a basic level, dealing with homonyms and synonyms in conceptualisation simply is unavoidable in one way or another. Systematic application of techniques can be very expensive, and should be reserved for complex situations –in which case they will be crucial. An informal technique is easily applicable in a continuous conceptualisation effort. Whether it is easy to implement a change depends on the extensibility of the system and its multilinguality support.

Writing Definitions

Creating Informal Definitions
The writing of definitions in natural language (though possibly with the aid of graphic representations; Galinski and Picht, 1997), is an activity that is deeply rooted the traditional fields like lexicography, but also in many professionals fields (e.g. legal and medical practice, science, and increasingly also ICT). It generally results in brief texts aiming to optimally describe the meaning of some word or phrase. What distinguishes it from raw material is its well-developed focus on meaning description and its strict occurrence in a meta-communication context.

High cost is probably the greatest downside of definition authoring. In particular, the maintenance of definitions can be a task that requires substantial resources. An interesting way of optimising meaning description is stepwise lexical decomposition (Dik, 1989). This approach involves definition of words by use of other words, and to continue this recursive process until the parties feel sufficiently confident.

Creating Formal Definitions
Formal definitions do not differ all that much from informal definitions in essence, it is just that they are much more carefully defined, within a conceptual framework that (if constructed properly) leaves no ‘loose ends dangling’ in terms of meaning. Once formally captured, meaning can be handled with the same kind of operational certainty as can a well-designed machine. However, with the advantages this has, some disadvantages come in. One is that sometimes linguistic ‘underdeterminedness’ can be highly functional (for example, if priority is given to understandability by a large, heterogeneous public, but also if ‘fuzziness’ and underspecification are needed as part of a process of gradual ‘meaning discovery”).

Once formal definition is part of a functional operational process, it might well be more maintainable, reliable, and portable than its informal counterpart, due to its more controlled nature and the conceptual and operational discipline that it necessarily introduces into the process. However, this benefit is reduced because the control of the formal definition process will often bog down in lack of expertise, resources, even stamina of agents involved.
Assessment: definition authoring (in whatever form) is an activity that, due to its functional as well as its non-functional properties, should be performed with an firm eye on its costs. If it has to be done, the possibilities are many and diverse: selection of approaches and representational means should involve careful considerations of how and by who the definitions will be used.

3.5 Concept Quality Checks

This phase is concerned with the validation of concepts and the aptness of concept descriptions. We discuss two main approaches to concept checking: systematic checking, and ‘checking by using’. Checking whether a set of concepts is still valid is an important element of the ‘conceptualisation feedback cycle’ (see Hoppenbrouwers, f.c.). Assuming that not all required changes in languages (as triggered by changes in their domain) can be foreseen, concept checking is what triggers focused concept change, and thus is a crucial element in concept evolution as part of information system evolution.

The most common, ‘natural’ approach to the checking of concepts is simply triggered by doubts or questions concerning concepts or concept descriptions as encountered as that concept is used. It thus concerns an ‘ex post’ sort of action. It is usually related to operational language use, though agents involved may only become aware of some deficiency through ‘conversation about language’. In fact, checking by using is typically part of more general meta-linguistic conversation (for example, concerning discussion of a particular utterance in context). However, it goes beyond the mere exchange of information about some concept: it involves the active investigation of what may be wrong with some concept or concept description, possibly to the point at which conceptualisation techniques are called in for support. Note that the latter presumes the availability of such tools, as well as the willingness and incentive on behalf of the involved agents to engage in active checking.

Checking by using thus typically (but not necessarily) concerns ‘non-mediated’ meta-communication: results of checking by using may be used directly for the resolving of some language breakdown, or even be linked to some form of ‘intervening’ meta-communication, i.e. result in a change of the information system or documentation involved. Results of concept checks, or even just a note concerning the encountered problem, may also be reported to some authority, who then may or may not take further action.

The systematic checking of concepts is an activity that entails both a clear sense of purpose (evaluating –certain aspects of– concepts, or concept descriptions: see Wright and Budin, 2001, p.870-1) and a prescribed method (in order to be systematic, some previously considered, rational line of action needs to be involved, and quite possibly carried out iteratively). The approach by nature opens possibilities for automation. Systematic checking can of course also be carried out ‘by hand’, using protocols.

Assessment

Ex post concept checking is closely related to the notion of ‘checking by using’, and therefore mostly intertwined with regular activities belonging to phases 2-4. In contrast, systematic concept checking can more typically be seen as a specific activity apart from other phases, and does not combine well with continuous conceptualisation.
4. Conclusion

To conclude, we briefly go through the conceptualisation phases again. Phase 1 is mostly oriented on ex ante approaches. The most promising extension of techniques to an ex post approach is to (automatically) collect/record communication samples, possibly including meta-communication. Such an approach is similar to the use of ‘translation memory’ in translation industry; this notion might be extended to ‘communication memory’. Phase 2 also covers diverse techniques, but they are fewer and more specialised—which is no surprise since the entire phase is more specialised. Phase 2 seems not so much ‘underdeveloped’ in terms of techniques as to suffer from the fact that it is very hard to predict what the good terms might be for a communication situation that yet has to come into existence. Experimental approaches may be a way out here. Phase 3 techniques do seem underdeveloped, in particular with respect to directly communication-oriented techniques. With exception of the interrelated techniques especially oriented on identification of candidate concepts (i.e. grammatical analysis, semantic typing, and to some degree coping with homonyms/synonyms), this phase involves attempts to acquire and apply explicit knowledge concerning the ‘how’ of concept selection. The central observation concerning Phase 4 remains that it should be pursued not further than strictly necessary. Word meaning can be described in so many ways and with emphasis on so many aspects that careful pre-selection of means (meta-languages but also procedures, techniques for editing and representation, and various other supporting mechanisms) is called for. Design of Phase 4 conceptualisation processes, if it occurs at all, is currently mostly performed within an ex-ante context. Ex-post extensions are virtually non-existent. Assuming that ex-post concept description will at times become necessary, and given the relatively low efficiency and usability of most concept description techniques, it seems worthwhile to develop appropriate meta-communicational description patterns and templates for operational, on-the-fly meta-communication. Importantly, not only language but also meta-language should somehow be tuned to language and communication practices that people are familiar with. Conceptualisation may thus perhaps be ‘brought to life’ (within some evolutionary set-up) by increasing efficiency and usability along the same lines as attempted with general information systems. For Phase 5, checking by using is most appropriate; in an ex post situation this occurs in a natural way.

One way of extending this research is to design a conceptualisation support system that can be made available behind any operational system. In this way, ex post (continuous) conceptualisation can be supported. However, we stress that such a system itself will not be sufficient; to cope with language evolution, the organization will need a comprehensive communication architecture.

References


