Towards Improved Mechanisms for Communication about Language in Information Systems Development

S. Hoppenbrouwers

Nijmegen Institute for Computing and Information Sciences/

NIII-R0326 November 2003

Nijmegen Institute for Computing and Information Sciences
Faculty of Science
Catholic University of Nijmegen
Toernooiveld 1
6525 ED Nijmegen
The Netherlands

Towards Improved Mechanisms for Communication about Language in Information Systems Development

Stijn Hoppenbrouwers
Nijmegen Institute for Informatics and Information Science,
University of Nijmegen, the Netherlands
stijnh@cs.kun.nl

Abstract

In view of dissatisfaction with the current static approach to language definition and use in information system development, we propose the initial development of a more dynamic, adaptive approach that is *complementary* to the static one. We should become more aware of the differences between various sorts of language and the way they do or do not match certain communicative uses. If language is to be used adaptively, meta-communication mechanisms should be available in order to achieve adaptability of both agreements about language and the computerised tools reflecting such agreements. Such mechanisms should be an integral part of "information system development systems".

Keywords

Information system development process, adaptive systems, user language, data structure, meta-communication

The main issue

We address some general issues of language and meaning as occurring in information system development and use. Language is an increasingly crucial linking pin in the development of Information and Communication Technology (ICT): it aligns human, socio-cognitive meaning with symbol-based constructs as used in engineering and computation. We are dissatisfied with the predominant view of language as a static construct. We insist that natural language use for conveying meaning is essentially *adaptive* and that languages *evolve and crystallise* in such use. Language is primarily a means for *creating shared meaning*, which is an adaptive process. We believe that to consider language to be a static phenomenon contributes greatly to the static nature of current information systems and that it hampers their functionality as media for communication (Hoppenbrouwers, 2003). In addition, language creation and adaptation in ICT extends beyond the description of "user language": it involves many different stakeholders and the entire range of languages and sub-languages they use (see the ArchiMate project: Jonkers et al., 2003). Since the call for dynamic, adaptive, evolving information systems is becoming increasingly loud, adaptive approaches to language should be looked at seriously. However, we should at the same time investigate their obvious limits in view of countering factors like technologically required stability, and standardisation (Hoppenbrouwers, 2003).

View on language

Our view on language differs from the mainstream view in linguistics (for example, that of generative linguistics) as well as that in mainstream ICT. Some clarification of our view on language is therefore called for.

- We view language primarily as a tool for communication (functional view)
- Our view on language starts out from language use for human-human communication
- "Language items" (either words or more complex forms) are seen as *combinations of language form and language meaning*. These can be separated in analysis, but from a functional point of view their combined status is crucial
- Knowledge of language resides in the minds of individuals, but also is of a social nature in that it aims to enable the aligning of meanings (interpretations) between individuals who speak a *sufficiently similar* language
- Both contextualised and decontextualised language are important, because it is precisely the delicate interplay between "generalised" and "instantiated" conceptualisation and interpretation that makes language work.

Though our main point of reference is "natural language", we do include in our view artificial forms of language: programming languages, mathematical languages, even schematic modelling languages like the UML (Booch et al., 1998). Also we refer here to the emerging debate concerning the link between "social" or "socio-cognitive" meaning (human-interpreted meaning) and "formal" meaning (machine readable meaning, arguably purely syntax); see (Grant Clark, 2003; Hoppenbrouwers, 2003). However, we prefer to talk about "socio-cognitive language *use*" and "formal language *use*" instead of "socio-cognitive language" and "formal language". For example, if a programmer uses a programming language, this generally involves both "formal meaning" (strictly syntactic constructs, in this case technologically tied to *precisely defined operational semantics*) and "socio-cognitive meaning" (the meaning given to the various linguistic items in the formally structured code *by the programmer's cognitive system and, to some extent, in line with other programmers' interpretation*).

Most symptoms of "languaging going wrong" are not primarily some violation of syntax, but the *failure to communicate what is needed in a particular situation* (context). This boils down to misunderstanding or non-understanding, possibly rooted in inadequate means for some person to express herself and thereby failure to "cause some intended meaning/interpretation in another being". Note that to language successfully, *enough* shared meaning has to be created (Hoppenbrouwers, 2003; Veldhuijzen van Zanten et al., 2003). What is "enough", and how it is to be conveyed, depends entirely on the situation. It may require anything from elaborate discourse and definitions, to a mere nod.

"Information System Development Systems" and Theory

By means of theories, we try to say something interesting or useful about some generalised, stable pattern or situation in an essentially dynamic world. Theories thus may concern dynamic phenomena, but the theory as such is intended to be stable. So with theories, we generally "try to hit an immobile target". Information, Organisation, and Business Process modelling as used in the design of information and communication systems, is not unlike the creation of theories. A careful, essentially static description is made of some particular domain, including the behaviour therein (processes, coordination, communication) and the language used by agents performing this behaviour (concepts, data structure, terminology). The tools currently used for "making" such theories are quite similar to the ones used in science: formal and semi-formal methods, mathematical notations, etc. This is partly the case because the use of advanced technology forces people to be "exact" or "formal" in describing domains, but also because we are accustomed to charting domains this way; it makes us feel in control.

However, complex and dynamic organisations and information system domains are by nature not very well suited for this approach: they are "moving targets". Consequently, the "theories" that information system developers produce need to be constantly changed, and the tools used for creating and writing down the theories cannot cope with this. It would be a better idea not to approach information system development as the development of a stable theory by default. If indeed we want relatively stable theories, we should perhaps not only theorise about information systems as products, but about the systems that bring forth information systems: "information system development systems". We assume that these systems are at least more stable than the information systems they produce; stable enough to be subject to scientific theorising. The resulting theoretical framework should cover both traditional static approaches and evolutionary approaches to information systems (Proper, 1994). The product and the development process should thus be modelled as one dynamic system. Note that both information systems and information system development systems are socio-technical (sub)systems. They emphatically include the human agents who use and create advanced tools for information exchange and communication.

Language and Information Systems

Language functionality is an essential part of most information systems (databases, transaction systems). Such information systems are essentially used by humans to "talk to each other", by means of a predetermined language. Therefore, their design includes a restricted set of language items, often referred to as "data structure", the product of "information analysis" of a “Universe of Discourse” (Halpin, 1995). The data structure of some domain is an important part of the "theory" covering that domain.

However, the data structure of information systems is generally based on the language of human users. It reflects, and should support, communication through natural language use (Frederiks, 1997). Taking a domain language as a solid basis for theorising about the information exchange in that domain conflicts with the fact that language use in "natural" communication is essentially dynamic. Imposing long-lasting static restrictions on language (what we call "freezing language") often damages its communicational functionality, though it is often unavoidable given the requirements of most information systems. Fortunately, negative effects can be remedied in various ways (Hoppenbrouwers, 2003). Not only may language be changed, but also clarifying information about it can be exchanged.

We distinguish three main areas of language use in information system development:

- Symbol-based engineering (programming languages)
- Language use enabled by the information system (user language)
- Inter-stakeholder communication (including modelling language and numerous different terminologies)

In our view, the use *and* creation/adaptation of language in all of these areas should be better aligned with the specific language usages.

In line with the search for stable theorising about information systems by focusing on the systems that produce them (Veldhuijzen van Zanten et al., 2003), we look at the sub-systems for creating and adapting the "frozen language" in a system, striving to create options for an evolutional approach to language freezing (and de-freezing). This requires language specification that moves away from the "theorising" approach to language description, which is rather limited. Instead it moves towards adaptive language specification as an essential part of general evolutionary systems development. We envisage the development of improved mechanisms and techniques for adaptive language description. Such techniques will be used by both system users and system developers.

Linguistic Meta-Communication

We propose an alternative to the predominant focus in ICT on *representations* and s*pecification* of language form (chiefly data structure) and language meaning (meta-data structure or other forms of definition). Though we recognise the vital importance of representations, we focus on *what needs to be done in order to get people to align their language in some situation or domain*. The action in question can generally be characterised as *linguistic meta-communication*: communication about language. We distinguish between types of linguistic meta-communication along two main lines: *anticipatory* (ex ante) as opposed to *reactive* (ex post), and *constructive* (intervening) as opposed to *informative* (non-intervening). Linguistic meta-communication can take place at various conceptual and linguistic levels (e.g. syntax, semantics, pragmatics); note that in principle, pragmatic information can be included in explicit agreements about language.

General Approach to Improving Communication about Language in ICT

We intend to pursue advancement in the following areas of research (though not necessarily with context of the ArchiMate project):

- Work towards better understanding of linguistic meta-communication processes, in particular patterns and strategies for it (through experimentation as well as study of real practice).
- Explore how a cybernetic approach to adaptive languaging might contribute to our general understanding of the problem
- Create a basic, generic, formal model of linguistic meta-communication (“action for achieving shared meaning”, enveloping and extending to “using representations to share meaning”), and link it to existing (semi-)formal models of language and meaning (data structures, ontologies, semantic web, etc.), thus replacing the predominant theoretical model of “linguistic meta-communication by representation” with one of “linguistic meta-communication as action”
- Develop a framework for comprehensive requirements engineering aimed at language use, linguistic meta-communication, and explicit conceptualisation (language description), in an ICT context
- Develop more advanced support for linguistic meta-communication (protocols, techniques, tools, groupware, dedicated and effective communication and authoring environments);

establish critical links with representation-oriented approaches to language and meaning (meta-languages, repositories, editors)

Acknowledgement: This paper results from the ArchiMate project, a research initiative that provides concepts and techniques to support architect in the visualization, communication and analysis of integrated architectures. The ArchiMate consortium consists of ABN AMRO, ABP, the Dutch Tax and Customs Administration, Ordina, Telematica Instituut, Centrum voor Wiskunde en Informatica, University of Nijmegen, and the Leiden Institute of Advanced Computer Science.

References

Booch, G., Jacobson, I., and Rumbaugh, J. (1998); The Unified Modeling Language User Guide. Addison-Wesley.

Frederiks, P. (1997); Object-Oriented Modelling based on Information Grammars; PhD Thesis, University of Nijmegen.

Grant Clark, K (2003); The Social Meaning of RDF; Web article: **http://www.xml.com/pub/a/2003/03/05/social.html;** March 05, 2003.

Halpin, T. (1995); Conceptual Schema and Relational Database Design; Prentice Hall, second edition.

Hoppenbrouwers, S. (2003); Freezing Language: Conceptualisation Processes across ICT-Supported Organisations; PhD Thesis, University of Nijmegen.

Jonkers, H., van Buuren, R., Arbab, F., de Boer, F., Bonsangue, M., Bosma, H., ter Doest, H., Groenewegen, L., Scholten, J. G., Hoppenbrouwers, S., Iacob, M., Janssen, W., Lankhorst, M., van Leeuwen, D., Proper, E., Stam, A., van der Torre, L., and Veldhuijzen van Zanten, G. (2003); Towards a language for coherent enterprise architecture descriptions; Proceedings of 7th IEEE International Enterprise Distributed Object Computing Conference (Brisbane, Australia).

Proper, H.A. (1994); A Theory for Conceptual Modelling of Evolving Application Domains; PhD thesis, University of Nijmegen.

Veldhuijzen van Zanten, G., Hoppenbrouwers, S., and Proper, H.A. (2003); System Development as a Communicative Process; Proceedings of the Seventh Multi-Conference on Systemics, Cybernetics and Informatics (Orlando, FL).