Overcoming Language Barriers by means of Computers

Roland Hausser

Universität Erlangen-Nürnberg
Abteilung Computerlinguistik (CLUE)
rrh@linguistik.uni-erlangen.de

Abstract

Language barriers arise between (i) different natural languages, (ii) different media of language, i.e., speaking, writing, and signing, and (iii) different kinds of agents, e.g., humans and machines. Up to now, these language barriers have been approached with different methods, e.g., the design of universal languages or machine translation for (i), systems of automatic speech recognition and synthesis for (ii), and programming languages for (iii).

This paper presents a unified approach for overcoming language barriers by modeling the mechanism of natural communication on the computer. The advantages are that users do not have to learn universal or programming languages, the accuracy of transfer between language media is much improved, and human-computer communication becomes optimally user-friendly.

1 Introduction

The natural languages are the first and primary means of communication between humans. Furthermore, different natural languages have been praised for their beauty, their expressive power, their versatility, their subtle reflection of particular cultures, and their powerful support to thinking.

At the same time, natural languages have been maligned by some as ambiguous, imprecise and misleading, and unsuitable for scientific description and reasoning. They have also been accused of dividing humanity by creating language barriers.

To overcome the limitations of natural language, there have been numerous attempts at designing a universal language. In Western thought, this project has a long tradition. During the 13th–18th century[1] it was motivated by three[2] goals:

[1]Some notable scholars engaged in this project were Ramon Lull 1232–1315, *Ars Magna*, originally *Ars compendiosa inveniendi veritatem*; Francis Bacon 1561–1626 *De Augmentis Scientiarum* (1623); René Descartes 1596–1650, letter to Mersenne, dated Nov. 20, 1629; Athanasius Kircher 1602–1680 *Ars Magna Sciendi* (1669); George Dalgarno 1626–1687 *Ars Signorum, vulgo Character Universalis et Lingua Philosophica* (1661); Gottfried Wilhelm Leibniz 1646–1716, remark on Descartes' letter to Mersenne (1664) and elsewhere.

[2]A fourth goal, popular in the Middle Ages, was the reconstruction of 'The language of Adam.' For an overview see Eco 1997.

1.1 DESIDERATA OF A UNIVERSAL LANGUAGE

1. To serve as an international auxiliary language, enabling people with different native languages to communicate with each other,
2. as a scientific language, providing a simplified system of symbolism for the exact expression of all actual and possible scientific knowledge, and
3. as a logical calculus, providing a powerful instrument of the human mind for the demonstration of contradictions and the discovery of new truths.

It is especially Gottfried Wilhelm Leibniz who is credited with the idea that these desiderata should be fulfilled jointly by one and the same language. However, 'Leibniz never did much himself to develop this system of symbolism though he was constantly urging others to do so.'[3]

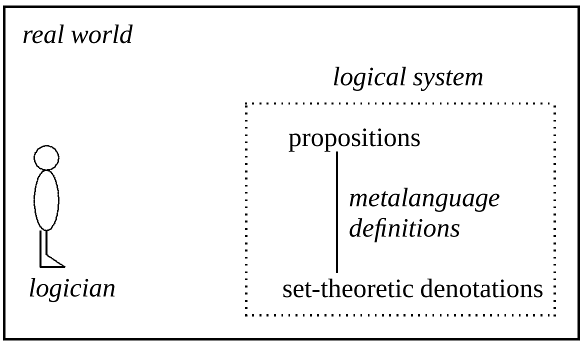
More recently, the three goals of 1.1 continue to be pursued, though mostly separately. Prominent efforts are Zamenhof's 1887 Esperanto, the program of the Viennese Circle, and Frege's 1879 Begriffsschrift, respectively.

Yet the dream of a common universal language has not been fulfilled, despite its potential usefulness and the repeated efforts. As a first step toward understanding the reasons for this let us explain the specific *functioning* of different kinds of languages, i.e., the logical, programming, and natural languages.

2 Three basic kinds of languages

The logical languages are artificial, designed by logicians for characterizing truth based on axioms and rules of inference. Truth is depicted as a relation between the *propositions* of a formal language and the 'world' defined as a set-theoretical *model*. The derivation of the propositions (syntax), the relation between the propositions and the model (semantics), and the model itself are all created as formal constructs by means of *definitions* which are formulated in a *metalanguage*.

2.1 RELATION BETWEEN A LOGICIAN AND A LOGICAL SYSTEM



The logical system is created by logicians for the purpose of proving consistency, deriving theorems, etc., but does not provide a concrete[4] communication partner – in

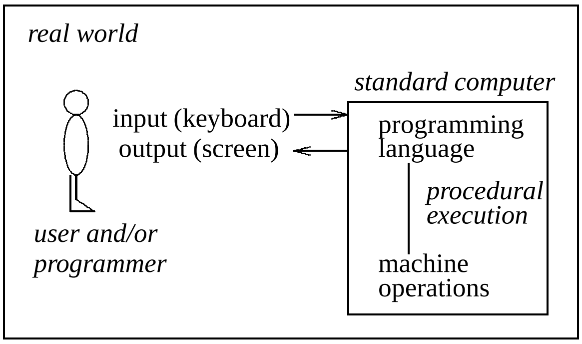
[3] J. Cohen 1954, p. 50. See also Mittelstrass 1970, pp. 435–45. According to Maat 1999, Leibnitz did something about the project, he just did not complete it.

[4] As indicated by *dotted* lines of the inner box representing the logical system.

contrast to programming systems (cf. 2.2 below) or natural language systems (cf. 2.3 below). The approach is nevertheless called 'realist'[5] by its proponents, because they claim that the structures described by their metalanguage definitions, e.g., sets, exist in the real world – a claim duly reflected in 2.1.

Another example of an artificial language is the programming languages. They are designed by computer scientists for interaction with a computer, based on a language input channel (keyboard) and a language output channel (screen). Inside the computer, the commands of the programming language are executed as machine operations (interpretation) the result of which is coded into expressions of the programming language (production) and presented to the user.

2.2 RELATION BETWEEN A USER AND A PROGRAMMING SYSTEM



Programming systems provide the user with a concrete[6] communication partner, i.e., the standard computer – in contrast to logical systems, which aim at a characterization of truth. Furthermore, expressions of programming languages are interpreted and produced on the basis of *procedural execution* – in contrast to logical systems, where everything is merely (meta)language-based definitions.[7]

The logical and the programming languages have in common, however, that they

1. are artificial languages designed by specialists for specific purposes,
2. are interpreted with respect to artificially defined worlds (models),
3. establish fixed relations between expressions and their referents, and
4. use a fixed, rigidly defined syntax defining well-formed expressions.

It is because of the latter two properties that logical and programming languages can fulfill the universal language desiderata 2 and 3 relatively easily (cf. 1.1).[8]

[5] The ontology underlying the logical languages has been classified as [–constructive, –sense], cf. Hausser 1989, Section 12.5, Hausser 2001a, or Hausser 1999/2001, Section 20.4.

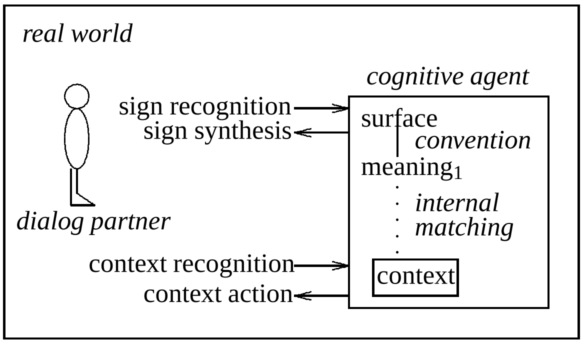
[6] As indicated by *solid* lines of the inner box representing the computer system.

[7] Contrary to widespread belief, the transition from the metalanguage-based definitions of most logical systems to corresponding procedural operations is usually impossible. For example, a computer cannot check *infinite sets* of *possible worlds* to determine the truthvalue of a proposition like Peter could be sleeping.. See Hausser 1999/2001, Sections 19.4 and 20.3, for more detailed discussion. For remarks on Prolog see Hausser 2001b, footnote 1, as well as Hausser 1999/2001, p. 383.

[8] At the same time logical and programming languages are inherently unsuitable to fulfill desideratum 1, i.e., communicating freely, either between humans (by providing a universal language for overcoming the language barriers raised by different native languages) or between natural and artificial cognitive agents (by providing user-friendly human-machine communication).

The natural languages (NL) – in contrast to the logical and the programming languages – have not been designed by specialists for certain tasks or purposes. Instead they have evolved naturally over time in their speech communities. They are characterized by a flexible handling of reference, meaning that the same sign type with the same literal meaning may be used to refer to a wide range of different referents in different contexts. This is the reason why the natural languages are uniquely suited for communicating freely about new and unexpected situations.

2.3 RELATION BETWEEN A DIALOG PARTNER AND AN NL-SYSTEM



An NL-system interacts with the external world at two levels, language and context – in contrast to standard computers, which are restricted to the language level. The two levels are related to each other cognitively by (i) lexically assigning literal meanings (meaning_1) to the language surfaces and (ii) matching between the literal meanings and corresponding cognitive structures at the level of context.

3 Incompatibility of communication and calculation

The functioning of the natural languages differs from that of the logical and programming systems in at least four ways:

1. The natural languages distinguish between the literal meaning of their signs, called *meaning*$_1$, and the corresponding referents at the level of the internal *context* – in contrast to the logical and the programming languages, where the referents themselves are used as meanings.

2. The natural languages establish the *surface-meaning*$_1$-relation on the basis of *conventions* (de Saussure's first law) which every speaker-hearer has to learn – in contrast to the logical languages, where the corresponding relation is based on metalanguage definitions, and the programming languages, where it is based on procedural execution.

3. Cognitive agents using natural language interact with the *real* world in terms of contextual *recognition* and *action* – in contrast to standard computers (which have only the programming language channel, lacking vision, locomotion, and other forms of recognition and action) and logical systems (which are based on artificial set-theoretic models and do not provide for any recognition or action at all).

4. The reference mechanisms of natural language are based on the *internal matching* between literal meanings[9] and contextual referents – in contrast to the logical and the programming languages, which are based on fixed, rigid relations between language signs and their referents.

It follows from the different functioning of the logical, programming, and natural languages that the three desiderata for a universal language listed in 1.1 cannot be fulfilled simultaneously – pace Leibniz. This is because communicating freely about new situations requires the natural languages' flexible reference mechanism based on internal matching. A calculus for demonstrating contradictions and discovering new truths, in contrast, must rely on a fixed relation between language surfaces and their referents, as is characteristic of the logical and programming languages.

4 Specialized variants of natural languages

The language barriers blamed on the natural languages are of two different kinds. One consists in misunderstandings caused by the alleged ambiguity and imprecision of the natural languages and arises between speakers of the same language. The other arises between speakers of different languages.

The attempt to overcome the first kind of barrier artificially by means of logical languages has a more gradual and widely practiced counterpart, namely the natural development of *technical* language. Technical language evolves not only in science, but also in such fields as law, construction, medicine, and even astrology.

Compared to colloquial language, technical language is characterized by pragmatic restrictions on the internal matching between the meaning_1 and the corresponding context (cf 2.3). Thus, the technical variants of natural language weakly resemble the logical and programming languages insofar as they fix the relation of reference between the language and the context, at least to a certain degree. Nevertheless, even the most technical variant of natural language must maintain the basic principle of internal matching in order to handle new situations of a known type.

The second kind of barrier is usually overcome by learning foreign languages. However, since this is time-consuming and there are many different natural languages, it is impossible to learn them all. For this reason there have been great efforts to design 'planned languages'[10] such as Esperanto.[11]

Planned languages are a special type of universal language, intended for international communication rather than for precise scientific expression or as a logical calculus. They are designed to be culturally neutral, in contrast to widely used natural languages like Mandarin Chinese, English, or Spanish. Also, they should be easy to learn and pleasant to use.

Planned languages are seemingly paradoxical in that they are artificial in the sense of being human-made, yet natural in the way they function. It is because of this functioning that planned languages are suitable for communicating freely about new situations. Furthermore, like colloquial natural languages, they are known to be subject to gradual changes, leading to the formation of dialects.

---

[9] I.e., the concept types of symbols, the pointers of indexicals, and the private markers of proper names, cf. Hausser 1999/2001, Chapters 3–6, and 2002a.

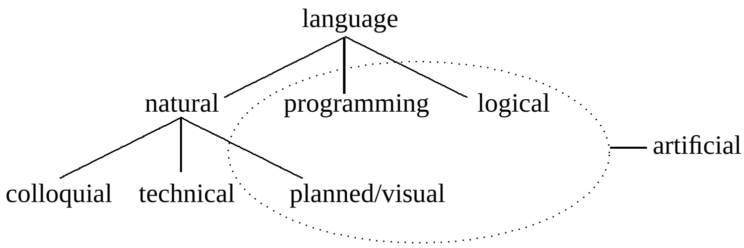
[10] Cf. Blanke 2001 for an overview. A natural – in the sense of 'non-planned' – way to overcome the barrier posed by two different languages is the creation of a Pidgin or Creole. See Liu 2001.

[11] For an analysis of Esperanto see Schubert 1988, 1993. Other examples of planned languages are Interlingua, Occidental-Interlingue, Romanid, Novial, Nov Latin, and Universal-Glot.

The same is true for 'visual' languages such as ASL (American Sign Language) or DGS (Deutsche Gebärdensprache), used by deaf people. While planned languages are spoken and written, visual languages are *signed*, using the hands, facial expressions, and body language.[12]

The different types of language discussed above may be classified as follows:

4.1 A functional hierarchy of the different language types



This functional hierarchy shows that the distinction between natural and artificial languages is not as simple as it might seem at first glance.

5 Degrees of understanding

The approaches to overcoming language barriers discussed so far have in common that they require users to *learn* additional languages: several foreign languages, a universal language, a logical language, etc. This is a serious disadvantage. Can it be avoided by automatic methods based on the technology of computers?

Unfortunately, the computers of today raise a language barrier of their own: in order to communicate with these machines, users must either learn programming languages and type in their commands, or learn graphical user interfaces and click through their many icons.

Much more comfortable than typing in or clicking through would be simply talking with the machine in one's natural language. This, however, would require the computer to *understand* natural language, at least to some degree.

An example of a low degree of language understanding is speech recognition, i.e., the automatic transfer from the medium of spoken language into the electronic medium. This would enable the user to speak the commands of a programming language into the computer instead of typing them in. For example,

user speaks: *Two plus two equals what?*
computer recognizes speech, computes, and synthesizes answer:
Two plus two equals four.

There are many applications where this kind of 'understanding' would be highly welcome, such as sending or receiving email while driving a car, riding a bike, or taking a walk. The procedure in question exemplifies low-level understanding because it consists in no more and no less than a transfer from one medium of language into another (here from spoken to digitally-coded written language).

An example of higher level understanding would be the interpretation and execution of a command like *Put the coffee on the table!* For this a transfer from spoken language

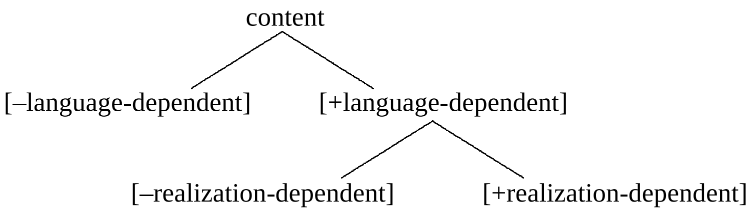
[12] In addition to national sign languages like ASL and DGS there exists the international sign language Gestuno, developed to overcome the language barriers created by the national sign languages.

to the type written commands of a programming language would not be sufficient. Instead, the cognitive agent has to be able to understand which objects the words *coffee* and *table* refer to in an open range of real world utterance situations, and what it means to execute the action *put_on*.

6 Degrees of abstractness in content representation

The transition from lower to higher level understanding is characterized by increasing abstractness in the representation of content, expressed by a hierarchy based on the features [±realization-dependent] and [±language-dependent]:

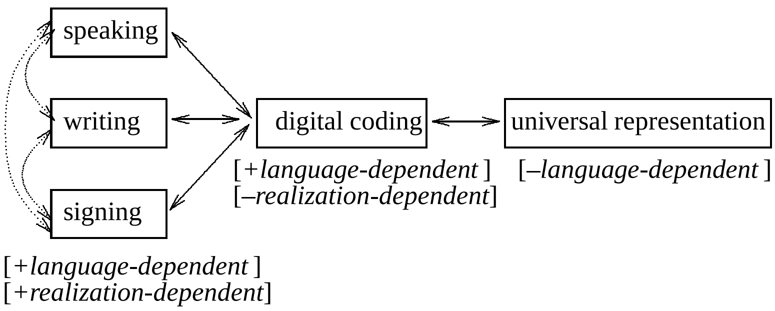
6.1 ABSTRACTION HIERARCHY FOR REPRESENTING CONTENT



As an example of a [–language-dependent] representation of a content consider a cognitive agent without language, e.g., a dog, recognizing *The cat is on the mat* and storing this content in memory.[13] In agents with language, this content may be represented in many different [+language-dependent] ways, for example in English as The cat is on the mat, in French as Le chat est sur le tapis, or in German as Die Katze ist auf der Matte. Such a [+language-dependent] representation may in turn be [–representation-dependent], such as digital or neural coding, or [+representation-dependent], such as speech, writing, or signing.

In a cognitive agent with language, e.g., a talking robot, the different media and levels of abstraction for representing content are related as follows:

6.2 TRANSFER BETWEEN MEDIA AND DEGREES OF ABSTRACTNESS



Traditional methods of transferring from one [+realization-dependent] medium to another (cf. dotted double arrows) are exemplified by a dictation-taking secretary or a

[13]The mistaken assumption that any representation of content requires language is discussed in Hausser 1999/2001 pp. 63–65.

sign language interpreter. There is also transfer within a given medium, such as transferring handwriting into print or print into tactile form, e.g., Braille. This kind of transfer requires that the human is able (i) to recognize the signs of the source medium and (ii) to generate the signs of the target medium.

Traditional methods of transferring from a [+realization-dependent] medium to a [–realization-dependent] medium, e.g., digital coding, are exemplified by humans typing spoken, written, or signed text into the computer. This requires that the human is able to recognize the signs in the source medium, whereas transfer to the digital medium is accomplished automatically by the machine.

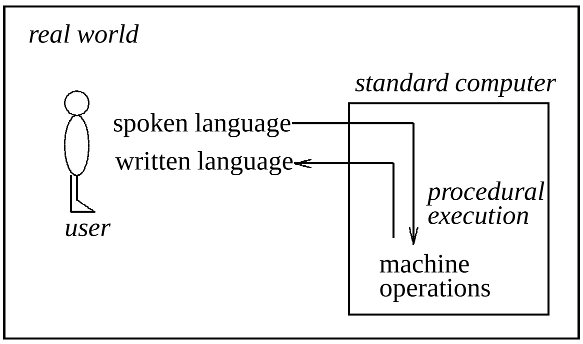
Traditional methods of transferring content between a [+language-dependent] and a [–language-dependent] representation are exemplified by human translation from one natural language into another – at least under the assumption that high quality translation requires a more or less complete *understanding* of the source language and a corresponding coding in the target language.[14]

7 Methods of automatic transfer

As an alternative to traditional methods of transfer, automatic methods are being developed. Today's methods of automatic transfer into the medium of digital coding are (i) automatic speech recognition for spoken language, (ii) OCR (optical character recognition) for written language, and (iii) automatic gesture recognition for signed language.

Current methods of automatic transfer are based on the interaction with and the functioning of standard computers (cf. 2.2). This is illustrated by the following schematic representation of a current speech recognition system:

7.1 USER'S INTERACTION WITH CURRENT SPEECH RECOGNITION



As a consequence of this structure, the functionality of natural language communication, as schematically characterized in 2.3, cannot be modelled and the transfer must be accomplished without the system understanding natural language.

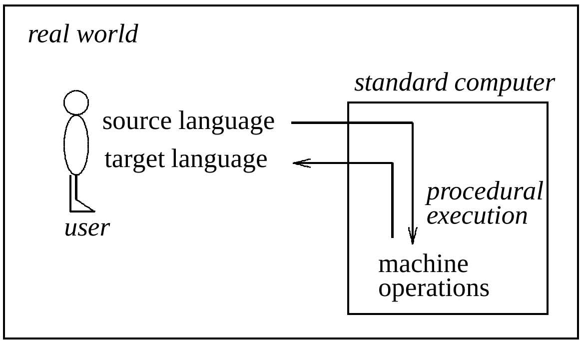
Today's most popular approaches to speech recognition are based on statistical models which generate a sequence of word hypotheses from the acoustic signal. The problem is correct word form recognition despite the phonetic, acoustic, within-speaker, and across-speaker *variabilities* arising in real world applications. These variabilities result

[14] A data structure for the universal represention of content is presented in Hausser 1999/2001 and 2001b. For a detailed analysis of the transfer between [+language-] and [–language-dependent] representations in the speaker and the hearer mode see Hausser 2002a.

in an enormous search space, for which reason robust, domain-independent, speaker-independent, continuous speech systems with a realistic vocabulary, i.e., comparable to an average human's, are still way out of reach.

Similar considerations hold for today's systems of machine translation. Like automatic speech recognition, they are based on the interaction with and the functioning of standard computers:

7.2 USER'S INTERACTION WITH CURRENT MACHINE TRANSLATION



As a consequence, such systems can only *process* natural language like a numerical problem. There is no theoretical basis for modelling *understanding*, such as successfully referring to the contextual structures intended by the speaker, proper inferencing, and correctly integrating the new information into the current data. Instead, practical translation systems take great pains to systematically work around understanding. Not surprisingly, their quality leaves much to be desired.

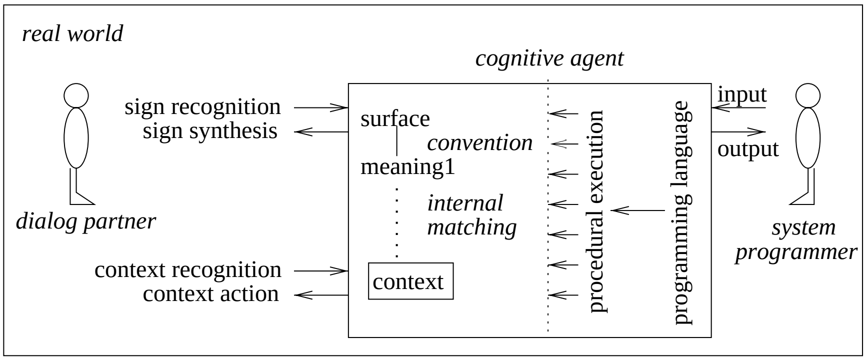
To improve the situation, systems of speech recognition, machine translation, and other automatic methods for overcoming language barriers are utilizing more and more linguistic knowledge, such as lexica, grammars for morphology and syntax, frequency information from corpora, etc. While this is a step in the right direction, it can not overcome the basic defect of using no theory of language at all (as in a pure engineering approach) or using theories of language (such as structuralism, behaviorism, model theory, speech act theory, or nativism) which were never designed to computationally model natural language communication and are consequently completely unsuitable for the task.[15]

In short, the attempt to gradually adapt the programming languages to the natural languages, as in 7.1 and 7.2, is just as futile as the attempt to turn a natural language into a programming language while maintaining free communication. In order to finally get beyond the current patchwork of low quality special purpose applications,[16] the computer software must be designed to model the overall functioning of natural language communication directly, as in the following schema:

[15]Considering the amount of time and money spent on natural language applications in the last fifty years, there is little to show for it – as poignantly illustrated by the recent bankruptcy of the world-wide industry leader.

[16]From the viewpoint of managing scientific research, the current situation is as inefficient as it is costly. What few seem to understand is the need for a coherent, functional, computationally suitable theory of language which models how communication between the speaker and the hearer actually works. Without such a theory, natural language technology will never be satisfactory.

7.3 ADEQUATE SYSTEM MODELLING NATURAL COMMUNICATION



Here, the function schemata of the programming languages 2.2 and the natural languages 2.3 are combined into one computational system. In order to implement the necessary recognition and action procedures at the levels of language and context, the hardware basis of a standard computer is not sufficient, however. What must be used instead is an autonomous robot with visual and auditory input, speech, locomotion, and manipulators.

The software part on the right-hand side of 7.3 serves to reconstruct the necessary components of natural language interpretation and production. Thereby the data structure of the context as well as the input and output procedures must be constructed first (cf. Hausser 2001b, 2002a). This requires the definition of concept types used for accomplishing contextual recognition and action, and concept tokens used for representing the result of these procedures in the data structure.

For the reconstruction of the natural language at hand, the concept types are *reused* as literal meanings which are lexically attached to surfaces. This setup is suitable for accomplishing reference: the language meaning defined as a type matches corresponding tokens at the level of context (internal matching). Another requirement is an algorithm suitable for reading content into and out of the database as well as for inferencing and querying (cf. Hausser 1992).

In addition, there must be components for surface recognition and synthesis, a morphological analyzer and generator, and a syntax-semantics component for interpretation and production relative to the level of context (cf. Hausser 2001b). Finally, there must be a control structure (cf. Hausser 2002b) which includes rules of pragmatic interpretation. For a detailed development of the overall system see Hausser 1999/2001, for an overview see Hausser 2001b.

8 Conclusion

The system described schematically in 7.3 can recognize, count, and manipulate geometric objects like squares, triangles, and circles of different colors, and it can describe in natural language what it is currently recognizing, reasoning about, or doing. It can also understand natural language to a considerable degree. For example, a command like Put the coffee on the table can be understood in different real world situations and properly carried out.

This kind of understanding provides the optimal means for reducing the search space of, for example, automatic speech recognition. A system understanding natural language follows the incoming meanings by providing suitable referents at the level of context. Each node traversed is connected to only a relatively small number of continuation nodes, providing extremely low perplexity. In addition, the system's strictly time-linear algorithm for morpho-syntactic analysis and semantic interpretation is perfect for speech recognition. These straightforward means of providing the long awaited and much-needed improvements on the robustness, range, accuracy, and speed of today's speech recognition systems can remove a most resistent barrier to inter-media transfer.

Understanding is also key for high quality machine translation. In such a system, the transfer of content from a speaker to a hearer of the same natural language is relatively easily extended to a transfer between different languages: instead of a monolingual system's connecting interpretation of language A with production in language A, a multilingual system connects interpretation of language A with production in language B. The resulting automatic high quality, high speed translation system obviates the need to learn other natural languages (which in many ways is a pity). Thus, an important barrier to inter-language communication can be overcome.

Finally, for human-computer communication an artificial system able to understand natural language and to respond in it would be the most user-friendly. Software programming is a way of telling the machine what to do. The better a machine can understand natural language, the less the need to use the programming language channel. Thus the learning of programming languages may become obsolete for humans – though the robots for building, programming, and servicing the system described must still 'learn'[17] them..

Remark: This paper benefitted from suggestions by Verena Mayer, Munich, Bavaria, Germany, Virginia Swisher, Pittsburgh, Pennsylvania, USA, and Haitao Liu, Xining, Qinghai, China .

References

Blanke, D. (2001) "Vom Entwurf zur Sprache," in K. Schubert (ed.), pp. 37-89.

Cohen, J. (1954) "On the Project of a Universal Character," *Mind* 63.

Eco, U. (1997) *The Search for the Perfect Language*, London: Fontana.

Hausser, R. (1989) *Computation of Language, An Essay on Syntax, Semantics and Pragmatics in Natural Man-Machine Communication*, Symbolic Computation: Artificial Intelligence, Berlin-New York: Springer-Verlag.

Hausser, R. (1992) "Complexity in Left-Associative Grammar," *Theoretical Computer Science*, Vol. 106.2:283-308, Dordrecht: Elsevier.

[17] Learning in the sense of having the robot upload a suitable software. This way of 'learning' is considerably easier than teaching a human.

Hausser, R. (1999/2001) *Foundations of Computational Linguistics, Human-Computer Communication in Natural Language*, Berlin, New York: Springer-Verlag, 2nd Ed.

Hausser, R. (2001a) "The Four Basic Ontologies of Semantic Interpretation," in H. Kangassalo et al. (eds) *Information Modeling and Knowledge Bases XII*, Amsterdam: IOS Press Ohmsha.

Hausser, R. (2001b) "Database Semantics for Natural Language," *Artificial Intelligence*, Vol. 130.1:27–74, Dordrecht: Elsevier.

Hausser, R. (2002a) "A Hypothesis on the Origin of the Sign Types," in A. Gelbukh (ed.) *Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science, Berlin-New York: Springer-Verlag.

Hausser, R. (2002b) "A Control Structure for Artificial Cognitive Agents," in H. Kangassalo et al. (eds) *Information Modeling and Knowledge Bases XIII*, Amsterdam: IOS Press Ohmsha.

Liu, H. (2001) "Pidgins, Creoles and planned languages - Linguistic Evolution under special conditions," in K. Schubert (ed.), p. 121-177.

Maat, J. (1999) *Philosophical Languages in the Seventeenth Century: Dalgarno, Wilkins, Leibniz*, Doctoral Dissertation, University of Amsterdam.

Mittelstrass, J. (1970) *Neuzeit und Aufklärung*, Berlin: de Gruyter.

Schubert, K. (1988) "Ausdruckskraft und Regelmässigkeit: Was Esperanto für automatische Übersetzung geeignet macht," *Language Problems and Language Planning*, 12(2), 130-147.

Schubert, K. (1993) "Semantic compositionality: Esperanto word formation for language technology," *Linguistics*, 31, 311-365.

Schubert K. (ed.) (2001) *Planned Languages: From Concept to Reality*, Brussels: Hogeschool voor Wetenschap en Kunst.