Technology and Disability 26 (2014) 93–103
DOI 10.3233/TAD-140407
IOS Press

Inclusive AAC: Multi-modal and multilingual language support for all

Mats Lundälv[a,*], Sandra Derbring[a], Katarina Heimann Mühlenbock[a], Annika Brännström[b], Bengt Farre[b] and Lars Nordberg[c]
[a]*DART (Sahlgrenska Univ. Hospital), Göteborg, Sweden*
[b]*Androtech Consulting KB, Göteborg, Sweden*
[c]*Femtio Procent Data AB, Göteborg, Sweden*

Abstract. The Concept Coding Framework (CCF) technology represents a long term commitment to develop and deliver an open infrastructure for multi-modal and multilingual language support for a wide area of applications. In this way the varying needs among several smaller groups of users of AAC (Augmentative and Alternative Communication) may be met as part of a more inclusively designed [1] mainstream environment of much broader interest. With support from the EU via the AEGIS project, a "CCF-SymbolServer" has been developed. It can be installed locally on any of the major desktop platforms (GNU/Linux, MacOS X and Windows), or online, to support many kinds of local or web based services and networked mobile systems. In any of these environments the CCF-SymbolServer can provide its multilingual and multi-modal representation services to other applications. Three such applications, developed and tested with users within AEGIS, are presented: 1) CCF-SymbolWriter, an extension for symbol support in LibreOffice/OpenOffice Writer; 2) CCF-SymbolDroid, an AAC app for Android mobile devices; 3) the new CCF supported version of Special Access to Windows (SAW6). Three current follow-up projects are briefly presented in the outline of perspectives for further research and development.

Keywords: AAC, Universal Design, AT, accessibility, graphical symbols, literacy, cognitive impairment, free and open-source software

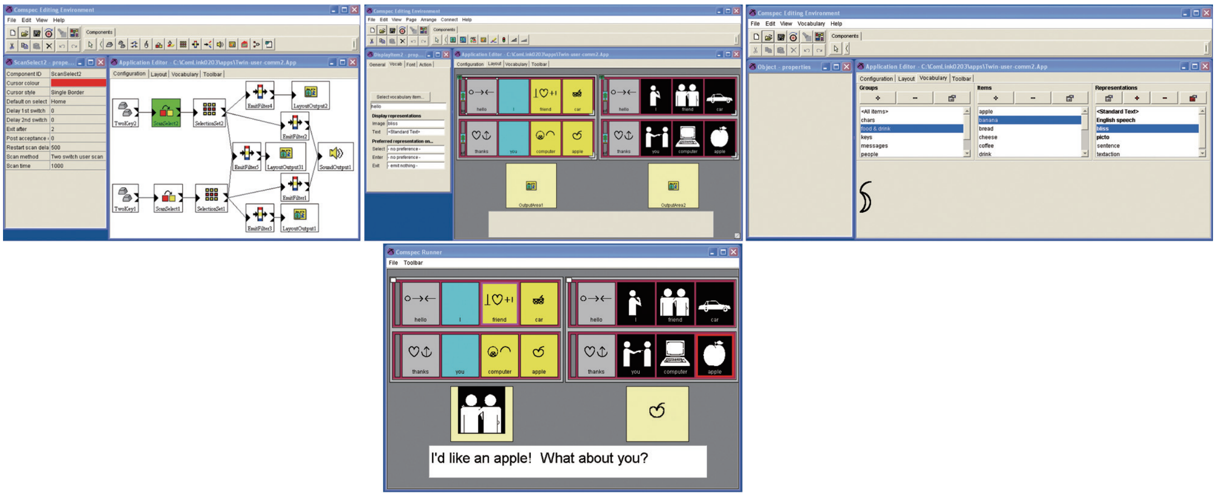
1. Introduction

A strengthened awareness is gradually spreading among us who are involved in supporting people depending on AAC for their communication, that the longer-term rights and needs of these individuals can only be well satisfied in an environment where the necessary attitudes, methodologies and tools for AAC support are very well and enduringly established. But this, on the other hand, is not what we have so far seen happening on any larger scale. We may experience temporary and local success stories, but on the whole the situation is still frustratingly unsatisfying and achievements fragile. What specific problems are we facing? What could and should be done to substantially improve the situation? In what ways must we change our approaches to meet the challenges ahead?

If we just focus on the Assistive Technology (AT) aspect, a reasonable hypothesis for responding to these questions and challenges will be: Do all we can to improve the preconditions for a general Inclusive and Universal Design [1,2] approach to achieving a more inclusive mainstream environment by offering options for long-term support for AAC needs, which are (and to be effective must be) of general value for the environment as a whole. In the Concept Coding Framework (CCF) [3] effort, the ambition is to step-by-step develop an open and free framework technology infrastructure for making multilingual and multi-modal language resources available in mainstream ICT environments as well as in AT.

There have been several attempts to approach these goals during the past two decades. Already in the early

[*] Corresponding author: Mats Lundälv, Kruthusgatan 17, 411 04 Göteborg, Sweden. Tel.: +46 705 298093; Fax: +46 31 155829; E-mail: mats.lundalv@vgregion.se.



I'd like an apple! What about you?

Fig. 1. An example of a Comspec-ComLink application (bottom-centre), generated from the three main system editors; the Configuration Editor (top-left), the Layout Editor (top-centre) and the Vocabulary Editor (top-right). The displayed application features a dialogue setup for two users with two different types of alternative input, translating between two different symbol systems with parallel speech and text output. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/THC-140407)

1990:s the Nordic-British ComSpec project (1991–1993) formulated the vision of a generic modular and component based AT platform for AAC software, including a central multi-modal language module [4]. This was brought further in the EU TIDE Comspec project (TP1169) [5] – with partners from Great Britain, Holland, Germany, Portugal, Norway and Sweden – which resulted in the ComLink open and component based Java environment for developing and adapting AAC systems for persons with multiple disabilities [6]. The language component and editor for linking multi-modal lexical representations to each other is at the core of this environment, and it addresses the needs, and provides tools for end-users, facilitators, system integrators and component developers. Figure 1 shows a ComLink configuration.

Parallel to this, the visions from the Nordic Comspec project were also brought into (by the Finnish Comspec partner) and developed further within the EU TIDE project ACCESS (TP1001). This project contributed substantially by emphasising the importance of employing state-of-the-art language technologies in the language environment of this kind of a modular AT and AAC environment [7]. Via a couple of intermediate projects, the heritage from the ACCESS project has ended up in the ITHACA framework [8], an open source environment for sustainable AAC products [9]. It includes a language sub-system that is both multilingual and multi-modal, as well as other components and tools with similarities to the Comspec-ComLink system, but depending on the Microsoft COM object technology.

Both the Comspec and ACCESS teams were in touch with, and/or partly inspired by related work by Patrick Demasco and Kathleen McCoy et al. at University of Delaware – A.I. DuPont Institute, and their COMPANSION system [10].

We can conclude that neither of these component based and modular systems has managed to attract enough further interest and resources to develop and sustain a critical mass of functionality and advantage to compete with the traditional models of dedicated and mainly proprietary AT and AAC software product development and marketing. Though visionary and in several ways ahead of their time – this was before any wide-spread awareness and popularity of free and open-source software development – these systems were also still limited to the perspective of dedicated AT and AAC development only.

Following the above early attempts to develop dedicated open technology based platforms for alternative communication, the WWAAC project (IST-2000-27518) [11] started in 2001. The baseline was that sending messages composed by symbols and pictures was difficult over the Internet, due to lack of standardised encoding schemes and common practises. WWAAC strove for the introduction of a common, open and vendor neutral multi-modal and multilingual language platform, based on Semantic Web technology standards, to overcome some of these difficulties [12]. A basic set of concepts was created, partly based on Princeton University WordNet® [13] and several pictures communication symbol sets and natural languages. The design and definition was published as the first version of the Concept Coding Framework (CCF) [14].

A project that took advantage of this development was the Nordic Symbered project. The aim was to develop an editor that effectively could help produce and adapt web documents based on XML to be presented as text with complementary support from graphic symbols. The CCF technology allowed encoded documents to be presented with symbols from any of the used symbol sets in combination with text from any of the offered natural languages – with possibilities to add new symbol libraries or languages to the database [15].

The results presented in this paper come from the recently finished EU project AEGIS (FP7, IP 224348) [16], where the CCF Framework has been used to develop components for symbol support in open source software for both mainstream and assistive technology, such as an extension for the LibreOffice/ OpenOffice Writer word processor, and AAC support in the Android mobile environment.

2. AAC and mainstream standard ICT – Some state of the art

ICT support for individuals who need AAC has typically been, and still is, provided in the form of dedicated software and/or devices. There will still be room for such, but there is a growing number of reasons for a move towards providing AAC functionality as part of standard mainstream ICT products:

- ICT products and services (such as smart-phones, tablets and mobile communication services) have rapidly become an integrated and often predominant part of everyday life, and are equipped with more general features (portability and usability combined with computing power for running many kinds of full-featured multi-media apps, larger displays of excellent quality etc.) making them suitable and desirable for AAC needs. It is natural that both users and their environment expect and request that a wide range of needs, including the specific ones in the area of AAC and language support, should be well accommodated as part of mainstream products.
- Some previously special functions of AT, such as text-to-speech synthesis, are gradually becoming mainstream technology.
- A solid infrastructure of flexible multi-modal and multilingual language representation technology is needed for good AAC support. There are potentially substantial advantages of developing such an infrastructure in the mainstream, both in terms of inclusion and participation, and in terms of cost, sustainability and availability – in particular where resources are scarce. More and more parents, (pre-school) teachers and others discover that access to synthetic speech and a range of graphical symbol representations is great for most learners at some stages in early literacy development and early new language learning. This is particularly the case in multi-cultural and multilingual environments.

Some of this integration is now rapidly happening as AAC software is being developed for and/or migrated to mainstream mobile devices in the form of mobile "apps" (though generally still with more limited or specific functionality than traditional dedicated AAC applications and devices). Though this is primarily only integration on the mobile platform and device levels, the consequences are still judged to be profound by experts in the field [17]. However, AAC and symbol support within standard activities and services is still a major step to be taken. The Widgit products "Point" and "Insite" [18] are examples of more integrated symbol support on the Web, but these are based on Widgit proprietary technology and lexical resources, and thus not open for a freer and wider range of applications by other developers and service providers.

In Chapters 3 and 4 below we present the, to our knowledge, so far unique efforts and novel outcomes related to the Concept Coding Framework developments in order to provide a foundation for multilingual and multi-modal AAC and general language support, as part of mainstream software services, and based on inclusive design principles, open technology standards, and free and open source software and lexical resources. But first, a discussion about some of the challenges that this work is and will be encountering, and which hopefully to a fair degree explain the necessarily long-term character of efforts in this area.

3. Challenges and preconditions for inclusive AAC support in mainstream ICT

A major challenge for the AAC field is the general complexity characterising the areas of communication and language in general, further added to by the growing socio-cultural diversity in modern European societies. The AAC area is living on top of this basic complexity, and adds to it by its own fragmentation in terms of a multitude of more or less well founded ideas and preferences about methodologies and tools, including choice of signs, graphic symbol systems, software tools etc. for different target user groups and needs. The reasonable response to these problems is to build AAC technology and resources on mainstream language technology efforts in order to bridge between languages, and add to that by levelling the ground for smoothest possible integration between different graphical symbol systems and sign language representations.

In other words, there is a need for an infrastructure for inclusive and integrated graphic symbol representation (as well as sign representation) of content and meaning in standard software environments. These infrastructural tools need to be based on open standards, be widely and freely available, and be multilingual and multi-modal so that more language representations may be added subsequently and in a distributed manner by local stake-holders. Components and tools need to be publicly available to make it easy to provide the multi-modal support in a widening range of different services, and for all kinds of potential users.

There are of course some more fundamental conditions and limitations that will create difficulties for AAC support in some environments and services. On the technical side, for example chat and messaging protocols are not supporting graphics. It is essential that a discussion is initiated around how such limitations may be overcome to allow future multi-modal communication also via these channels.

Other challenges are facing us in terms of attitudes, predominance of proprietary resources, and the still slow development of established open standards and resources for multi-modal and multilingual vocabulary interoperability.

If we look specifically into the AAC field and at the developers and maintainers of graphic symbol resources, these typically work in isolation from each other, and try to promote their respective systems in competition with, and exclusion of, others. This is unfortunately still typical for both commercial and non-commercial actors. Widgit has shown some positive openness by supporting the linking of the PCS and Blissymbol resources to the Widgit symbol library via its concept database. This integration is however restricted to parts of the proprietary Widgit software and services ecosystem. Unfortunately it's not only the commercial actors that show little interest in connecting up to other symbol vocabularies via common mainstream lexical resources. Also the non-commercial symbol system developers have by tradition been working within their separate boxes, and largely continue to do so. As long as the resources are available under some kind of free licensing they are at least available for integration by external and independent initiatives, such as the CCF. Here the non-commercial licensing restrictions for some systems cause unfortunate, unnecessary, confusing and counter-productive barriers between non-commercial and commercial actors co-operating around common free resources. Another problem with most graphic symbol systems, both proprietary and free, is their generally loose and ad hoc semantic definition and organisation. This makes the alignment between these resources and natural language lexical resources extra difficult. The main exception to this in the proprietary domain is the Widgit resources. Among the non-proprietary ones, Blissymbolics is the outstanding one in terms of clarity in definition and structure. Blissymbolics is therefore an important resource for aligning other free symbol resources between each other and further on towards well defined mainstream natural language lexical ontologies. This will however never be a trivial task and, as all language technology development, will require much work and a long-term perspective. Limited economic resources have been, and will continue to be, a major problem.

We are convinced that the CCF based developments (most recently within the AEGIS project) may serve as one platform and inspiration, among several others, for further European and international co-operation in this field. In addition to the European programs and the international standardisation bodies, it seems sensible to link such work to developing international frameworks like "Raising the Floor" and "GPII" (Global Public Inclusive Infrastructure) [19].

4. Recent Concept Coding Framework developments

One of the many objectives of the recently finished AEGIS project was to research, prototype and test freely available software services for inclusive graphical symbol support as part of mainstream environments to benefit people with communication, cognitive and multiple impairments. This has resulted in the development of a "graphical symbol server" based on the CCF technology.

The "CCF-SymbolServer" [20] can provide multilingual and multi-modal representation services locally on any of the major desktop platforms; Windows, MacOS X and, more tentatively, GNU/Linux. It is also available online to serve many kinds of web services and mobile devices. The server functionality is central for the planned future developments based on CCF. It communicates with the different CCF applications and different users, and with any new CCF application developed according to the CCF API. Users can set up their own local servers, which allows independence from the current development team, and true open-source.

Currently two symbol systems (Blissymbolics [21] and ARASAAC [22]) and four languages (English, Swedish, Spanish and Dutch) are supported, and the design is open to add more, graphic and sign libraries as well as language lexicons. Even though the current CCF technology has the great advantage of being open for such amendments, it should not be concealed that this is currently a major undertaking. Adding a new lexicon or symbol library to the CCF requires it to be linked up, either manually or semi-automatically, to the concepts in the CCF ontology database with substantial manual editing by persons with good linguistic competences. For good look-up functionality, the inflection patterns for the contained words in each language lexicon also need to be reasonably well covered. The complexity of the tasks is greatly dependent on how well structured and organised the new lexical resources are. This is undeniably a major hurdle for the wider adoption and use of the CCF. The strategy to overcome this is based on two elements: a) to gradually improve the interfaces and tools of the CCF to facilitate the work (see the first bullet in Chapter 4 below); b) to continue the efforts to gradually link and harmonise the CCF ontologies to existing and emerging open and free mainstream lexical and grammar resources with multilingual mappings, in order to facilitate reliable automatic amendment of new languages, and minimise the need for manual editing (see the final conclusions of Chapter 4).

A word in a given language can be sent to the CCF-SymbolServer. The server will look up possible meanings or concepts in its databases, and return these as Concept IDs with available possible symbol representations. If the concept ID is known, the server can directly return alternative representations in other supported languages and symbol systems. Figure 2 shows an overview of the CCF ecosystem.

The scope of the CCF database is built on the vocabularies of the symbol sets used – primarily from the more well defined Blissymbolics lexicon. This means there are about 5,000 concepts that all have one or more symbol representations. Linguistic data for concepts is extracted from the WordNet® [13] lexical ontology and stored in the database with links to the different representations. Given results from earlier phases, this process has been optimized and debugged to avoid as many unwanted connections as possible.

Moreover, WordNet only holds words from the four most common parts of speech; nouns, verbs, adjectives and adverbs. The symbol vocabularies, as well as the everyday language, have representations for additional function words like pronouns, prepositions, conjunctions and interjections. The symbol vocabularies also contain representations for more specific words not found in WordNet, typically connected to the field of disability and assistive technology. All those words have always been handled as a separate part of the database, but since they have no way of automatically being sorted correctly, efforts have been made to provide as smooth and accurate lookup and representation as possible.

Three applications interacting with the CCF-Symbol Server have been developed within AEGIS:

- CCF-SymbolWriter, an extension for symbol support in LibreOffice (or OpenOffice) Writer. It works like any extension to LibreOffice/OpenOffice. Once you install it, a menu is added to the word processor's interface. It allows you to select how you want it to work, graphic symbol system preferences, and a few other settings. Buttons are also added to the toolbar for quick access. When activated, the extension links up to the locally installed CCF-SymbolServer, which pops up in a window displaying the symbols the server finds for every word written into the document. While the symbols appear as graphics in the CCF-SymbolServer display, they may also be inserted into the document on top of the text word, but then as a private area Unicode font. Only in this way, the tool can make use of the standard Ruby annotation support of ODF (Open Document Format) to display the symbol straight above the word it correlates to (as only plain text is accepted in the annotations). When several symbol representations are available for a word, the user can choose which one to be inserted into the text by the server. The server automatically picks up the language setting of the current part of the document and adjusts the word-to-concept and symbol look-up accordingly, as long as the language

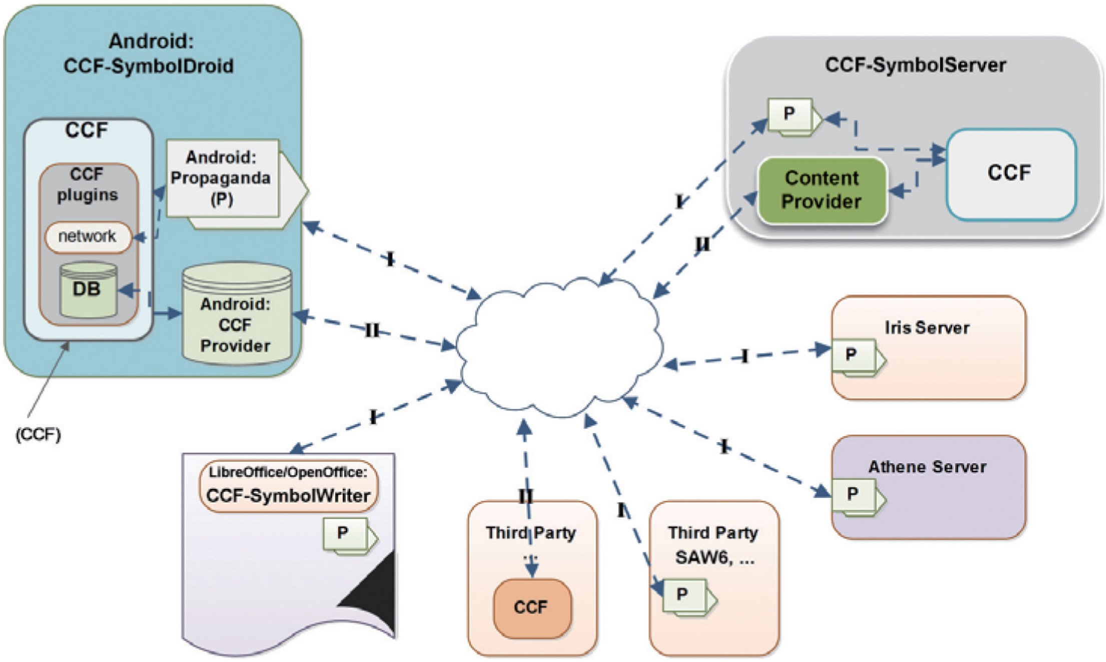


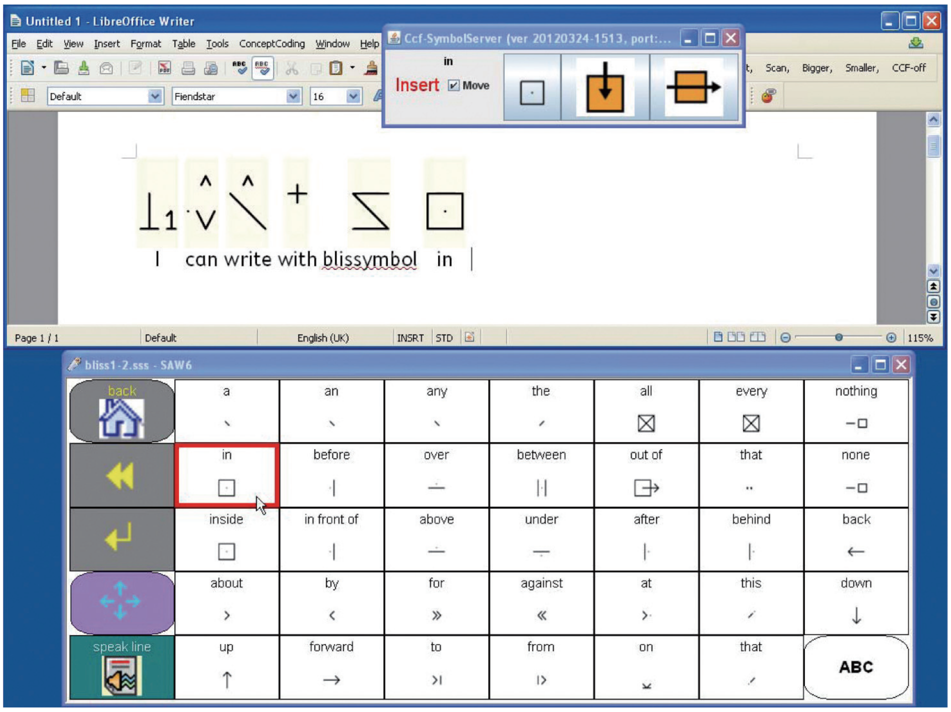
Fig. 2. CCF ecosystem overview with the CCF-SymbolServer providing its services to a range of existing and potential applications. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/THC-140407)

is among the ones supported (currently English, Swedish, and to some degree Spanish and Dutch). The CCF-SymbolWriter extension allows graphical symbol support for several kinds of needs within the mainstream Writer word processor environment:

a. The needs of struggling text users to support and confirm comprehension while writing and reading text, often in combination with text-to-speech (TTS) support
b. The needs of AAC symbol users to have access to full symbol representation (as far as possible) with symbols displayed on top of each word (or multi-word phrase)
c. The needs of helpers (parents, teachers, therapists etc.) to support the above needs by preparing and presenting documents with graphic symbol support

- A new CCF supported version of Special Access to Windows (SAW 6) [23], an advanced free and open-source on-screen-keyboard application that allows the creation of symbol selection charts for the control of any mainstream program on a Windows system. In this version of SAW, the program can link up to the CCF-SymbolServer, if installed and running on the system, and provide word-to-symbol look-up in design mode for creating on-screen symbol selection charts. Such concept coded charts may then be semi-automatically translated between the supported languages and symbol systems with the help of the CCF-SymbolServer.
  For a sample of SAW 6 together with CCF-Symbol Writer see Fig. 3.
- CCF-SymbolDroid, an AAC app for Android mobile devices, with the following functionality:
  a. Face-to-face AAC with graphic symbols, text and the system speech that is available on the device
  b. AAC for indirect communication via messaging in several forms, including text SMS, email and directly to other registered users via the CCF server
  c. Access to the full CCF vocabulary database via the online CCF-SymbolServer, including semi-automatic conversion of created set-ups between the supported languages and symbol representations
  d. Log in with user ID to access the online CCF-SymbolServer services – save, restore and share local set-ups via the server

This application is the least mature among the three presented here, and is still waiting for full-fledged support for user account management, tailoring of screen


Untitled 1 - LibreOffice Writer
File Edit View Insert Format Table Tools ConceptCoding Window Help
Ccf-SymbolServer (ver 20120324-1513, port:...
in
Insert ☑ Move
Scan, Bigger, Smaller, CCF-off
Default
Fiendstar
16
Page 1 / 1 Default English (UK) INSRT STD 115%
bliss1-2.sss - SAW6

I can write with blissymbol in

back	a	an	any	the	all	every	nothing
	in	before	over	between	out of	that	none
	inside	in front of	above	under	after	behind	back
	about	by	for	against	at	this	down
speak line	up	forward	to	from	on	that	ABC

Fig. 3. The CCF-SymbolWriter extension providing symbol representation within LibreOffice Writer with the help of a locally installed CCF-SymbolServer. Input is provided via a SAW 6 on-screen symbol selection set. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/THC-140407)

layouts for different screen resolutions, and some other features needed for wider marketing as a consumer product.

Figure 4 shows four sample screens from CCF-SymbolDroid.

5. Conclusions and further development

Given the feedback from end users, experts and developers within the AEGIS user pilot evaluations [24] and later experiences, the CCF services and applications are continuously developed. For example:

The support in the CCF-SymbolWriter extension is extended to handle multi word expressions and the possibility to interchange a word for a given symbol. This means that you could write ‘dog’ to get the symbol for dog in your document – and then replace ‘dog’ with the name of your own pet. The extension will also automatically identify the language based on your word processor’s settings, and ask the symbol server to find that word in the correct language in its databases. All possible meanings and concepts linked to that word are returned in form of the concept IDs and available symbol representations. The system handles all common inflection forms for all four languages, even if the support for English and Swedish is more extensive.

Since the end of the AEGIS project the development of the Concept Coding Framework, and the resources included in and built around it, are gradually developed in several different perspectives:

- The architecture of the framework, as well as the protocols, APIs and services of the CCF-SymbolServer and the CCF-SymbolDroid AAC app are currently subjects of substantial revisions and upgrades.
  - ∗ The need for federated services has been recognized, so that ontologies and lexicons can be maintained and made available from different cooperating servers, publicly available application servers as well as local servers. This is a move to encourage local, national and regional efforts in creating and providing more lexicons to the CCF community by maintaining them locally and automatically sharing them.
  - ∗ The work-flow of CCF is changing to support a stepwise refinement of ontologies and lexicons, that can then be published incrementally through the federated servers. This also enables

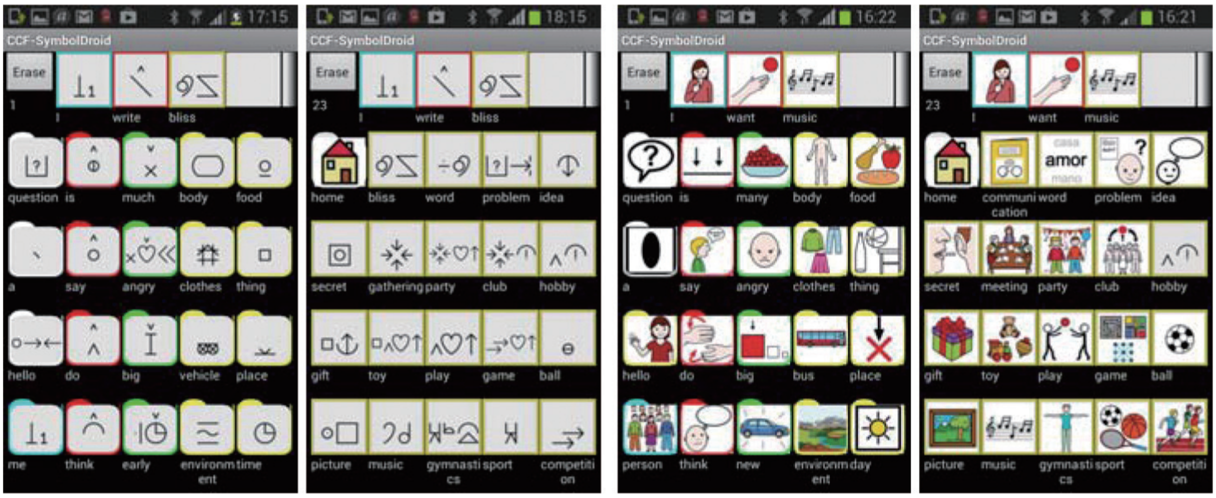


Fig. 4. The CCF-SymbolDroid AAC app for Android devices – four sample screens are shown; one home screen and one sub screen with Blissymbolics, and two corresponding screens with ARASAAC symbols. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/THC-140407)

the creation of local user ontologies and lexicons that users can publish.

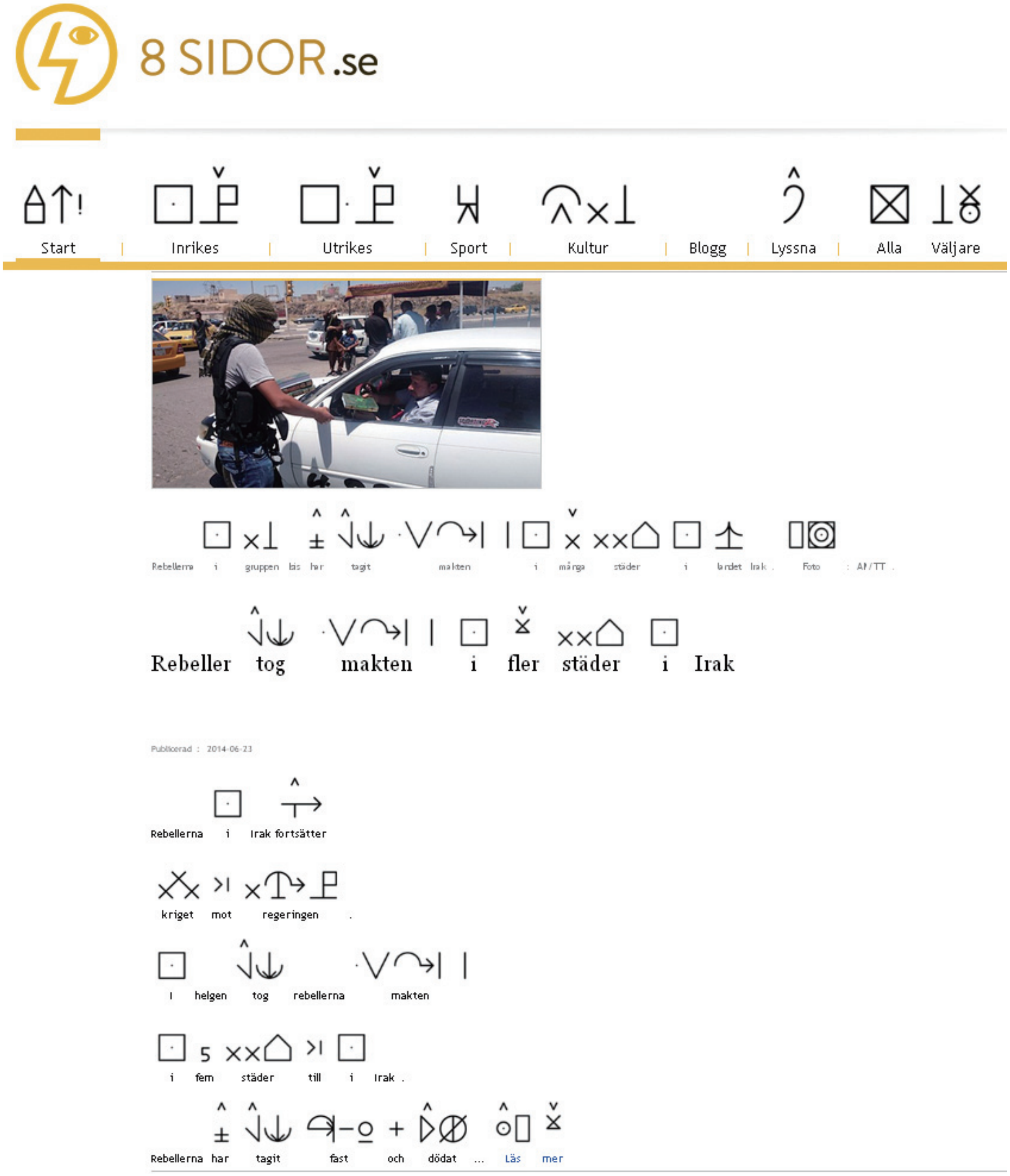
∗ The architecture of the framework as well as the building methods will be available through www.conceptcoding.org. This will include the provision of Maven archetypes [25] that can be used to create software projects that will automatically use CCF resources as libraries and enable the connection to the federated services. The archetypes that are currently under development are for Android applications and Java programs.

– So far, at least two applications interacting with the CCF-SymbolServer have been developed after AEGIS:
The first one is Nysnö – Symbol supported "easy to read" news texts on the web. It aims to develop a web based tool for providing Bliss and ARASAAC symbol support to easy-to-read daily news produced by the Swedish '8 SIDOR' publishing house. Underlying the final output exemplified in Fig. 5, are various language technology methods. After tokenization, the text is annotated with part-of-speech labels, using the HunPos tagger [26], trained on Swedish easy-to-read texts. The accuracy of the tool is highly dependent upon correct lemmatization. For this purpose, the information forwarded by the parser is used for lookup in specific wordlists containing around one million word forms and proper nouns. The entries in the CCF-SymbolServer database are supplied with codes, assigning them to either nouns, verbs, adjectives, adverbs, prepositions, pronouns, ambiguous or unknown part-of-speech. Based on this information, the tool automatically selects the symbol that matches a specific lemma. If a lemma has more than one lexeme associated to it, the most common one in easy-to-read context is chosen by means of a frequency check in the SweVoc core vocabulary list [27]. Following this automatic process, the author is presented with the text aligned with appropriate symbols for further revision and final publication.

The editing tool might also be useful for authors who want to supply information text, written in an easy-to-read format, with appropriate AAC symbols in order to enhance comprehension. After project closure, the tool is maintained and will be available under an open source license at the domain of www.conceptcoding.org.

The second one is related to a master thesis work at University of Gothenburg and Chalmers University of Technology by Marcus Tillander, under the supervision of Peter Ljunglöf. It is called "Simplipedia – Making Wikipedia Accessible" [24], and investigates the effectiveness in representing Simplipedia content with Blissymbols via the CCF-SymbolServer, and various ways to use Natural Language Processing tools to enhance the accuracy of symbol representation.

These new applications are still at a prototype stage and will be subject to a number of further refinements before regarded to be mature enough to be offered to



Rebellerna i gruppen Isis har tagit makten i många städer i landet Irak . Foto : AP/TT .

Rebeller tog makten i fler städer i Irak

Publicerad : 2014-06-23

Rebellerna i Irak fortsätter kriget mot regeringen .

I helgen tog rebellerna makten i fem städer till i Irak .

Rebellerna har tagit fast och dödat ... Läs mer

Fig. 5. A sample of Blissymbol representation of a news text from the Swedish plain text news service “8 Sidor”. Natural language processing techniques are gradually introduced to improve the completeness and accuracy of the symbol representation. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/THC-140407)

end users. Such needed improvements are adequate handling of proper names (of persons and geographical places etc.), and further improvements in word disambiguation and look-up of correct symbol representation in general. At this stage these applications provide further proof of concept regarding the wider usefulness of the CCF-SymbolServer functionality for a growing number of additional services, in particular web based ones. The outcome of the AEGIS project pilot user testing of the basic functionality of graphical symbol support for better comprehension of text was mainly positive [24]. Based on these findings, we may assume that this kind of multi-modal representation scaffolding of text interpretation will be appreciated by a range of users with different kinds of difficulties with text comprehension. Further user evaluation with mature versions of these services will be needed to fully confirm and qualify these findings.

– Furthermore, with support from VINNOVA (Swedish Governmental Agency for Innovation Systems), a project called LTLOD@SB, running during 2014 in collaboration between the Department of Swedish at the University of Gothenburg and DART, will make sure that the abundant lexical resources of Språkbanken (the Swedish Language Bank) [24], together with the CCF resources, are made openly available in the same Linked and Open Data formats.

The described efforts and results are all small but coherent steps in the direction of gradually establishing and integrating AAC vocabulary resources and functionality as a natural and integral part of mainstream free lexical and language technology resources, applied in software and services for a more open-ended range of use in the evolving information society.

Acknowledgements

The development of the CCF technology has been partly funded by the European Commission via the WWAAC and AEGIS projects. Additional support has been received from local Swedish funding and from the Nordic Council (SYMBERED).

References

[1] What is Inclusive Design, IDRC 2013: http://idrc.ocad.ca/index.php/about-the-idrc/49-resources/online-resources/articles-and-papers/443-whatisinclusivedesign.

[2] Vanderheiden GC, Universal Design. What It Is and What It Isn't, Trace R&D Center, University of Wisconsin-Madison, Rev. 5/6/1996. Available from: http://trace.wisc.edu/docs/whats_ud/whats_ud.htm.

[3] Concept Coding Framework (CCF): www.conceptcoding.org/.

[4] Svanæs D, Comspec. A Software Architecture for Users with Special Needs, InterAct'93 and Chi'93 conference companion on Human factors in computing systems, ACM, 1993.

[5] Comspec/ComLink: http://www.handicom.nl/en/projects/Comspec.html, and: www.javakomp.nu/infosidor/comlink-en.html.

[6] Lundälv M, Hekstra D, Stav E, Comspec, a Java Based Development Environment for Communication Aids. TIDE conference in Helsinki, 1998.

[7] Antona M, Stephanidis C, Kouroupetroglou G, Access to Lexical Knowledge in Modular Interpersonal Communication Aids, 1999. Volume 15, December 1999, AAC Augmentative and Alternative Communication. Available at: www.ics.forth.gr/files/publications/antona/1999/Antona_et_al.pdf.

[8] ITHACA: http://speech.di.uoa.gr/ithaca/framework.html.

[9] Pino A, Kouroupetroglou G, ITHACA: An Open Source Framework for Building Component-Based Augmentative and Alternative Communication Applications (2010), Available from: www.researchgate.net/publication/215205140_ITHACA_An_Open_Source_Framework_for_Building_Component-Based_Augmentative_and_Alternative_Communication_Applications/file/d912f509a7fe13e93b.pdf.

[10] McCoy KF, Demasco P, Some Applications of Natural Language Processing to the Field of Augmentative and Alternative Communication, 1995, Proceedings of the IJCAI-95 Workshop on Developing AI Applications for People with Disabilities, pp. 97-112. Available from: www.eecis.udel.edu/∼mccoy/publications/1995/McCoDema95.pdf.

[11] The World Wide Augmentative & Alternative Communication project (IST-2000-27518 WWAAC): www.wwaac.eu/.

[12] Judson A, Hine N, Lundalv M, Farre B, Empowering disabled users through the Semantic Web. In Proceedings of First International Conference on Web Information Systems and Technologies, Miami, USA (WEBIST) 2005, pp. 26-28.

[13] Princeton University WordNet, a lexical database for English, http://wordnet.princeton.edu/.

[14] Lundalv M, Judson A, Farre B, Nordberg L, de Waard L, Deliverable D10 – "Code of Practice" (2004), Available from: www.wwaac.eu/products/Docs/D10_v10%20CoP.pdf.

[15] Lundälv M, Mühlenbock K, Farre B, Brännström A. SYMBERED – a Symbol-Concept Editing Tool. LREC – Language Resources and Evaluation Conference, Genua, 2006, 1476-81.

[16] AEGIS – Accessibility Everywhere: Groundwork, Infrastructure, Standards, – co-financed by the European Commission's 7th FP, 2008–2012. www.aegis-project.eu/.

[17] White Paper on Mobile Devices and Communication Apps from AAC-RERC 2011, Available from: http://aac-rerc.psu.edu/index.php/pages/show/id/46.

[18] Widgit – www.widgit.com/.

[19] Raising the Floor – http://raisingthefloor.org/, and GPII – http://gpii.net/.

[20] Lundälv M, Derbring S, Nordberg L, Brännström A, Farre B, Graphic Symbol Support in Open/LibreOffice Shaping Up – Graphic Symbol Server and Inline Symbol Font Display Based on the CCF, 2: nd International AEGIS Conference and Final Workshop 2011, pp. 282–293. Available from: www.slideshare.net/aegisproject/conference-proceedings-2011-aegis-international-workshop-and-conference.

[21] Blissymbolics Communication International – www.blissymbolics.org/.

[22] ARASAAC – www.catedu.es/arasaac/.

[23] Special Access to Windows, Available from: www.oatsoft.org/Software/SpecialAccessToWindows.

[24] AEGIS project Consolidated Evaluation Report, Deliverable 1.5.2, pp. 86-114, available from: www.aegis-project.eu/images/Deliverables/AEGIS_D1.5.2_final_pilot_results.pdf.

[25] Archetype Maven Project; Introduction to Archetypes, Available from: https://maven.apache.org/guides/introduction/introduction-to-archetypes.html.

[26] Halácsy P, Kornai A, Csaba C, HunPos – an open source trigram tagger. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion

Volume Proceedings of the Demo and Poster Sessions. Association for Computational Linguistics, Prague, Czech Republic 2007, pp. 209–212.

[27] Heimann Mühlenbock K, Johansson Kokkinakis S, SweVoc – A Swedish vocabulary resource for CALL. Proceedings of the SLTC 2012 workshop on NLP for CALL, 28–34. Lund: Linköping University Electronic Press.

[28] Tillander M, Simplipedia-Making Wikipedia accessible, master essay, University of Gothenburg and Chalmers University of Technology 2014. Available at: http://hdl.handle.net/2077/35356.

[29] Borin L, Forsberg M, Olsson L-J, Uppström J, The open lexical infrastructure of Språkbanken. Proceedings of LREC 2012. Istanbul: ELRA. 3598–3602.