

# Modelling Volo, an Augmentative and Alternative Communication application

Antonina Dattolo

SASWEB Lab, Dept. of Mathematics and Computer Science  
 Università di Udine, Gorizia, Italy  
 email: antonina.dattolo@uniud.it

Flaminia L. Luccio

Dept. of Environmental Sciences, Informatics and Statistics  
 Università Ca' Foscari Venezia, Venezia, Italy  
 email: luccio@unive.it

**Abstract**—In this paper, we present a formal representation of an Augmentative and Alternative Communication (AAC) application, called Volo, for users with Autism Spectrum Disorders (ASD). We discuss existing AAC applications and present a formal visualisation model of Volo based on *zz*-structures, hyper-orthogonal, non-hierarchical structures for storing, linking and manipulating data. To the best of our knowledge, this is the first work that tries to give a formal model to AAC techniques. We finally present the prototype of this new Volo application.

**Keywords**—*Zz-structures; mobile app; Augmentative and Alternative Communication; Autism Spectrum Disorders.*

## I. INTRODUCTION

Language delays and impairments are one of the common characteristics of children with Autism Spectrum Disorders (ASD) [1]. However, these children often show good visual abilities [2], thus in the last years standard speech therapies have been combined with Augmentative and Alternative Communication (AAC) techniques: they are powerful methods that combine different visual components in order to create syntactically and semantically correct sentences.

AAC techniques can be of different nature: they can either be based on unaided systems, i.e., rely on the use of body actions such, e.g., gestures, sign language, etc., or on aided systems which are based either on lite technology such as Picture Exchange Communication System (PECS) books, letter boards, or on high technology such as dedicated devices containing images and sounds [3] (see Figure 1).

### A. History of AAC systems

The first examples of Augmentative and Alternative Communication (AAC) systems go back only to 1960, when letter and picture boards, and typewriters were first introduced as a way of communicating by very few nonspeaking individuals [4]. Issues that deeply impacted the development of AAC devices were: in 1975, the approval by the Congress of the Public Law 94-142 (Education of All Handicapped Children Act), which assured a free appropriate public education to all children with disabilities; the development and diffusion of microcomputer technology, given that prior to that these devices were handmade electrical systems, or non-portable computer systems [5].

The first example of a commercially available dedicated AAC device is the Canon Communicator from Canon Inc., a small device with a keyboard and a strip printer. After that researchers concentrated on portable voice synthesiser leading to the production, in 1978, of the first commercial AAC device with speech synthesis, Handivoice, from the Federal Screw

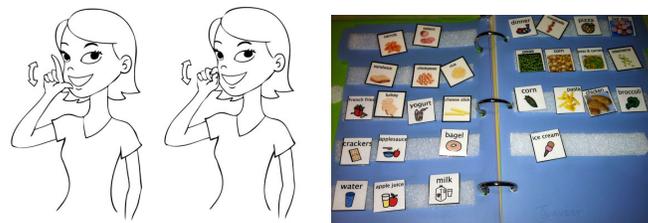


Figure 1. An example of sign language (left) and of a PECS book (right).

Works and Phonic Ear. Nowadays, given the big advances in microcomputer technology, AAC devices have become more and more sophisticated and adaptable for all the wide variety of users [6].

In this paper, we are interested in formal models and visualising techniques for high technology AAC devices, so we will concentrate on them. In particular, we will consider dedicated applications for mobile systems, like smartphones and tablets.

The study of current literature highlighted that, independently from the type of app which is chosen, there are some basic principles followed by all of them: the main feature is a picture-based approach, given that it is proved to be effective on children that demonstrate the comprehension of cause and effect actions. A training approach that promotes symbolic communication and is based on lite technology is the PECS, developed in 1985 by Lori Frost and Andy Bondy. This approach works well for those children that demonstrate emerging intentional communication as it promotes interactions among users [7]. An example of a PECS book, which is a velcro communication board, is shown in Figure 1-right. Each child with ASD is different and has different abilities, thus the use of this book, and in general any learning target, may differ.

There are different training levels which are introduced to the child by a specialised therapist. In Phase 1, the simplest one, the child initiates a request by removing a picture from the board (the unique one there), and by handing it to an adult in exchange for a desired item or action. When the child shows a good ability at this level, it moves to Phase 2, in which it has first to select the picture, and then physically bring it to a partner which is not nearby. In Phase 3, the child has to discriminate one picture among many other pictures (as in in Figure 1-right); in Phase 4, it has to build a sentence using a sequence of pictures, e.g., a picture representing the concept

“I want”, and another picture representing the desired object; in Phase 5, it has to reply to questions such as “What do you want?”, with a sequence of pictures. Finally, in Phase 6, it has to respond to questions and comment on items, e.g., using pictures which define concepts such as “I see”, “I hear”, “I feel”, etc. [8].

The AAC apps that we will consider in the following are all based on this picture-based approach.

### B. Features of AAC devices and scanning methodologies

High technology AAC systems have many different features: they can range from dedicated devices to specialised apps from different platforms; their size and weight may hardly affect their portability and accessibility, which are very important features for children with ASD which are typically ambulatory. The methods for storing and retrieving sounds and messages may differ, e.g., offering real or synthesised words as a spoken output. Other features are the access methods which determine how the user will select symbols on the AAC device, such as pointing with fingers, using eye gaze, etc.. Finally, the methods we are interested to analyse and consider in this paper are related to the visualisation, scanning and choice of images.

The way users may scan and select images differ from a device to another:

- *Automatic scanning* is the basic standard technique. The device presents different symbol choices while the user waits and, when the desired picture is reached, it is chosen by the user by a pressure. This can be done using 1-step scanning such as *linear scanning* in which pictures are scanned linearly, and *circular scanning* in which are scanned by following a circular pattern. However, both types of scanning work well when the number of presented pictures is limited [2]. The process can be speed-up using the *row-column scanning* (or sometimes the *column-row scanning*) which consists of 2-steps scanning where a row of pictures is presented at once, it is scanned and when the desired picture is reached the user has to select it.
- *Group-item scanning* derives from the row-column scanning and assumes the refinement in subcategories of pictures. Groups, i.e., classes of symbols are presented at once; the user chooses a group and then the pictures inside this group are presented. These pictures can be subgroups or final items. This scanning technique assumes a three- to four-steps scanning process and is thus the most difficult for users to manage, but at the same time it allows their the use of a wider range of images.
- *Directed scanning* has an opposite approach: the user keeps the switch pressed and s/he releases it when the item is reached. This allows the user to move his/her hand on the device without accidentally activating symbols; on the other hand, it requires the user to focus on the pictures for a long time.
- *Step scanning* assumes that the user presses the switch one step at a time to move the to the next symbol, and when the desired symbol is reached, s/he stops.

Comparing these selection methods, it turns out that automatic scanning requires less actions by the user (only a click

when the item is reached), but the user has to patiently wait for the picture. The other methods require more physical ability of the user but increase his/her cognitive process and provide a direct feedback for an action. For all methods, note that the communication rate is still very low, at most 43 words per minute [9]. For this reason, many existing AAC systems use picture-based accelerating techniques. For example, the sentence “Hi, how are you”, could be coded as a picture of a waving hand + a finger pointing, i.e., a sequence of symbols that code in a compact way a more complicated sentence [2].

The paper is organised as follows: in Section II, we present the characteristic of many existing apps for ASD children, and in particular we concentrate on the visualisation techniques. In Section III, we describe the Volo formal model, based on the the zz-structures, while in Section IV we present the prototype. Note that, to the best of our knowledge, this is the first work that tries to give a formal model to AAC techniques. We conclude in Section V.

## II. RELATED WORK

In this section, we describe some of the commercial applications based on AAC techniques, which are used by children with ASD, and we will compare them from a visualisation point of view.

The commercially available apps are many and range from free to very expensive tools. There are applications for iOS (as [10]-[11]), for Android (as [10][12][11]), or for special dedicated devices (as [13][14]). We concentrate here on tools that can be used by children with strong communication delays and allow their the selection of images. The differences among these apps are first obviously in the graphics: some apps use sketched images, other real pictures, most of them provide the user with an initial set of pictures and allow the import of new images from a personal computer, a camera, etc. (as [10][15][12]-[11]).

Another feature is the possibility of adding sounds, which can be synthetic or natural (as in [10][16][17][12][11]), or can also be recorded (as in [16][15][12]). Most of the applications are also multilingual. The Niki-talk application [15] has also social features, i.e., it allows the user to type a message and eventually to tweet it.

Some apps allow the creation of calendars: the daily routine might be organised in sequence of actions which describe the activities of the day in a fixed temporal order as in Figure 2-right where the user visualises a sequence of actions to follow, in order to get dressed (as in [11][18]).

Although all these apps might have some different features, the visualisation characteristics seem to be common: in particular, all the tools we have analysed assume dynamic displays, i.e., the set of pictures which the user is able to visualise may dynamically change after a selection. Moreover, pictures are displayed in a grid form, i.e., in rows and columns, whose size can be, in most cases, customised.

In Niki-talk [15], the user may select between two categories of actions (“I want”, “I am”), a keyboard icon that links a page where the user may type and hear words, and a paint brush icon that opens another page where the user can draw and save his/her drawings. These two icons (keyboard and paint brush) appear in each internal view together with items of the



Figure 2. An example of the Niki-talk app [15] on the left, and of the Pics Aloud calendar [18] on the right.

same categories - see Figure 2-left where the user selected the action “I want”.

In Proloquo2Go - Symbol-based AAC [17], images of different categories may appear together (as subject, verbs, etc.), and are differentiated by the border color. In simple views, only images of the same category are displayed inside the grid. The app displays in the top the sequence of selected images. Images may be rearranged by a user customisation, for example, inserting images in a given position, etc..

TTalk AAC [19] shows categories and simple images together, while Let’s talk [16] displays many categories at the beginning and it allows for two distinct uses: the user can be alone, or helped by an external person that suggests images. Pics Aloud [18] has a slightly different approach: it displays many real pictures of different categories, and, when the user selects the image, the system repeats loudly a whole sentence. For example, if the image is a swimming pool, when the user selects it the app says something like “I want to go to the swimming pool”, i.e., it produces a sentence for the user.

Voice4u [11] may display a set of predefined images in an alphabetical order, otherwise by clicking on a special button, categories appear and the user may browse inside them.

Alexicon AAC [10] and TalpToTalk [12] are Web applications; they displays many categories and objects in a grid form. Alexicon allows sequential and row/column scanning, and also auditory preview scanning. TalpToTalk allows the extension of an image word to a whole sentence. GoTalk [13] and QuickTalker [14] are two examples of apps that run on a dedicated device. They both display images in fixed 2 × 2, or 9 × 9, etc. grid, containing objects, categories and objects.

### III. A FORMAL MODEL FOR PECS-BASED APPS

In this section, we introduce the basic definition of a formal and general model for describing generic AAC-based applications, that can be used to formally describe all the applications mentioned in the previous Section II. In this paper we concentrate on the formalization of our model, the extension to all the other applications is an ongoing work.

The model is based on zz-structures which are a graph-centric system of conventions for data and computing, able to simply connect different elements through contextual, semantic connections [20][21]. Readers interested in a deep discussion on zz-structures, its formal description and practical applitudes can refer to [22]-[23].

Using these zz-structures is possible to simply describe knowledge representation, visualisation and the scanning process.

*Definition:* An AAC-based app is a zz-structure,  $AAC-A = (MG, SL, sl)$  where

- $MG = (V, E, f)$  is a multigraph (a graph where pairs of vertices might have multiple edges connecting them) composed of:
  - a set of *vertices*  $V$ ;
  - a set of *edges*  $E$ ;
  - a surjective function  $f : E \rightarrow \{\{u, v\} \mid u, v \in V, u \neq v\}$ ;
- $SL$  is a set of semantic labels;
- $sl : E \rightarrow SL$  is an assignment of semantic labels to edges of the multigraph.

$AAC-A$  is subject to the following constraint:  $\forall x \in V, \forall k = 1, 2, \dots, |SL|, deg_k(x) = 0, 1, 2$  where  $deg_k(x)$  indicates the *degree* of the vertex  $x$  (that is, the number of edges incident to it) labeled by the semantic label  $k$ .

In other words, an AAC-based app is an edge-labeled (equivalently edge-coloured) multigraph where the vertices are either singletons, or are connected by the labels in linear paths, or cycles.

Each vertex of a zz-structure is called a *zz-cell* and each edge is a *zz-link*. We can extract from the zz-structure a sub-graph, composed by all the connected components (linear paths or cycles) of a particular semantic label. Each connected component is called *rank*, while the sub-graph of ranks of the same label is called *dimension*. Each label is associated to a specific semantic context; for this reason, ranks and dimensions provide a semantic interpretation of the zz-structure.

An example of  $AAC-A$  is shown in Figure 3, where normal, thick, dashed, dotted and double lines represent five different dimensions. (Note that, the images used here in Figure 3, and later for the presentation of the zz-structure and the prototype, have been downloaded from many different sites of the Web.) The first row (the dimension called *category* -



Figure 3. An example of a simple zz-structure.

thick line), collects the main categories of items (like food, sports, games, dvds), while the first zz-cells (sports and food) of the two vertical dimensions (identified respectively by normal and double lines) generalise the set of images contained in that dimension. They are called *maincells*, and their role in the dimension is different from the role of the other cells:

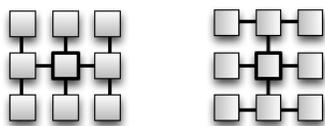


Figure 4. A schematic representation of a 3x3 H-view and I-view.

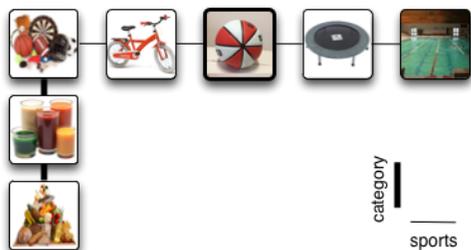


Figure 5. An example of 5x5 H-view.

they symbolise the dimension, and represent their neighbouring cells. Analogously, also the second and third cells of the food dimension (sweets and first courses) are maincells: they identify two sub-categories of food.

An AAC-A may be visualised in different ways: in this paper, we use the H-view [24], a rectangular view (based on two dimensions, called primary and secondary) where the focus *zz*-cell is placed at the center of the view on a Cartesian plane (the cell with bold border), while the primary dimension occupies the horizontal central path, and, for each *zz*-cell, the *zz*-cells, connected to it, and belonging to the secondary dimension, occupy the vertical paths. A schematic example of a 3x3 H-view is shown in Figure 4-right; the name H-view comes from the fact that the columns remind the vertical bars in the capital letter H. In Figure 4-left is schematised a 3x3 I-view. Obviously these views contain exactly 3x3 (in general  $n \times m$ ) *zz*-cells, only if each cell has the necessary neighbours in both primary and secondary dimensions. The formal definition of these views is provided in [22].

Figure 5 shows an example of 5x5 H-view, where the focus is the *zz*-cell representing a basketball, highlighted by a bold border; the horizontal dimension is *sports*, and the vertical is *category*. Potentially, in an H-view the same cell could appear in different positions as it may represent the intersection of different semantic contexts (for example, a basketball may be also in the category *games*).

#### IV. THE MOBILE PROTOTYPE

In this section, we describe the prototype of a new app, called Volo we are willing to develop (Volo comes from latin, and its translation is “I want”). The app proposes a customisable and adaptive (the description of the adaptive aspects of the Volo app lies outside the goals of this paper and is not proposed here) user interface: a user that has very limited capacity to search through images will have an initial configuration containing only pictures of different, frequently used items. An example of an initial view is shown in Figure 6.

The interface shows five items, that slowly scroll up horizontally. Each item looks bigger when it comes to the



Figure 6. Volo app: a simple, initial view: the user selected a tart.

center, while the others are smaller and opacified. When the user touches an item, it comes to occupy the central position (if it was not already there), the border assume a brilliant color, and the item is added to the history of the user and visualised at the top of the view.

The interface is very simple: in addition to the items, it contains only the home button (top-left corner), and the sequence of selected images. The selection of an item may be associated with a reading voice (customizable in different languages). There is also the option of setting the app so that the reading voice pronounces only the first syllable. This is useful when the child has started to repeat words but is not able to produce it by its own, so the app provides a small hint. To improve user performance the required pictures have to be efficiently accessed, thus objects are viewed following a statistic of most frequently searched items.

The first screenshot (Figure 6) goes back to Phases 1 to 3 of PECS. More capable users will have a more complex initial configurations. They will start by choosing a subject, then a verb, then an object (Phase 4 of PECS), and so on. Thus, we assume that the app is customizable depending on the different users, and depending on the different learning pathways they are able to follow (some decisions are taken by therapist/parents which have access to the app). Images are also customisable with imported pictures; the app however already provides the user with real daily life pictures.

We also assume that pictures for users in Phases at least 4 will be categorised: for example, to select a particular food the user will first choose the image of food, and then will visualise a list of different types of food. Depending on the user this can be refined by defined subcategories, for example, food-sweets-tart (see the example in Figure 7). While defining categories we are using images which include inside sub-elements of the category itself, for example, to represent food we use an image that contains vegetables, fruit, bread, etc. (see Figure 7 left). This choice derives from the fact that many people with ASD disorders represent concepts through sets of related images; for example, to represent the concept of the color green they may visualise different objects of that color, like grass, leaves, etc. [25].

No matter in which view the user is working, we assume that our app is based on the *zz*-structure approach. The user is able to visualise few objects in one dimension, i.e., in a row or



Figure 7. An example of division in subcategories: food → sweets → tart.



Figure 8. Volo app: the category food is proposed to the user, and he/she may select it.

in a column at the center of the screen. The user moves from a picture to another one using step scanning, i.e., it presses the switch to move to the next symbol, and stops when it reaches the desired picture, or by swiping the fingers to scroll through images and then pointing the picture.

At this point a new dimension (of the AAC-A) appears in the other direction still at the center of the screen, visualising an H-view (if the previous dimension was horizontal now it appears vertically, or vice versa), and with a subset of items. If the searched item or category is not viewed, the user may move using a circular scanning.

We now illustrate an example for a user that is able to work up to Phase 4 of PECS. The final request is of a tart. The app visualises a page with a row containing the images of sports, drinks, food, games, dvds (see Figure 8), which are general categories.

The user presses the switch or swipes to scroll, and stops when the image of the food is at the center. At this point a new vertical dimension appears and shows the images of sub-categories: sweets and first courses (Figure 9).

The user presses the switch and stops when sweets appears at the center, and a new horizontal dimension appears with images of specific sweets (Figure 10).

Finally, the user presses the switch and stops when the image of a tart appears. In the meantime, in the top the user visualises the sequence of images (zz-cells) that have been selected food-sweets-tart.

What is really new in our model and app is the use of the single zz-structure dimension. All the existing apps work using a grid view, i.e., the screen is filled with rows and columns of items. Using only two dimensions (an horizontal and a vertical

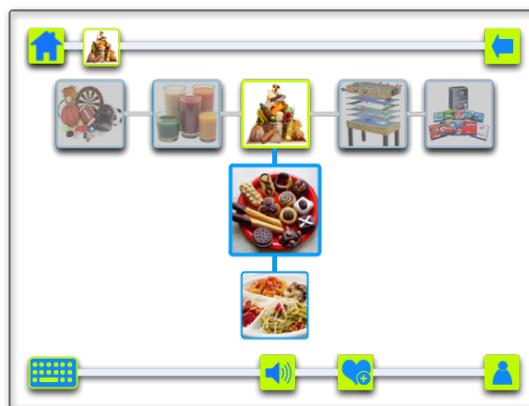


Figure 9. Volo app: the user selected the category “food”, and the app proposes its sub-categories (sweets, main courses).

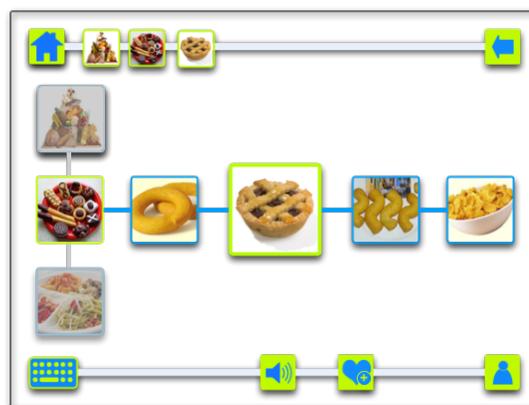


Figure 10. Volo app: the user selected the sub-category “sweets” and the app proposes a set of sweets.

line), limits the number of visible objects thus simplifying the understanding of the user. Moreover, we assume that while the user is navigating on an image (i.e., the image is at the center), it may visualise the other dimension (as a preview). For example, in the previous example depicted in Figure 9 the user navigates on the view, presses and arrives at the category “drink” (thus visualising a dimension with images of drinks), and while realising it is not correct it moves to the food image. This is very important when the child is still not able to categorise because it slowly starts to associate the category image to the sub-items.

We assume that if the user chooses a wrong image there are different solutions: (a) for not very experienced users, we assume the possibility of pressing a complete reset button (i.e., go back to the initial view), represented by the home button (on the top-left corner of Figures 6, 8, 9, 10); (b) more experienced users may also select a back button (represented by the arrow button on the top-right corner in Figures 8, 9, 10), that goes back to the previous view, both for the horizontal and vertical lines, and for the set of selected items in the top.

For children in school age that are starting to read, the app may also provide writings (in the chosen language) in capital letters, so to simplify the reading (if the desired language is not present, the writing is omitted). An example is shown in

Figure 8 where, below the image, the user can read the related English word.

Another feature for expert users is a “like button” (represented by the heart button in Figures 8, 9, 10) with which a preferred item is selected by the user itself. We also assume that some options are customisable by clicking on a special icon (represented by the user button, on the bottom-right corner in Figures 8, 9, 10): the colours, the size of the grid, the initial view, the border of the images, the possibility of adding or removing a writing, adding the sound of a complete word, of a syllable, etc.

In this paper, we chose to illustrate the setting for users with very strong language disabilities. However, our prototype allows to create more complicated views in which the sequence of images that can be chosen are, for example, subject → verb → adjective → object, where the subject can be chosen from a set of images containing the user, parents, brothers, sisters, etc. (assuming the app has been customised with imported images, otherwise the user finds sketched images), a set of images for verbs, adjective and objects.

An additional feature of our app is a reward button, a hand with a thumb up, which is not shown in the figures, but is possible to add it from the setting page. It is well known that positive rewards help the user fixing a concept, as a matter of fact positive reinforcement is the most widely applied principle of behaviour analysis (a commonly used therapy for ASD disorders). Thus, we add the possibility, for the person that is interfacing with the user (or by the user itself when it increases its ability to use the app), to reward the user not only by answering his/her request, but also by pressing this reward button. The app then executes something funny, for example, the app says “well done!” while opening a page where there are stars falling down.

## V. CONCLUSION AND FUTURE WORK

In this paper, we presented a formal model and the prototype of the new application Volo we are willing to develop in the coming future.

Volo has some nice features which, to the best of our knowledge, seem to be lacking in other applications. In fact, Volo has been developed having in mind users with very strong language disabilities, and thus its navigation structure is simple and usable, although the underlying model is general and scalable.

As a future work, from a formal point of view, we are preparing a formal and more detailed description for proving that our formal model includes, and can completely describe all the different existing AAC-based applications. From an implementation point of view, we are willing to engineer the app, starting from the actual prototype. Moreover, we are planning to include an expert therapist in the team in order to evaluate both the proposed model and, what is more crucial, to experimentally evaluate the impact on a group of users with ASD. Finally, it would be interesting to propose the real app to teachers at school, and to compare it with other existing approaches also from a usability and accessibility point of view.

## ACKNOWLEDGMENT

This work has been partially supported by the PRIN 2010 Project *Security Horizons*.

## REFERENCES

- [1] American Psychiatric Association, Ed., *The Diagnostic and Statistical Manual of Mental Disorders: DSM 5*. BookpointUS, 2013.
- [2] S. Dodd, *Understanding Autism*. Elsevier Australia, 2005.
- [3] C. Weitz, M. Dexter, and J. Moore, AAC and children with developmental disabilities., S. Glennen and D. C. DeCoste, Eds. San Diego: Singular Publishing Group, 1997, vol. Chapter 11.
- [4] H. R. Goldberg and J. Fenton, *Aphonic communication for those with cerebral palsy: Guide for the development and use of a communication board*. New York: United Cerebral Palsy of New York State, 1960.
- [5] G. C. Vanderheiden and K. Grilley, *Nonuocal communication techniques and aids for the severely physically handicapped*. Baltimore, MD: University Park Press, 1976.
- [6] G. C. Vanderheiden, “A journey through early augmentative communication and computer access,” *Journal of Rehab R&D*, vol. 39, no. 6, May 2009, pp. 39–53.
- [7] A. Bondy and L. Frost, “Mands across the water: A report on the application of the picture-exchange communication system in Peru,” *The Behavior Analyst*, vol. 16, 1993, pp. 123–128.
- [8] “Pyramid educational consultants, inc.” 2014, <http://www.pecs-canada.com/> [accessed: 2015-01-08].
- [9] A. Szeto, E. Allen, and L. M., “Comparison of speed and accuracy for selected electronic communication devices and input methods.” *Augmentative and Alternative Communication*, vol. 9, 1993, pp. 229–242.
- [10] “Alexicom AAC,” 2014, <http://www.alexicomtech.com/> [accessed: 2015-01-08].
- [11] “Voice4u. Helping your communication,” 2014, <http://voice4uac.com/> [accessed: 2015-01-08].
- [12] “TalpToTalk,” 2014, <http://www.taptotalk.com/> [accessed: 2015-01-08].
- [13] “GoTalk,” 2014, <http://www.attainmentcompany.com/> [accessed: 2015-01-08].
- [14] “Quicktalker family of communication devices,” 2014, <http://www.ablenetinc.com/Assistive-Technology/Communication/QuickTalker> [accessed: 2015-01-08].
- [15] “Niki-Talk,” 2014, <https://play.google.com/store/apps/details?id=it.alessandrolarocca.nikitalk> [accessed: 2015-01-08].
- [16] “Let’s talk by Toriken Media Information Lab,” 2014, [http://ne-kite.com/index\\_english.html](http://ne-kite.com/index_english.html) [accessed: 2015-01-08].
- [17] “Proloquo2go,” 2014, <https://itunes.apple.com/au/app/proloquo2go/id308368164?mt=8&ign-mpt=u0%3D4> [accessed: 2015-01-08].
- [18] A. Burrington, “Pics aloud,” 2014, <https://itunes.apple.com/us/app/pics-aloud/id402333607?mt=8> [accessed: 2015-01-08].
- [19] “TTalk AAC by T-Box srl,” 2014, <https://itunes.apple.com/it/app/talk-aac/id537958879?mt=8> [accessed: 2015-01-08].
- [20] T. H. Nelson, “Welcome to ZigZag,” 1999, <http://xanadu.com/zigzag/tutorial/ZZwelcome.html> [accessed: 2015-01-08].
- [21] —, “A cosmology for a different computer universe: data model mechanism, virtual machine and visualization infrastructure,” *Journal of Digital Information: Special Issue on Future Visions of Common-Use Hypertext*, vol. 5, no. 1, 2004, p. 298.
- [22] A. Dattolo and F. L. Luccio, “A formal description of zigzag structures,” in *Workshop on New Forms of Xanalogical Storage and Function - HT 2009*. Turin, Italy, June 29-July 1 2009, pp. 7–11.
- [23] A. Moore, J. Goulding, T. Brailsford, and H. Ashman, “Practical aptitudes: Case studies of applications,” in *Proceedings of the 15th ACM Conference on Hypertext and Hypermedia (HT’04)*. Santa Cruz, California, USA, 9-13 August 2004, pp. 143–152.
- [24] M. J. McGuffin, “A graph-theoretic introduction to Ted Nelson’s Zstructures,” January 2004, <http://http://www.dgp.utoronto.ca/~mjmccguff/research/zigzag/>.
- [25] T. Grandin, “How does visual thinking work in the mind of a person with autism? A personal account,” *Philosophical Transactions of the Royal Society*, vol. 364, no. 1522, May 2009, pp. 1437–1442.