A kindergarten experiment in linguistic e-learning

Andrea Valente[a] and Emanuela Marchetti[b]

[a]Dept. of Computer Science AUE – Aalborg University in Esbjerg;
[b]Center for Medieval Studies – University of Southern Denmark (Dk)

av@cs.aue.auc.dk; e.manum@gmail.com

Abstract

As part of the Blasq project, we are developing a set of linguistic games to be used in kindergartens. The first of these games is Crazipes, that we are currently testing in a Danish kindergarten, with the support of the local teachers. Here we discuss the architecture of the game, its potentials as a linguistic e-learning tool, together with the design and methodology adopted for the study. Some early results are also discussed.

Edizioni Erickson – Trento
www.erickson.it

Je-LKS Journal of e-Learning and Knowledge Society — *Vol. 2, n. 1, marzo 2006 (pp. 131-142)*

1. Introduction

E-learning is becoming an important technology in the field of computer-assisted education, and the focus of many studies in that area is university or high-school. However within the community there is a growing interest in adapting e-learning to primary school subjects.

One of the biggest challenges for this scaling down is that many educators oppose to the idea of exposing children too much and too early to computers or to engage them in long solitary, indoor activities. And these fears can be even stronger when targeting kindergartens.

E-learning has also been applied to linguistics, again for older students, with some degree of success (Chang and Schallert 2005). We believe, together with Tokuhama and Espinosa (2001), that there can be a great benefit in adopting linguistic e-learning tools with kindergarten children: in fact early exposition to different languages seems to be crucial for language learning.

Our aim is then to develop a new multi-language learning system, consisting of many interconnected play-and-learn games, capable of running on various devices and also to be played with more classic, passive toys.

In this paper we present the first of these tools, Crazipes, and describe the approach we are following to conduct a field experiment in a kindergarten close to our department.

2. Crazy recipes

Crazipes, short for *crazy recipes*, is a multi-modal[1] game based on pictorial translation, computer graphics and animation, voice interfaces, and then tangibles, such as toys made of paper, so that kids can play both with and without a computer. Crazipes is also part of a larger project named Bla^{sq} (read it «bla square» or «bla-bla»), whose aim is to create a multi-lingual environment, a virtual place where children are challenged to engage in different games, partially based on passive paper-tangibles and partially media-enhanced. We address children, between 3 to 5 years old, to exploit the known natural tendencies of preschooler kids to familiarize with languages (Tokuhama and Espinosa 2001), in this way we hope that our tool also help them while challenging difficult tasks, such as learning to read and write. Interaction and animation are the core tools we are using to make elements and structure of various languages visible, at the same time making our game more attractive to children, and so providing an enjoyable and fun playing experience.

Several disciplines are involved in the design and implementation of Crazipes: linguistic, pedagogy, cognitive science, and anthropology from a humanistic point

[1] In the sense advocated by Papert [Papert].

of view. And from the technical point of view: human computer interaction and software engineering, Java programming and 2D graphics, and finally animation, for the artistic aspects of the project.

The Crazipes prototype is developed in Java and it comprises two interdependent packages: the first modelling the recipes, their ingredients and all the linguistic aspects of the application and the other related to the definition and deployment of a kindergarten-level, multi-modal user interface.

2.1 The recipe domain

The central classes of the recipe package are Recipe and Language, both abstract (see figure 1). They fix a contract for their respective subclasses: a Recipe object represent the description of a single recipe, together with its ingredients and preparation steps. Given an instance of class Language, a Recipe object can generate a text (a string) with its own human-readable description, expressed in that specific language. So, in our idea recipes act as data, while languages provide rendering contexts.

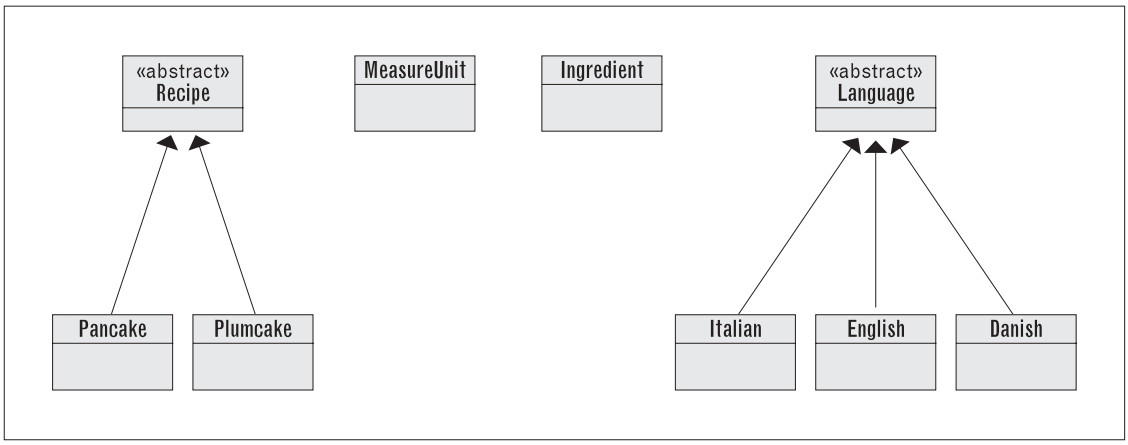

Figure 1 Architecture of the recipe package.

The Language class provides the core linguistic constructs, a kind of *cooking jargon API* that can be used when defining recipes.[2] For example we have actions like: *doWork, doWarm* and *doPourInto.*

Subclasses of Recipe, like Pancake or Plum cake, implement concrete recipes and they could be turned into singletons in later versions of the package: in fact with the present architecture, all instances of class Pancake end up representing multiple, immutable copies of the same recipe.

[2] The 2 recipes have been adapter from [FDU] and [FCB].

On the other hand, subclasses of Language implement concrete languages; each of them contains a dictionary, providing a mapping from the abstract cooking API to readable sentences.

As an example, the Plum cake class defines its preparation-phase more or less like this (on the left):

l.doWarm(BUTTER);
l.doWork();
l.doAddTo(SUGAR , BUTTER);
l.doMix();
l.doAddTo(EGG);
l.doMix();
l.doAddTo(FLOUR , JAM);
l.doMix();
l.doPourInto(MOULD);
l.doCookWithTempForUntil(OVEN ,"180","30",
GOLDEN)
	warm butter,
	work it,
	add sugar to butter,
	mix,
	add egg,
	mix ,
	add flour to jam,
	mix ,
	pour it into mould,
	cook it with oven at 180 degrees
	for 30 minutes until golden

where the variable l represents a reference to the specific language (here it is English), in which the pancake recipe is rendered. On the right column, the English rendering of the recipe is visible. Here the rendering is very crude, but in later version we will improve the English class (as well as the other languages), so that *cook _something_ with oven* could be translated more properly as *bake _something_*.

Moreover, in many languages concordances are important,[3] so future versions of Crazipes should be able to handle them as well.

One of the main advantages of this simple architecture is its flexibility: recipes and languages are fairly decoupled, so whenever a new language is added, all the previous recipes can be rendered in it without modifications. This is due to the fact that we do not need to translate the same recipe from any language to any other, but instead we first rewrite the recipe using a formal set of terms (ingredients and the cooking API); then we translate/render that formalized structure[4] directly into each one of the different languages available in the system, thus reducing the complexity of this mapping from n^2 to linear. In this way each new language class is in itself quite simple to write, because it needs only a simple dictionary from cooking commands and ingredients to natural-language words.

Note also that the recipe-rendering metaphor central to our architecture can easily scale beyond textual descriptions, to provide voice and animation renderings.

[3] For example in Italian adjectives can be singular or plural and masculine or feminine, depending on the word they refer to.

[4] Many domains can be formalized in similar ways, as for example in [Propp], where folk-tales structures are analyzed.

2.2 Kindergarten-friendly user interface

The main concern for us, while developing the graphical user interface for Crazipes, was that our users are children 3 to 5. This means they cannot read even in their native language, so we decided to couple text with icons and speech (as in the simple GUI of the painter program Tux Paint, by Bill Kendrick and others). To solve this problem we created a *talking button,* that can be clicked to perform an action, like a normal GUI button, but that will also show a textual hint of what the button represents, and at the same time, reads the hint aloud. Figure 2 shows what happens when the mouse cursor stands for more than a few milliseconds on top of the button representing *camel milk*, in our program: a hint appears and a wave file containing the words «camel milk» in English is played.

Figure 2 Talking Button.

We decided to use pre-recorded wave files, instead of generating speech in real-time, mostly because it is very difficult to find a speech synthesis library that can cope with multiple natural languages. However we think that having such a library would greatly simplify the work needed to incorporate new languages into our game (as well as reduce the size of the application), and it would also be interesting to see the reaction of 3 to 5 year-olds to synthetic voices.

In the current version of Crazipes, the UI works in three languages: Italian, English, and Danish; the players find our talking buttons almost everywhere in the GUI, from the initial login dialog (see figure 3a), to the game itself (see figure 3 b and c).

Logging in without writing or reading can be a problem in itself: we need children to log-in in order to keep the data about each game session separated, but they cannot write a password or read instructions about how to create an account in the game. Again our solution was to use visual information, so after asking permission to the parents, we took pictures of all children in the kindergarten. In this way a preschooler can log-in simply by clicking on his or her face. Although

this is not as secure as a standard login system with passwords, we believe it is a reasonable approximation. Each picture in the login dialog is implemented as a talking button, with the voice of one of the teachers reading the names aloud (see figure 3a, where pictures have been blurred to ensure the privacy of the kids); as a matter of fact, hearing the names of their classmates seems to be quite funny, at least for some of the children, so they used some time to carefully inspect the login dialog, before entering the game.

Once passed the login stage, the player enters the game, and she will be asked to select 4 ingredients from 12 available, 4 in each of the 3 languages: the final result will in any case be a pancake.

An important principle in a multilingual environment is to be consistent in identification, so we assigned each language to a particular color and a national flag (figure 3, frame b and c). Ingredients are arranged into 3 rows, one for each language, and into columns, indicating the kind of ingredient: as for example the ingredient *milk*, can be associated with *cow milk*, *camel milk* or *sheep milk*.

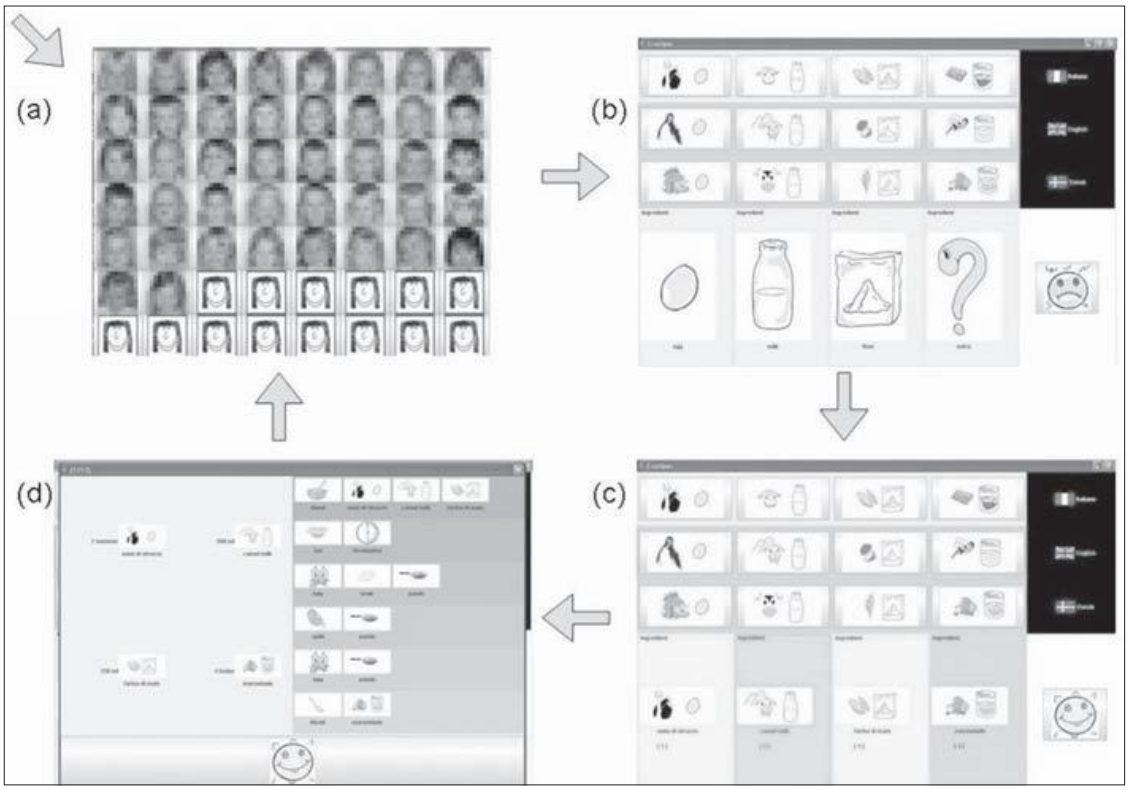

Figure 3 Crazipes walkthru.

Only when all 4 ingredients have been selected, the player can see her recipe, or she can change her selection. Once done, she can press the smiling-sun button and get a description of her personal (and usually a little crazy) version of the recipe.

Recipes in Crazipes are a mixture of reality and fantasy: they are realistic since the recipe does exist, and the instructions can be used to reproduce it, but the ingredients involved can be quite absurd, as camel milk or penguin egg.

Ingredients and preparation of the recipe are all represented by hand made drawings; the different phases of preparation are divided into lines by a color gradient, to suggest that the visual recipe should be *read* from the top-left.

While the player is looking at the final recipe, she can also collect some paper icons (that we provide together with the PC and the game), and a printed copy of the template of the recipe: using these tangibles she will be able color and glue together her visual instructions for the pancakes and later to cook her pancakes at home, or *pretend play* about cooking with her friends.

At this stage of development the final rendering of a recipe is quite static and probably not so interesting for children, so we decided to add a few *special effects* to make the game more attractive. For example, an animation appears on the screen after having clicked twice (or more times) the same ingredient, or after clicking a specific pattern of buttons. This is actually quite a standard trick, used in many games to encourage players to explore the game more in depth. In Crazipes there are 3 animations, hidden in the system and to be discovered by the kids, they include: the face of a cow that will pop-up and cover all of the screen, then *moo* in three different ways, a penguin that blows and freezes the screen (see figure 4), and a strong vibration, like an earthquake, that breaks the screen. Before each one of these distractions, the game blocks and the user has to wait for the animation to stop, in order to finish her recipe.

2.3 Distribution and deployment

As all tools in the BlaSq project, the current version of Crazipes is freely available at the BlaSq website: http://cs.aue.aau.dk/~av/BlaSq/crazipes-demo.zip, distributed

Figure 4 The penguin animation: rising (on the left) and ice-blowing.

under GNU GPL. Later this year, using the feedback from the early experiments, we will improve the game and provide it with a free editor, so that other researchers or institutions can download, customize and deploy their version of Crazipes.

In line with the main ideas behind e-learning, we are also discussing a development for the Crazipes game so to enable children to design their own recipes, in a mix of human languages, then exchange them via internet; since in our tool ingredients and instructions always have a visual representation (and can be *acoustically rendered* in one or more languages at the same time) recipes created in this way will be pretty simple to *read* even for children speaking different languages or too young to be able to read. These recipes will also be easy to share between kindergartens in different countries, possibly by means of on-line repositories (see Najjar et al., 2003, and more in general http://en.wikipedia.org/wiki/Learning_object and http://ltsc.ieee.org/wg12). In this scenario, teachers should be in charge of the administration of such repositories.

As for the current kindergarten experimentation, new versions of the game will be realized and deployed all year long, in order to test the children's interest and performances, especially from the linguistic point of view.

In its current implementation, our game runs constantly, in kiosk-mode, on a stand alone-machine, and is accessible (locally) at any moment. But we discovered very soon in our experimentation, that children do not always close the program properly when done: some of them simply leave the session going. For this we had to implement a time-out function that closes the game and restarts the login dialog, if no event is generated for more than 10 minutes.

3. An experiment in a Danish kindergarten

Our experiment is designed on the basis of linguistic and psychological linguistic studies with children, but since we do not have a strong experience in pedagogy and psychology we are going to use any possible help and information from the teachers, who (at least in our case) are pedagogues, and appear to have a solid, practical experience and knowledge of their children.

The goal of our research is to verify if a videogame, involving words in multiple foreign languages, can interest preschool children and make them curious about other languages, remember the words used in the game so to prepare the development of a meta-linguistic conscience. We are also interested in what kind of interaction can emerge between them during a game session (will they play alone or cooperating with others?), if gender or age differences are visible or affect their interest in language(s). In this first experimentation our program will be tested with about forty children that, following teachers' suggestion, are around five years old. All of them have already practical experience with PCs and videogames, as there is also a PC in the kindergarten with same old games. They are all native

Danish speakers, except two brothers from an English-Danish mixed family and one native Arabic speaker, who can understand Danish even if he still was shy when answering back.

A PC with our program will be left at the kindergarten, running in a sort of kiosk-mode, so that kids could use it any time they want, also without supervision. Then we will work on two kinds of different data. First we have data collected by the program itself during its usage by kids, their action are in fact automatically registered in a simple CSV-format file (compatible with both Excel and OpenOffice spreadsheets). Each line of file records a game session: initial date and time of the session, name and age of the player, every step and choice made, the final combination of ingredients, and the duration of the session, expressed in seconds. These data are collected without any effort from our part, and can be structured in diagrams, showing behavioral patterns during game sessions.

It will be later possible to group player behaviors with respect to their age and/or gender, see if clusters of similar behaviors emerge. We are also interested in performing a longitudinal evaluation of the data representing the same player in time: we could see for example if the session becomes shorter after the first times (i.e. the player grows faster in the use of the visual interface) or if there is a tendency to experiment more before deciding, thus resulting in *slower* sessions.

Finally we will collect data by direct observation, paying regular visits to the kindergarten, around twice a week, and we will keep a diary, following an ethnographic method often adopted in this kind of research (Unsworth 2000). At the beginning, we will instruct the children about how to use the program, then we will observe them; we wish they get used to our presence so that they will behave in a natural way, therefore we will not go there every day, to not disturb too much their regular activities. We can always ask the teachers to provide us with information about children's behavior during our absence.

In this first part of the experiment we will leave the computer at the kindergarten for one month (September 2005). The version of Crazipes will be a prototype and we intend to improve it following the feedback from children and teachers. New versions will be gradually realized and brought to the kindergarten during the school year 2005-2006; in this period our strategy will be the same outlined above: go there circa twice a week and register relevant data both automatically on the PC and manually on a diary.

It was not easy for us to design this experiment, because there is not a unified methodology in linguistic studies (Bennet-Kastor 1988). The use of audio-video registrations became usual since the seventies and the eighties, but it will probably create us even more problems, for example a recording apparatus can be expensive, require maintenance, and its presence could bother teachers and children.

We know that while keeping a diary it is difficult to select what to write, and notes can be influenced by expectations (as explained in Bennet-Kastor 1988 and

Unsworth 2000), therefore we will base our analysis also on the data collected automatically by the program and most of all on the information provided by the teachers.

For what we can see, in Danish society, multi-lingual skills are regarded as very important, and the vast majority of people can use more than one language, apart from Danish, in everyday life (e.g. reading texts and watching television); the openness to foreign languages is also supported by the traditional mass-media, in fact television programs and movies are subtitled in Danish, not dubbed. However, in kindergarten the main concern is to make children able to use Danish, especially if their families came from abroad; children usually start learning other languages only from primary school and cartoons or kids movies are usually translated into Danish.

In such a rich linguistic environment, we found that both the kindergarten personnel and families are reacting enthusiastically to our initiative to test these prototypes with their children. As long as privacy is concerned, we discussed the issue with the teachers, who asked for families' permission, before we started the experiment.

3.1 Early observations

Our experiment has just begun, but we have already some interesting information about children behavior and so elements to improve our program.

First of all we have to say that boys were more confident in using a PC and willing to try the game by themselves. Girls instead were quite shy and did not show much interest about the game, we had to ask teacher's help to call them to try it, and then some of them had problems with the mouse.

Children, especially boys, had a cooperative way of playing, they gave each other suggestions about how to structure the recipe and how to make a particular animation appear. We have already seen this kind of behavior the first time we entered the kindergarten, one boy was playing with a videogame and other two were participating giving him advices.

For what we could see until now, they do not really seem to realize the absurdity of ingredients like penguin eggs, that was supposed to be the funny part, but they like to see animals and hear voices from the program. They liked the animations too, once they discovered one they tried to make it appear again and again, suggesting the others to choose the *magic button*, but it seems that none of the kids, until now, found the earthquake effect, that requires a complex sequence of actions.

In general, their reaction to tangibles is more uniform: both boys and girls got the idea about taking the printed tangibles home, and they asked for more copies, at some point or another of the experimentation.

Unfortunately, kids do not see a goal in the game and they do not seem really interested in the presence of foreign words. However, we could see that two

brothers, native Danish and English speakers, liked to mix English and Danish ingredients and they repeated the words aloud: they probably like the feeling that words in both languages sound familiar to them.

So even now, a few weeks into the first run of the experiment, we can appreciate some trends in children's behavior: they play together, there is no interest in linguistic but in colors and animations, they seem not to have realized the peculiarity of the absurd ingredients involved.

This probably means that the game needs to have a clearer purpose: a goal to reach or some kind of puzzle to solve, and anyway it should be more strongly connected to the linguistic aspects.

4. Conclusions and future works

In this paper we discussed the design and early results of a linguistic experiment that we are starting in a Danish kindergarten. Using a program with playful interface, we are attempting to let children explore many languages at the same time, and our working hypothesis is that a system like that should not be uniquely based on computer and active media, but the playing experience should continue in the *real world* (i.e. in the playground, at the kindergarten or at home).

However, the right mix of passive paper-tangibles and media-enhanced contents is quite difficult to realize, therefore we are analyzing data about the behavioral patterns of children playing our game, and we will have to confront our findings with the teacher's experience.

Some aspects of the game are interesting for the children (like the fact that the PC *speaks*, and the pop-up animations), but other seem to go unnoticed (e.g. the presence of different languages in the same user interface).

We are aware that there is plenty to improve in our game, and especially we need to provide a more precise goal for the children to feel challenged. In future versions we would like to have a kind of *crescendo*: the final part of the game should animate the recipe step-by-step, in a funny and loud way, naming ingredients in the different languages, so that children can later in-act their own recipes using the given paper tangibles.

As soon as the prototypes reach a reasonable stage of development, we will put them online, for others to use and test, and we hope in this way to collect data about patterns of behavior in different countries.

Acknowledgements

We would like to thank the anonymous referees for their comments and useful suggestions; they really helped improving this paper.

BIBLIOGRAPHY

Bennet-Kastor T. (1988). *Analyzing children's language-Methods and Theories.* Blackwell, Oxford New York.

Chang Y.F. and Schallert D.L. (2005). The design for a collaborative system of English as foreign language composition writing of senior high schools students in Taiwan. *Proc. of the 5th ICALT, Kaohsiung Taiwan, 5-8 July 2005.* Published by IEEE, Los Alamitos California (USA).

Ojakangas B.A. (1964). *The Finnish cookbook.* Crown Publishers New York.

Various authors. *FoodDownUnder - Food and Culinary Categories* [WWW document] URL: < http://www.fooddownunder.com/ > accessed on 20th December 2005.

J. Najjar, E. Duval, S. Ternier and F. Neven (2003). Towards interoperable learning object repositories: the Ariadne experience. *Proceedings of the IADIS International Conference WWW/Internet 2003* (Isaias, P. and Karmakar, N., eds.), vol 1, pp. 219-226, 2003.

Papert S. (1993). *The children's machine. Rethinking School in the Age of the Computer.* Basic Books, New York.

Propp V. (1968). *Morphology of the folktale.* University of Texas Press.

Tokuhama-Espinosa T. (2001). *Raising Multilingual Children.* Westport, Greenwood Publishing.

Bill Kendrick and others. *Tux Paint - Open Source Drawing Software for Children.* [WWW document] URL: < http://www.newbreedsoftware.com/tuxpaint > accessed on 20th December 2005.

Unsworth L. (2000). *Researching language in schools and communities functional linguistic perspective.* London, Cassel.