An Analysis of Current Language Learning Software

Laura Vawter and Alke Martens
Computer Science Department, Rostock University, Albert-Einstein-Str. 22, Rostock, Germany

Keywords: Language Learning, Computer Assisted Language Learning.

Abstract: To start the development of an optimal digital learning environment for language learning we have started with the first question: What current language learning software is available to users? To answer this question an analysis of current computer Assisted Language Learning (CALL) software is necessary. This paper expounds on the systematic analysis of 69 current language software. Based on this structured analysis, we have developed a first framework of requirements for an digital language learning environment. For this purpose, control, user input, software feedback and theoretical educational frameworks were analysed within this investigation. The analysis demonstrates a lack of constructivist frameworks and a higher prevalence of behaviourist educational frameworks in current language software.

1 INTRODUCTION

In the context of computer based training (CBT) or computer aided instruction (CAI), a plethora of systems are available. On one hand, there are many forms of software types available, e.g. simple web-based training systems (WBT), adaptive systems, intelligent tutoring systems, etc. From the perspective of computer science, these system types are all based on different programming ideas, models or even paradigms. To develop these systems, the computer scientists make use of a variety of techniques, stemming from artificial intelligence, agent technology and/or software patterns. These developments are often influenced by modern research trends in computer science. On the other hand, educational psychology and insights from instructional design (aka. didactics) have influenced the development of computer-based learning software. Additionally, the way instructional design and educational psychology influence a computer-based learning software development, is often dependent on the amount of time available for the development of the software. For example, to develop a software which is based on behaviouristic learning paradigm is comparably easy, whereas developing a cognitively demanding or a constructivist environment is demanding for programmers and content developers.

Computer Assisted Language Learning (CALL) has been used in and outside of language classrooms since the 1960's. Current research has sought to explore how historical uses of CALL impact language learning as well as how CALL has been influenced by educational theories and research language acquisition (Hegelheimer and Chapelle, 2000, Hulstijn, 2000, Chapelle, 1998, and Doughty 1987; as referenced in Bordonaro, 2003).

The first step in our research seeks to go beyond the historical uses of CALL and investigate what CALL software is today. In order to investigate what CALL software is available, a retrospective analysis must be made regarding types of language learning systems as well as forms of CALL software.

1.1 Systems

From a computer science perspective, there are many systems used in CALL software. Historical systems like CBT or CAI (Martens, 2004), are not as prevalent as Interactive Learning Environments (ILE) or Intelligent Tutoring Systems (ITS) in modern CALL.

The first ITS systems were developed by learning psychologists in the late 1970s. However, the core of ITS has remained the same. Martens expands on how ITS incorporates expert knowledge, pedagogical knowledge, learner, and user interface models. Furthermore, she expounds that these systems are often a combination of Artificial Intelligence and Computer Aided Instruction. The ITS system acts as a tutor that reacts to the users "progress and needs, his level of knowledge, and his performance in the actual

Vawter, L. and Martens, A.
An Analysis of Current Language Learning Software.
DOI: 10.5220/0007727101750183
In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019)*, pages 175-183
ISBN: 978-989-758-367-4

context" (Martens, 2003). Thus, an ITS can, depending on the underlying learner model, make adaptation to the learner's progress, decisions and/or prior knowledge and expertise. Adaptation can take place regarding the content, the navigation elements, and the presentation style. This overwhelming amount of flexibility comes with comparably high development costs and time Consequently, ITS is nice to have, but often not realized by companies due to cost.

Chou and Hillman, et. al. describe ILE as involving the interactions of "learner-content, learner-learning, learner-instructor, and learner interface" within a software (as referenced in Wang, et. al., 2009). However, system adaptation is not always present in ILE software. In general ILE is a niche development, which realizes only some aspects of ITS.

ILE and ITS are, in most cases, are individual learner focused. However, in the context of language learning where communication is the focus, we find the following software types:

- Computer Mediated Communication (CMC)
- Computer Supported Collaborative Learning (CSCL)
- Network Based Language Training (NBLT)

CMC is described by Stockwell and Tanaka-Ellis as "distance environments" or "blended learning environments" (Stockwell and Tanaka-Ellis, 2012). In these settings the software provides the connection between the user, instructor, content and assessments. One common representation of this format is language schools, university departments or institutions offering classes or seminars online. Blake describes CMC as utilizing "social computing tools" like forums, blogs, emails, Skype, or instant messenger programs. In most forms, thus, we find a combination between computer-based settings (or CMC) and the presence of teachers and learners (e.g. classroom) (Blake, 2011).

Scott, C. and Engal describe CSCL as a "cultural constructivist approach" (Scott, C. and Engal, 1992). Chapelle describes it as a software or platform through which users interact and collaborate with each other or an instance where users in the same room or through local area network connections interact and collaborate (Chapelle, 2001).

NBLT is characterized as taking place on a "local area network" or "wide area network". (Chapelle 2001). Additionally, Chapelle categorizes pedagogical activities included in NBLTs as Microworlds, Grammar Checkers, Pronunciation Feedback Systems, ITS, Concordances Programs and Word Processors (Chapelle, 2001).

1.2 Educational Framework

CALL software systems also differ in how Behaviourist, Cognitivist and Constructivist educational theories influence them.

Behaviourists educational elements can be identified with Skinner's research into "drill and practice integrated learning systems". The tasks within these systems are scaffolded in a hierarchical structure based on complexity and managed according to a "stimulus/response feedback loop" (as referenced in Niederhauser and Stoddart, 2001). The feedback the user receives in these systems is immediate and based the "correctness" of their input. Egenfeldt-Nielsen further explains the reliance of these systems on rewards (Egenfeldt-Nielsen, 2006).

The cognitivist educational psychology is evident in these systems by the prevalence of differing tools, activities or formats that promote higher order thinking (Stockwell, 2012). Interestingly, these manifestations often mimic Bloom's Taxonomy and require users to predict, produce and reflect on their language input.

In CALL systems influenced by constructivist learning theory, users manipulate, discover and explore content within the system (Hogle, 1996, Niederhauser and Stoddart, 2001). They may incorporate micro-worlds (Egenfeldt-Nielsen, 2006) or the support of peer-peer interaction (Becta, 2001, referenced in Mitchell and Saville-Smith, 2004).

With all of this insight into the complexity of language software systems the question remains: what CALL software systems are available nowadays? Furthermore, what elements are found in these software systems?

2 BACKGROUND

An initial investigation of linguistic and computer science research specified what form our evaluation must take. The following is a short explanation of the research behind our questions.

2.1 Research Questions

Martens describes an adaptive system as flexible to any changes in the learner's development or in the condition of the user's input into the system (Martens, 2004). Similarly, Brusilovsky expresses that an adaptive system modifies its feedback to the user's needs, problem solving strategies, and understanding demonstrated in the learning process. This includes the altering of feedback in relation to a user's repeat mistakes- like the provision of hints or clues to address these user-specific problems. (Brusilovsky 1998). Thus, we sought to answer the questions: *how does the software respond to user's input? What feedback does it give?*

M. Hron defines adaptability and flexibility in software systems as the ability of learners to have unrestricted access to content within a system. (referenced in Martens, 2014). From the perception of language learning, flexibility is related to the ability of users or learners to negotiation the meaning of content (Long, 1996). Adaptation and flexibility is also seen by Pennington and Stevens, as providing users with the ability to choose "the mode and format to demonstrate their knowledge" as well as the freedom to learn according to their interests" (Pennington and Stevens, 1992). Thus, we sought to answer the question: *what level of control do users have in the software?*

Pennington and Stevens ascertain that by providing learners with diverse modes of input and requiring user production of content rather than imitation and memorization, the learner's acquisition of language may be positively affected (Pennington and Stevens, 1992). Similarly, Hulsyojn and Laufer found a corrilation between user involvment in a task and language acquisition (as referenced in DeHaan, 2005). Thus, we sought to answer the questions: *what is the user input within the software? What forms does it take?*

Niederhauser and Stoddart describe how behaviourist styled drill and practice systems are used in language education (Niederhauser and Stoddart, 2001). Also, Reinders and Darasawang describe the benefits of using cognitivist "metacognitive strategies" in language learning (referenced in Stockwell 2012). Rieber describes the positive effect of exploration and discovery in constructivist systems on learning (Rieber, 2005). Thus, with this criteria, we sought to answer the question: *What type of educational psychology frameworks are evident in the systems?*

Much research has found CMC systems to be beneficial to language learners (Zhao, 2003). Additionally, Chapelle expands on the use of collaborative learning with CSCL in language classroom and the effects on language acquisition. Similarly, Chapelle expounds on the importance of identifying NBLT aspects to have a more accurate inference of the effects on language learners (Chapelle, 2001). Thus, we sought to answer the following question: *What systems types are evident?*

3 ANALYSIS

This paper details our research into current CALL software system including our evaluation process of these system. A total of 69 systems were investigated.

3.1 Feedback

First, we sought to differentiate between systems that had an awareness of user's input and systems that did not. An example of a system that didn't have system awareness would be a software program that offered videos, clips or text in regard to a topic and didn't track if the user watched or read the content. Similar systems received the label "No User Awareness". In contrast, as system that tracked what videos or clips the user watched and those they didn't would be considered in this investigation as having user awareness and would have received the label of "User Awareness". In the investigation, we further sought to analyse the forms of user awareness that were present in the systems. We further categorized the awareness of a system as either "immediate" or "accumulated" and further specified forms from there

In regard to "immediate" awareness the system identified whether the user's response was correct or incorrect it received the label of "Answer". If the system in addition to identifying the correctness of the input gave an "explanation", "hint" or "translation" it received such labels (see Figure 2). A label of "Audio" was received if the system gave a sound following user input- this was not necessarily in response to the correctness or incorrectness of the input. In contrast, if, for example, the system played an audio clip without the user clicking on a button it received the label of "No User Awareness: Audio".

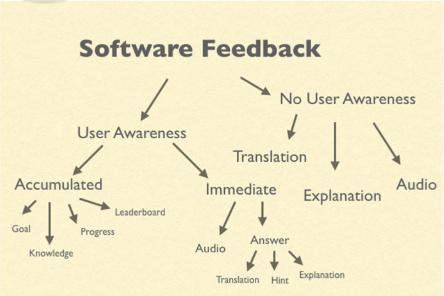

Figure 1: Software Feedback.

We intended to further specify immediate feedback within the system by measuring what form the software's feedback took. If the system gave a short explanation of why user input was correct or incorrect it received the label "explanation". This is in contrast to the system giving an explanation of a grammar rule without user input. This would receive the label of “No User Awareness: Explanation”. Likewise, if a system gives a translation of a word or sentences without user input the system receives the label “No User Awareness: Translation”. Additionally, if the system indicated which part of the user’s input was incorrect the system received the label of “Hint”.

As pertaining to “accumulated” system feedback the label “leaderboard” was used in systems that had leaderboards, and the label “progress” was used for systems that tracked users long-term use. “Progress” may be in the format of a system tracking how often the user logged-in, how many exercises, units, levels or lessons the user completed, how many vocabulary words the user learned, or how many experience points they earned from completing a certain type or amount of exercises. Similarly, the label of “goal” was given to systems that set goals for the user in relation to their progress (i.e. learn this many words today). Finally, if the system kept record of the accuracy level of the user’s input it received the label of “Knowledge”.

3.2 Adaptation

The second analysis we conducted was in regard to system adaptation to the user. There are four types of choice investigated in each software Choice, Repetition, Feedback and Difficulty (see Figure 2).

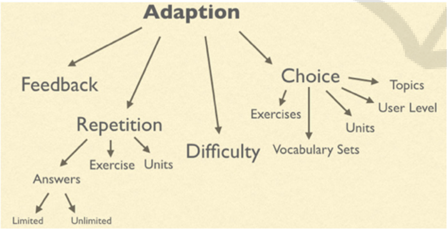

Figure 2: Adaptation in CALL.

If the system allowed the user to choose which exercises to complete (this could involve skipping some activities or exercises or choosing the order in which they do each activity), it received the label of “Choice: Exercises” If the system allowed the user to skip or change units, levels, stages, topics, or models etc. it was given the label “Choice: Units”. If the user could choose what language level (i.e beginner, intermediate etc) the software received the label “Choice: User Level”. Finally, if the user could select the vocabulary present in the units and exercises the software received the label of “Choice: Vocabulary Sets”.

In addition to the user’s choice within a system, if the user could repeat a set of exercises the software received the label of “Repetition: Exercises”. In contrast, if the user could not repeat individual exercises, but could repeat a set of exercises the software would receive a label of “Repetition: Units”. Additionally, if the user could repeat input within an exercise the system received the label “Repetition: Answers”. If there was a restriction within the system as to how many times the user could repeat an answer, the software received either the labels of “Repetition: Answers: Limited” or “Repetition, Answer: Unlimited”.

In addition to choice and repetition, we investigated “Feedback” and “Difficulty”. If the user can give input on the system as a whole or the systems’ correction of the user’s input the system received the label “Feedback”. The feedback could occur in or outside of the system -- a separate link to the software website, or a discussion board or blog within the software. If the user can change or alter how difficult an exercise or set of exercises are within the system the software received a label of “Difficulty”.

3.3 User Input

First, we differinciated between input the user gave in and outside of the software system. The input that users gave outside of the system received the label “Outside” and any input given by the user within the software system received the lable “Inside”. From here different forms of “outside” and “inside” input was broken down into different forms (Figure 3):

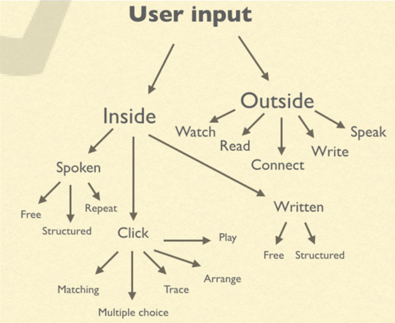

Figure 3: User Input.

If there was a text, passage, blog, post etc that the system didn’t track if or when the user read the content it received the label “Outside: Read”. Additionally, videos or clips provided by the system without awareness of user acess received the label “Outside: Watch”. Along the same lines- if the system provided printable worksheets with activities or exercise or provided a downloadable PDF versions of it's content it received the label of "Outside: Write". If the system connected a user with another user for instruction or speaking practice, but the user had to connect with them outside of the system using Skype or another medium it received the label "Outside: Speak". If the system connected users but did not give feedback on user's input like in a discussion forum the system received a label of "Outside: Connect". If, in contrast, the system connected users and the it recorded, responded, or tracked the users spoken input the system received a label of "Inside: Spoken".

If the system recorded the spoken input of the user, the recorded input was divided into whether the input was a repetition of a pronunciation (label "repeat"), a spoken response following an exact pattern specified by the system (label "structured"), or a spoken response with no constraints by the system (label "free"). The "structured" response could also be in response to a multiple choice or fill in the blank task. Similarly, if the user's written input needed to follow an exact pattern specified by the system it received the label "Written: Structured"- if not it, it received the label "Written: free".

If the user's input was in the form of clicking, the input was distinguished by the task the user was preforming: matching text or objects (label "matching"), a multiple choice of objects, pictures or texts (label "multiple choice"), tracing letters and numbers ("trace"), arranging words or letters ("arrange"), or playing a mini-game (label "play").

3.4 Educational Framework

The final categorization we completed was for educational psychological frameworks. Any system that emphasised the repetition of material for acquisition of language or the refinement of user input through reward and punishment, it received the label "Behaviourist". Furtherore, any system that emphasized stages of Bloom's Taxonomy in its content received the label "Cognitivist". Finally, any system that had aspects of user-user collaboration, interaction or competition, as well as systems with exploratory content received the label of "Constructivist".

4 RESULTS

The following section relays the results of the analysis. The results are organized according to each topic and research questions.

4.1 Software Feedback

As to the feedback present in current CALL systems, 91.30% presented user awareness at some level. Overall 79.71% presented immediate feedback and 69.57% presented accumulated feedback.

Of the immediate feedback and in response to user input, 23.19% of the systems played a sound bite. Overall, 88.71%, identified if the user input was correct or incorrect (label "answer"). Of those, 7.27% gave a translation, 47.27% gave an explanation, and 5.46% gave a hint in response to incorrect user input (see Figure 4).

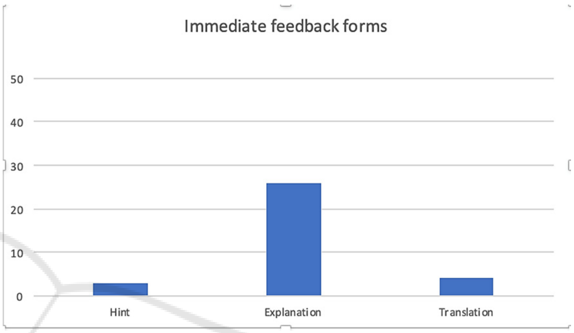

Figure 4: System response to incorrect input.

In total, 93.75% of the systems tracked the progress of the user, and 45.83% tracked the user's knowledge. Additionally 22.92% of software systems set goals for the user and 10.42% offered a leader board (see Figure 5).

Overall, 40.58% of the systems had elements that did not have an awareness of user input. Of these, 72% offered translations, 92% offered audio clips, and 32% offered explanation.

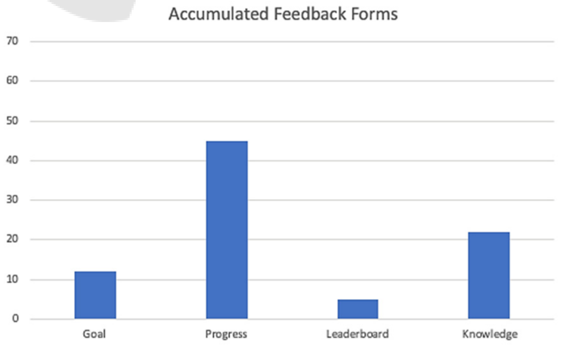

Figure 5: Accumulated feedback.

From our analysis, the systems present rudimentary adaptability to the user's needs. Though there was a variation in the forms of feedback in our analysis, there was no adaptation of feedback to learner's problem solving strategies or repeated mistakes. In general, the varying of feedback forms within a system was not correlated to individual user input.

4.2 Adaptation

From the investigation of system adaptation, only 11.59% of the software allowed for user feedback. Likewise, only 10.15% of the software allowed for user control of difficulty. Overall, 95.65% of the software allowed for repetition of content. Of those, 66.67% allowed repetition of exercises and 56.06% allowed for repetition of units. As to the repetition within tasks, 56.52% allowed for the repetition of input within tasks. Of these, 58.97% allowed for unlimited repetition and 41.03% allowed limited repetition.

As to choice, 66.18% of systems surveyed allowed for user choice. Of those, 98.55% allowed for user choice of level, 63.24% allowed for user control over exercises, 8.82% allowed for user control over the vocabulary, and only 7.35% of systems allowed control over user level (See Figure 6).

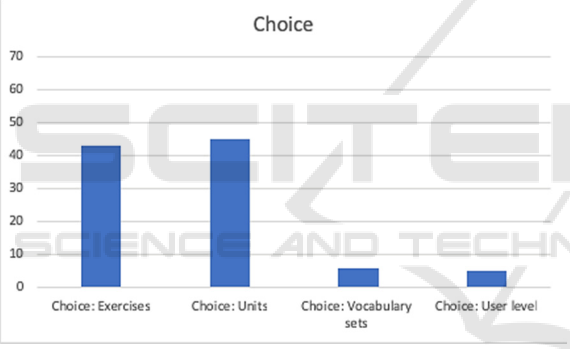

Figure 6: User choice within the systems.

Most systems we analysed had little to no system adaptability or flexibility. Though users had some control over the content of the systems, as well as some control over repetition and difficulty, users did not have unrestricted access to the content. In all cases the systems, in the least, determined the method for which the users explored the content. Likewise, though some systems changed content based on user selection of vocabulary or topic, no system adapted the learning path of the content to the user- a characteristic of a true adaptive ITS system (Martens 2004). Thus, though the systems had aspects of adaptability present, they could not be considered to have high adaptability.

4.3 User Input

From the analysis user input, we found that 44.93% of systems allowed for input "Outside" of the system and 88.41% allowed for input "inside" the system.

Of the input outside of the system, 54.84% provided reading material, and 38.71% provided videos to watch. Furthermore, 45.16% connected users in either a CMC or CSCL format. Additionally, 35.48% provided a platform for face to face interaction (label "Speak") and 22.58% provided a platform for written discussion.

Of the input inside of the system, 91.80% was "Click", 47.54% was "Written" and 34.43% was "Spoken" (See Figure 7).

Of the click input, 89.29% was a multiple choice task, 48.21% was an arranging task, 10.71% was a tracing task, and 21.43% was a mini-game (label "play"). Additionally, of the written input, 69.23% was structured, and 34.62% was free. Of the spoken input, 95.24% was repetition, 66.67% was a structured response and 23.81% was a free response.

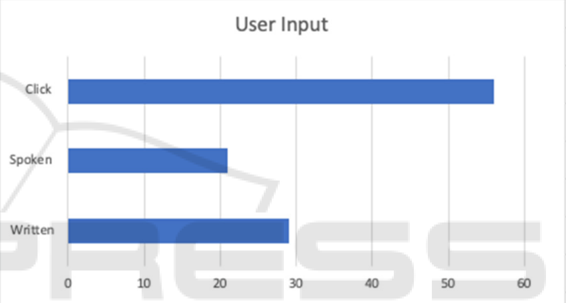

Figure 7: User input in the systems.

Strictly from a task-based perspective, most investigated systems provided variation in user input. It was evident in our analysis that many systems preferred one form of input over another, but even if the system restricted user input to one form (for example, only written input), the systems still provided a diverse amount of tasks within that form.

4.4 Educational Framework and System Identification

From the analysis of software it is clear that a majority of the CALL software, 81.16% can be categorized as Behaviouristic or having behaviourist elements. Additionally, 24.64% were categorized as having cognitivists elements and 31.88% were categorized as constructivist. Furthermore, 15.90% were CMC systems, 5.80% were CSCL systems, and 88.40% were NBLT systems. Finally, 11.59% of the systems were ILE- the remaining were categorized as ITS (See Figure 8).

Software	System		Framework
17 min Languages	NBLT	ITS	Behaviourist Constructivist
ABA English	NBLT CMC	ITS	Behaviourist
AudioNovo	NBLT	ITS	Behaviourist
Babble	NBLT	ITS	Behaviourist
Book Punch	NBLT	ITS	Cognitivist
Business Letter Punch	NBLT	ITS	Cognitivist
Busuu	NBLT CSCL	ITS	Behaviourist Constructivist
Cant Wait to Learn	NBLT	ITS	Behaviourist Constructivist
Capt'n Sharky	NBLT	ITS	Behaviourist
Confused Words Fix Up	NBLT	ITS	Behaviourist
Critical Thinking Skills: Upper Grades	NBLT	ITS	Behaviourist
Critical Thinking Skills: Reading	NBLT	ITS	Behaviourist
CyberTeachers	NBLT CMC	ITS	Behaviourist
Drops	NBLT	ITS	Behaviourist
Duolingo	NBLT CSCL	ITS	Behaviourist Constructivist
Earworms	NBLT	ITS	Behaviourist
Easy Peasy: English for Kids	NBLT	ITS	Behaviourist
eLanguage	NBLT	ITS	Behaviourist Cognitivist
Emil und Pauline Auf dem Hausboot	NBLT	ITS	Behaviourist
Emil und Pauline Auf Madagaskar	NBLT	ITS	Behaviourist
Emil und Pauline Deutsch und Mathe	NBLT	ITS	Behaviourist
Emil und Pauline in England	NBLT	ITS	Behaviourist
Exceller	NBLT	ITS	Behaviourist
Go Talk	NBLT	ITS	Behaviourist
Grammar Fitness - Advanced	NBLT	ITS	Behaviourist
Grammar Fitness - Basic	NBLT	ITS	Behaviourist
Grammar Fitness - Intermediate	NBLT	ITS	Behaviourist
HandsOn Turkish-compact	NBLT	ITS	Behaviourist
HandsOn Turkish	NBLT	ITS	Behaviourist
iTalki	CMC	ILE	Constructivist
Johnny Grammar Word Challenge	NBLT	ITS	Behaviourist
Kidspeak	NBLT CSCL	ITS	Behaviourist Constructivist
LearnEnglish	CMC	ILE	Constructivist Cognitivist
LearnEnglish Grammar	NBLT	ITS	Behaviourist Cognitivist
LearnEnglish Kids: Playtime	NBLT	ITS	Behaviourist
LearnOasis	NBLT	ITS	Behaviourist
Lernerfolg Grundschule	NBLT	ITS	Behaviourist
LinguaLeo	NBLT	ITS	Behaviourist
LinguaPlex	CMC	ILE	Constructivist
Little Pim	NBLT	ITS	Behaviourist
Memrise	NBLT	ITS	Behaviourist
Michel Thomas Method	NBLT	ITS	Behaviourist
Mondly	NBLT	ITS	Behaviourist Constructivist
MosaLingua	NBLT	ITS	Behaviourist
Muzzy	NBLT	ITS	Behaviourist
Open Punch	NBLT	ITS	Cognitivist
Paragraph Punch	NBLT	ITS	Cognitivist
Pimsluer	NBLT	ITS	Behaviourist
Pim Track	NBLT	ITS	Behaviourist
Preply	CMC	ILE	Constructivist
Prinzessin Lillifee	NBLT	ITS	Behaviourist
Reading Comprehension Booster	NBLT	ITS	Behaviourist
Reading Skill Builder	NBLT	ITS	Behaviourist
RealTalk	CMC	ILE	Constructivist
Rocket Languages	NBLT	ITS	Behaviourist Cognitivist
Rosetta Stone	NBLT	ITS	Behaviourist
Speexx	NBLT	ITS	Behaviourist

Figure 8: Software identification.

Software	System		Framework
Starter Paragraph Punch	NBLT	ITS	Cognitivist
Study Cat	NBLT	ITS	Behaviourist
Tandem-Language Exchange	CMC	ILE	Constructivist
Teach Your Monster To Read	NBLT	ITS	Behaviourist
The Talk List	CMC	ILE	Behaviourist
Transparent Language Online	NBLT	ITS	Behaviourist Cognitivist
UTalk	NBLT	ITS	Behaviourist
Verbling	CMC CSCL	ILE	Constructivist
Yabla	NBLT	ITS	Behaviourist
Vocabulary Stretch	NBLT	ITS	Behaviourist
Vocabulary Super Stretch	NBLT	ITS	Behaviourist
Voxy	NBLT CMC	ITS	Behaviourist Constructivist

Figure 8: Software identification (cont.).

Our analysis found that the systems which were primarily NBLT, even if they had CMC or CSCL elements were ITS systems. Furthermore, we found that there was much variation in the extent of user adaptability within the ITS systems. Some systems simply track user progress or user knowledge, but have little to no adaptation of the content or system to this tracking.

5 FURTHER RESEARCH

From a CALL perspective, only the forms of NBLT, CSCL, and CMC were analysed. Further research is needed in order to fully understand all CALL software forms available to users. Furthermore, an even greater database would be beneficial to the analysis of current systems.

As to interface of these systems, no analysis was done. The examination of the relation of software's interface, including changes with the addition of a third party like a teacher or parent would be beneficial. More information is needed as to how each software incorporates these third parties and what changes that causes (if any)to the workings of the software units, activities, reward systems or educational frameworks. Subsequently further investigation is needed into intended users. Is the software geared to 3 year olds or 10 year olds? What is the gender of the intended user? Can the software be adaptive to group settings or peer-to peer activities or just is it primarily for individual users? Additionally, further study could investigate what percentage of each program incorporates behaviourist elements like drill and practice, cognitivist elements that mirror Bloom's Taxonomy or constructivist elements of peer-to-peer collaboration.

In addition to these further possibilities, the most important further examination that is needed is in regard to the ITS in CALL software. Due to the great variability in user awareness and adaptation, a detailed analysis is needed to determine what specific aspects of a CALL system are flexible and how are they adaptive to the user. Withstanding that from this investigation a spotlight will be shown on which systems have the highest user adaptability.

Finally, the additionally and private goal of the authors' is in-depth investigation into what a constructivist language learning software demands in comparison to a behaviourist language learning software. This can be accomplished through the analysis of a constructivist focused software system like Minecraft. From here, the elements of a constructivist learning environment can be grasped and said concepts can be applied to a language learning software.

REFERENCES

Blake, R. J. (2011). Current trends in online language learning. *Annual Review of Applied Linguistics,* 31, 1-17.

Bloom, T. M. E. (1965). Bloom's taxonomy of educational objectives. Longman.

Bordonaro, K. (2003). Perceptions of Technology and Manifestations of Language Learner Autonomy. *CALL-EJ Online*, *5*(1), 1–19.

Brusilovsky, P.: Intelligent Learning Environments for Programming: the Case for Integration and Adaptation. *Proc. of AI-Ed*, Washington, US, (1995) 1–7

Chapelle, C. (1998). Multimedia CALL: Lessons to be learned from research on instructed SLA. *Language Learning & Technology*, 2(1)

Chapelle, C. A. (2001). *Computer applications in second language acquisition*. Cambridge University Press.

DeHaan, J. (2005). Learning language through video games: A Theoretical Framework, an Evaluation of Game Genres and questions for Future Research. SP Schaffer and ML Price (Eds), 229-239.

Doughty, C. (1987). Relating second-language acquisition theory to CALL research and application. In W.F. Smith (Ed.), *Modern Media in Foreign Language Education: Theory and implementation* (pp. 133- 167)

Egenfeldt-Nielsen, S. (2006). Overview of research on the educational use of video games. *Nordic Journal of Digital Literacy*, 1(03), 184-213.

Harrer, A., & Martens, A. (2006, June). Towards a pattern language for intelligent teaching and training systems. In *International Conference on Intelligent Tutoring Systems* (pp. 298-307). Springer, Berlin, Heidelberg.

Hogle, J. G. (1996). Considering games as cognitive tools: In search of effective" edutainment.". *ERIC* Clearinghouse.

Martens, A. (2003). Centralize the tutoring process in intelligent tutoring systems. *In Proc. of the 5th Internat. Conf. New Educational Environments ICNEE*, Lucerne, Switzerland.

Martens, A. (2004). *Ein Tutoring Prozess Modell für fallbasierte Intelligente Tutoring Systeme* (No. 281). AKA-Verlag für Wissenschaft, Medizin und Technik.

Martens, A., & Hellmig, L. (2014, July). Blends, Patterns, Und Flips--A Method-Based Approach. In 2014 IEEE 14th *International Conference on Advanced Learning Technologies* (pp. 393-395). IEEE.

Mitchell, A., and Savill-Smith, C. (2004). The use of computer and video games for learning: A review of the literature. London, UK: Learning and Skills Deveopment Agency.

Niederhauser, D. S., and Stoddart, T. (2001). Teachers' instructional perspectives and use of educational software. *Teaching and teacher education*, 17(1), 15-31.

Hegelheimer, V., & Chapelle, C. A. (2000). Methodological issues in research on learner-computer interactions in *CALL. Language Learning & Technology*, 4(1), 41-59.

Hulstijn, J. H. (2000). The use of computer technology in experimental studies of second language acquisition: A survey of some techniques and some ongoing studies. Language Learning & Technology, 3(2), 32-43.

Pennington, M. C., & Stevens, V. (1992). Computers in Applied Linguistics: An International Perspective. Multilingual Matters 75. Bristol, PA: Multilingual Matters.

Rieber, L. P. (2005). Multimedia learning in games, simulations, and microworlds. The Cambridge handbook of multimedia learning, 549-567.

Scott, T. et al. (1992). Chapter 5: Computers and education: A cultural constructivist perspective. *Review of research in education*, 18(1), 191-251.

Stockwell, G., and Tanaka-Ellis, N. (2012). Diversity in environments. Computer-Assisted Language Learning. *Diversity in Research and Practice*, 71-89.

Stockwell, G. (Ed.). (2012). *Computer-assisted language learning: Diversity in research and practice*. Cambridge University Press.

Wang, Q., et. al. (2009). Investigating critical thinking and knowledge construction in an interactive learning environment. *Interactive learning environments*, 17(1), 95-104.

Zhang, D., et. al. (2004). Can e-learning replace classroom learning?. *Communications of the ACM*, 47(5), 75-79.

Zhao, Y. (2003). Recent developments in technology and language learning: A literature review and meta-analysis. *CALICO journal*, 7-27.