Learning a Song : an ACT-R Model

Belkacem Chikhaoui, Hélène Pigot, Mathieu Beaudoin,
Guillaume Pratte, Philippe Bellefeuille and Fernando Laudares

Abstract**—**The way music is interpreted by the human brain is a very interesting topic, but also an intricate one. Although this domain has been studied for over a century, many gray areas remain in the understanding of music. Recent advances have enabled us to perform accurate measurements of the time taken by the human brain to interpret and assimilate a sound. Cognitive computing provides tools and development environments that facilitate human cognition simulation. ACT-R is a cognitive architecture which offers an environment for implementing human cognitive tasks. This project combines our understanding of the music interpretation by a human listener and the ACT-R cognitive architecture to build SINGER, a computerized simulation for listening and recalling songs. The results are similar to human experimental data. Simulation results also show how it is easier to remember short melodies than long melodies which require more trials to be recalled correctly.

Keywords**—**Computational Model; Cognitive Modeling, Simulation; Learning; Song; Music.

I. Introduction

MUSIC has been studied by multiple domains and researches are still ongoing, focusing on its perception and transmission. The oldest and commonest way to transmit it is most certainly oral tradition. In this mode, the singer (transmitter) teaches the composition to another singer (receiver) through several repetitions. This presupposes accuracy on both the transmitter's and the receiver's sides [1]. According to common practice, learning verbal material through song should facilitate word recall [2]. Studies demonstrated that learning lyrics would be easier if accompanied by music [3]. The benefits of music are undeniable. For example, recent research has shown that the use of music is a great way to learn foreign languages [3]. Furthermore, the general hypothesis is that music enhances a learning environment due to its emotional power [4]. Music is wonderful for teachers to play quietly in the background when teaching new concepts to their students. Indeed, S.H. Russell has shown that students learned and remembered mathematics, spelling, science and history facts quicker and easier while listening to music.

Oral learning is done essentially through repetition which is also the basic mechanism for learning music. In a study on learning lyrics conducted by [2], the authors tested if repetition of a line of text helps the participants remember it. A better recall was observed for the first line of lyrics when it was repeated twice, while the recall of the rest of the lyrics declined as the song progressed. To study how people retain music, experiments have been conducted to explain, firstly how sounds and music are processed by the brain and secondly which memories are involved when learning songs. Today, we get a clear understanding of how the brain interprets the sound. It is first analyzed by the inner ear and converted into nerve impulses [5], this step lasts between 2 and 5 milliseconds [6]. The audio signal then goes back up to the cerebral areas of the conscience. By doing so, the information is purified and refined [6]. The conscience identifies the nature of the audio signal and interprets it; this step lasts about 50 ms. Therefore, perception and interpretation of a sound, such as a note, last for up to 55 ms [6]. As a consequence of this processing time, it would be hard for a human to distinguish a set of notes when their interval is less than 55 ms; in musical terms, this implies a rhythm of approximately 900 beats per minute. At a higher rhythm, the notes would appear to be mixed to a human listener. After the sounds have been interpreted in notes, the music is remembered using the notes' characteristics and stored in specific memories. The melody is composed of successive lines of single tones or pitches perceived as a unit. More often, adults store melodic information under relative frequencies patterns, and readily recognize a melody, whether sung in a high or a low pitch range [7]. The sounds are stored in a phonological loop, which plays an important role in the acquisition of acoustic signal systems like language and music [8]. This shows the usefulness and importance of the working memory in retaining melody, especially during the recall phase [9]. The main goal of our project is to simulate the three steps of song's processing : the listening, memorizing and recalling phases. The model we built, called SINGER, should be consistent with results found in the recall theories, particularly with the number of repetitions needed to learn a song. We assume that the sounds have been previously processed by the brain. We focus on the way music is remembered, rather than the kinds of memory involved.

To achieve our goals, we first represent the music with commands that can be interpreted by the ACT-R cognitive architecture. ACT-R is a production system based architecture in which knowledge is represented as facts and rules [10], [11]. The commands are loaded into our model to be processed and manipulated using specific ACT-R perceptual motor modules such as the auditory module. The song should be heard several times before the recall phase in order to provide evidence of the repetition benefits. The aim is to simulate how the human brain retains a simple melody. The SINGER model is built upon the cognitive architecture ACT-R which provides tools based on psychological theories.

II. Theoretical background

In this section we describe firstly how the concept of omission is modeled in ACT-R, secondly the audio module

Belkacem Chikhaoui, Hélène Pigot, Mathieu Beaudoin, Guillaume Pratte, Philippe Bellefeuille, Fernando Laudares are with the Computer Science Department, Faculty of Science, University of Sherbrooke, Sherbrooke QC J1K 2R1 Canada, email: {belkacem.chikhaoui@usherbrooke.ca}

of ACT-R used in our study to simulate song's learning, and finally the work done previously in ACT-R to help modeling music recall.

A. Omission modeling in ACT-R

To model the memory capacities of human being, ACT-R provides a way to simulate omission. Omission is modeled using a subsymbolic mechanism. Each chunk in declarative memory is associated with a subsymbolic value called "activation level". Only chunks with the highest activation levels will be selected when a retrieval request is made to the memory. Similarly, a subsymbolic value called "utility" is associated to each production rule, The utility value reflects the contribution of the rule in terms of achieving the current goal. When a conflict situation arises during the production rules selection, a mechanism called conflict-resolution selects the one with the highest utility.

B. Audition module in ACT-R

The audition module is part of the perceptual-motor modules. Our model uses the ACT-R audition module to "hear" the music. Each audio event is simulated in ACT-R. When an audio event is detected, a chunk is sent to the primary buffer of the audition module: the audicon on request (via a production rule). Two other subsystems, the positional system (where) and the identification system (what) will process the audio event in order to send it to the audio buffer. The positional system is used to find an audio event in the audicon. According to some constraints provided by a production rule (for example, the first audio event that has not been processed yet), the system will put the chunk representing that audio event in the aural-location buffer. The identification system is used to attend to audio events which have been found by the positional system. It allows the model to determine whether the sound is a type of speech, a tone, or a digit. ACT-R supports only these three sound types. This system must be invoked by a production rule (much like the positional system): it takes the audio event from the aural-location buffer, requests to shift aural attention to that specific audio event, processes the sound and creates a chunk representing the sound into the aural buffer [12] [13].

C. ACT-R and music recall

In ACT-R cognitive architecture as the other cognitive architectures, there is no specific mechanism dedicated to music processing, and there are no models in our knowledge developed in ACT-R to simulate music learning. However, the emphasis is made on the conceptual part of recall, either serial recall or free recall, and in terms of forward or backward recall of words or texts as reported in Anderson et al [14]. In ACT-R, the length of words to retain affects considerably the correctness of the recalled version of these words [14]. As reported in theories of perceptual processing, high load conditions leads to difficulty in maintaining a perceptual trace over a length of time and require active rehearsal, such as retaining a pitch in memory while other tones are presented [15]. The concept of rehearsal is proved empirically in [14] [16] and carried out in some models developed using the ACT-R cognitive architecture [14]. Therefore, a model of songs recall should then be based on the recall's characteristics mentioned above.

III. SINGER Model implementation

This section focuses on the simulation of the cerebral area's processes to interpret and store a sound. According to music theory, songs are decomposed hierarchically in two levels: the musical phrase and the note. A song is composed of phrases defined as a logical grouping of notes. A note is characterized by its frequency, its duration and the time position where it appears in the song. Our model, SINGER, listens and recalls songs defined as a set of musical phrases. The next sections present the SINGER implementation in ACT-R. First, an audio event is generated for a specific note. Then, during the listening phase, SINGER recognizes a sound placed in the audio buffer and stores it in the declarative memory. Finally, SINGER tries to recall the song. The learning and recall processes will be conducted in three main phases. These phases are detailed in the following sections.

A. Phase I: Notes generation phase

As described above, ACT-R supports only three types of sounds, namely: speech, tone and digit. Creating a note in ACT-R is carried out using the new-tone-sound command. This command simulates the conversion from the external sound to an internal sound, which can be decoded by the ACT-R architecture. Unfortunately, this command does not allow the extraction of the note's characteristics from the audio buffer, only its frequency is available. We therefore use the Musical Instrument Digital Interface (MIDI) standard to represent a note. The three sounds characteristics of (MIDI) such as the pitch, the duration and the intensity are then extracted using the new-other-sound command that gives more accessibility for the sound's characteristics. We first develop a software module which converts the original audio file, stored in MIDI (.mid) format, into new-other-sound ACT-R commands with the relevant parameters. The original audio file contains a melody composed of a single track, where only one note is played at a time. The converter program is independent from the SINGER model.

The new-other-sound command uses the following syntax: (new-other-sound content duration delay processing-time instr absolute-time) where the parameters are the following:

- content: characteristics of a note. This list of parameters defines the note to be stored in the memory. The note's parameters are described in the listening phase section;
- duration: sound duration in milliseconds;
- delay: additional delay in milliseconds. The delay indicates a delay before the audio event. It is always set to 0 in our model;
- processing-time: time in milliseconds. The processing-time indicates the time taken by ACT-R to decode this sound;
- instr: not used in our model;

- absolute-time: absolute time in milliseconds. It indicates when the audio event will be triggered in ACT-R.

The processing time for decoding the sound is estimated at 5 milliseconds, in order to conform to the time found in the literature. The complete time for perceiving and interpreting the sound is obtained by combining the processing time of the production rule (50 milliseconds) and the processing time of the audio event (5 milliseconds), for a total of 55 milliseconds. To allow the listening of a song, a list of new-other-sound instructions is loaded into SINGER, modeling the song's notes.

B. Phase II: listening phase

The listening phase begins with the ACT-R new-other-sound commands generated during phase I. This phase is the core of our model as it is responsible for processing the musical phrases and notes. We first present the hierarchical representation of the songs using musical phrases and notes; the listening process follows based on the usual learning process of ACT-R.

1) Musical phrase: The concept of musical phrase remains vague in the literature. It is defined as a set of notes constituting a logical whole which feels natural to humans. Despite the huge studies on musical macrostructure since more than one century, there is no enough information to easily characterize a phrase [15]. In our model, a musical phrase is therefore defined as a set of notes of fixed duration. This characteristic is found among western songs. Each phrase contains notes and silences, the latter is characterized by the absence of notes. In our model, the phrase is identified by a numeric index which, multiplied by the phrase duration, gives us the absolute time where it starts in the melody.

2) Note: In our model, the note is the basic memory unit for the computational representation of the melody. It is defined with the following characteristics:
(note frequency duration position position-next phrase-of-belonging is-the-first-note) where the attributes are the following :

- frequency: is the sound frequency in Hertz.
- duration: is the note duration in milliseconds.
- position: is the relative position of the note in its musical phrase, measured in milliseconds.
- position-next: is the position of the next note in milliseconds. It indicates the relative position of the next note in the musical phrase, measured in milliseconds.
- phrase-of-belonging: is the index of the musical phrase to which the note belongs.
- is-the-first-note: is a boolean. It specifies if the note is the first one of the musical phrase.

The note representation is supported by Halpern, Levitin and Cook [15], in which the representation of a learned melody preserves multiple properties, such as absolute frequency and precise tempo of notes.

3) Music processing: Given these phrase and note definitions, we now explain the music processing as well as the memory modules involved. The conversion and decoding process of the melody into ACT-R commands, along with the loading of notes into the SINGER model, constitute the strong aspects of our model. This process realistically simulates the listening phase of the song. The listening phase responds to audio events generated by the ACT-R command new-other-sound. This phase lasts as long as audio events remain to be processed. These events represent the notes of the melody currently being perceived and learned. As described previously, audio events are placed in the aural-location buffer, which simulates the passage from the external ear to the internal ear. For each new sound detected, a production rule captures the event and stores it temporarily in the audio buffer. The sound is then available to SINGER, which extracts the note's characteristic and stores it in declarative memory. The song is then represented as a set of notes, which are linked by the mean of the phrase they belong to and the position of the next note in the phrase.

C. Phase III: recall phase

The recall phase is the final process of SINGER. The two previous phases have highlighted the listening, processing and storing steps. The notes are so far stored in the declarative memory and should be recalled by SINGER to sing the song. The recall is made sequentially using the goal memory chunk to indicate the next note to be recalled. The recall phase uses the subsymbolic activation mechanism to fetch elements in declarative memory, and the calculation of production rules utility when a conflict situation arises. A recall fails when the note is not retrieved. It leads to an error state, which can be resolved in two ways. Either SINGER tries continuing to the next phrase by recalling the first note of the next musical phrase, or it tries recalling the first note of any musical phrase chosen randomly. In the second case, SINGER retrieves from its memory any note which begins a phrase. The selected note is the one with the highest activation level. In case of errors, SINGER will more likely jump to the next phrase, but still the utility of the by-chance-production would lead it to pick up any phrase in memory. This error mechanism allows SINGER to forget a note; but if it remembers it, it will do so perfectly. No mistake can be made upon the characteristics of a note. Once a note is recalled, its characteristics are written in an output file that will later be used to create the final melody. The output file contains all the frequencies, absolute temporal positions and durations of each note recalled. At the end, this file is converted into MIDI format through a reverse converter. This MIDI file contains the SINGER's recalled version of the original song. Figure 1 summarizes the SINGER simulation, from the initial song presented in MIDI format to the recalled song restituted in MIDI format.

IV. Experiment

SINGER simulates people learning a song. The learning process is highlighted by presenting the songs more than once during the listening phase. Each time a song is presented to SINGER, it stores the notes in the declarative memory. In fact, when a note is presented (with same characteristics), SINGER reinforces its activation level in the memory. Multiple listenings leads to raising the activation level of the notes, which then facilitates the recall phase by diminishing the chances of omission. We conduct two experiments: one to compare the results with the literature, a second to explore SINGER's capacities. In the literature, the effect of repetition has been tested with human subjects [2]. A song is sung once to each participant in order to be familiarized with the new song. Afterward, the first line is presented and the participant has to repeat it. Lines are then repeated two by two. Each line of the song is repeated only once, except the first line which is repeated twice. This experiment is conducted with 36 participants. In our experiment, the simulations aim to reproduce the repetition effect. The song is presented to SINGER from one to four times before the recall phase. Contrary to the experiment in [2], we did not present the song previously and the repetitions occurred for the whole song at once.

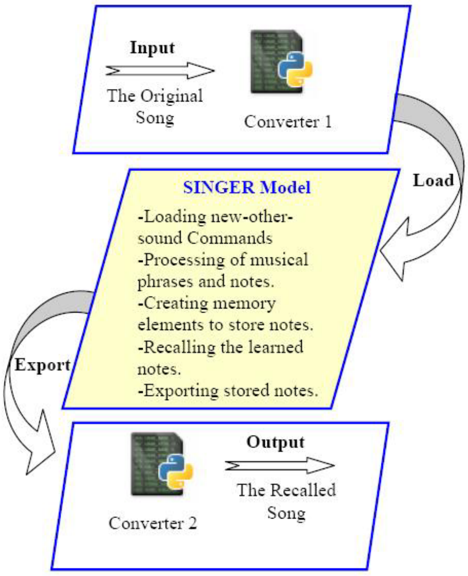
Input
The Original Song
Converter 1
Load
SINGER Model
-Loading new-other-sound Commands
-Processing of musical phrases and notes.
-Creating memory elements to store notes.
-Recalling the learned notes.
-Exporting stored notes.
Export
Converter 2
Output
The Recalled Song

Fig. 1. The SINGER Model.

The second experiment tests how SINGER copes with the song's length. It is expected that it needs more repetitions to learn a longer song. The simulation is conducted using two different songs: experiment A with 32 notes, experiment B with 62 notes.

V. Results

A. Repetition effect

The model was run 100 times for each listening case. A listening case is determined by the number of repetitions in the listening phase, from one (S1) to four (S4). The results of these simulations are expressed in percentages of notes well recalled (Table I). The success rate increased with the number of repetitions, leading to a quite perfect recall after four repetitions.

The comparison of our simulations with the literature experiment is difficult because results in the literature are expressed in percentages of words recalled, and number of lines attempted [2]. Moreover the protocol of the experiment was not reproduced exactly in the simulation, where the song is not presented line by line. Still, as shown in table II, the results in the literature present the same tendencies as the S2 listening case, where the song is sung twice for both experiments.

TABLE I
Simulation results of the SINGER model in percentage.

	S1	S2	S3	S4
Recall (%)	36.2	62	87	95
SE	4.0	3.0	5.9	3.4
Subjects	100	100	100	100

TABLE II
Results (Mean Recall and Standard Error) of Peretz & al. and the SINGER model.

	Peretz & al.	SINGER Model	
	M	S1	S2
Recall (%)	60	36.2	62
SE	3.2	4.0	3.0
Subjects	36	100	100

The recall proportions of our model in the first simulation are lower compared with those found in [2]. On the other hand, the recall proportions of the second simulation are similar to those found in the literature. According to common practice and theories of learning, repetition enhances the learning success. Results in the third and fourth simulations show that the learning rates increase when the number of repetitions increase as shown in figure 2. These results are consistent with the theory.

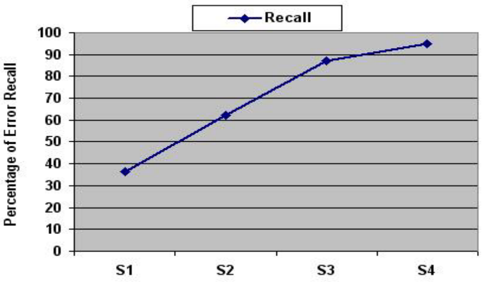

Fig. 2. Increase of learning rate depending on the number of repetitions

B. Simulation results for long melodies

A long melody was presented 50 times to the SINGER in order to compare the results between learning a long melody and a short one as shown in table III.

As predicted, the recall proportions are lower than those obtained in experiment A, using a shorter song (table I). We observ that the number of repetitions slightly increases the learning rate, the recall increasing from (18.1 %) in S1 to (21.2 %) in S2. These results are lower than those with a short melody, where the recall went from (36.2 %) in S1 to

TABLE III
LONG MELODIES SIMULATION RESULTS (RECALL AND STANDARD ERROR) OF THE SINGER MODEL.

	S1	S2	S3	S4
Recall (%)	18.1	21.2	23.7	42.5
SE	9.1	12.7	11.4	20.7
Subjects	50	50	50	50

(62 %) in S2. These results are shown graphically in figure 3.

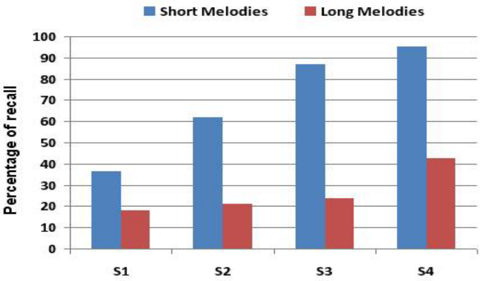

Fig. 3. Comparison of the results of the first and second experiment.

In the first experiment, the SINGER goes through a small number of repetitions to be able to recall the song perfectly. However, in the second experiment, the SINGER must listen to the song more times to be able to recall it perfectly. The error rate in the first experiment is reduced significantly from (63.8 %) in S1 to (5 %) in S4, unlike in the second experiment, where the error rate decreased slightly, from (81.9 %) in S1 to (57.5 %) in S4.

VI. GENERAL DISCUSSION

The aim of our project was to simulate the song learning process, focusing on the memorization aspect. It was implemented using the ACT-R cognitive architecture. The results of our model were compared with those obtained by [2]. The experiment of [2] was made with two types of participants: musicians and non-musicians, unlike our model, which is a generic one that makes no difference between musicians and non-musicians. In the first experiment, the results of the first simulation, in which SINGER listens to the song once, were lower than those obtained by [2] (36.2 % vs 60.0 % of recall). The results of the second simulation, in which the SINGER listens to the song twice, were almost the same (60 % vs 62 % of recall). Those results could be explained by the fact that the participants in [2] had already heard the song once to familiarize themselves with it before the first test. According to [2], recall of the first song lines should be better than the rest, and successes in recalling decrease as the song progresses. Similarly, SINGER seems to remember the beginnings of the songs with fewer errors. This is due to the rehearsal effect required to maintain the beginnings of the songs, as observed in [14]. We also observed that, in several cases, when SINGER forgot a phrase, he also often forgot the following phrase, which is similar to the results obtained by [2] and [17]. Finally, the recall of the last elements was generally prone to fewer errors, which is in accord with the results obtained by [2]. The SINGER model supports the concept of repetition used to enhance the learning rate, which is illustrated by S3 and S4 simulations with a recall success rate of 87 % and 95 % respectively in the first experiment. In the second experiment, where SINGER learns long melodies, we found that the recall error rate was high in the first simulation (81.9 %), and decreased only slightly with increasing repetitions. In fact, the recall error rate reached 57.5 % in S4, which is far greater compared to that of S4 (5 %) in the first experiment. This could be explained by the limited capacity of the working memory, which may only contain few elements [9]. As reported in [14], the list length will have an effect on recall, which support the results obtained with long melodies. Thus, in general the correspondence between the theory and our data is quite good.

VII. CONCLUSION

During our research, we observed a scarcity in the documentation about the audio module of the ACT-R architecture. However, we knew that the perceptual module functionalities were similar to the visual module ones, which are better documented and used in several experiments [18], [19], [20]. The new-tone-sound instruction did not fulfill our needs. We overcame this limitation by using the new-other-sound instruction, augmented with the necessary parameters. The conversion and decoding process of the melody into ACT-R commands, along with the loading of commands into the SINGER model, constitute the strong aspects of our model. This process realistically simulates the listening phase of the song. This is a pragmatic methodological choice. Other methodological choices have also been used to simplify the recall process in order to facilitate its implementation, such as the linked list used to fetch the subsequent notes of a musical phrase, a flawless recall of the note's characteristics and the management of recall errors. Some of these choices are supported by scientific literature, such as the use of the musical phrase structure [15], while others were consequences of the technological challenges of implementing a simulation of human cognition. The results of the experiment show that when SINGER listens to a song once, it recalls its beginning and ending with more precision than the middle elements. However, the recall becomes better when SINGER listens to the song several times before moving to the recall phase. In comparison, a longer melody requires more listenings for a perfect recall. In the case of learning long melodies, we did not find the corresponding results in the literature. Therefore, our simulation results add new data to this research area.

VIII. FUTURE IMPROVEMENTS

Some improvements should be brought to the SINGER model. First, the actual model makes mistakes only on the sequence of notes in the songs. It should be extended to allow errors upon the note's characteristics, such as its frequency. When an error occurs, SINGER selects the first note of the following phrase or of a phrase selected randomly. The cognitive architecture ACT-R provides a partial matching mechanism, where similar memory units are mixed up. Adding similarity criteria between notes will add potential errors where a quasi similar note should be retrieved instead of the good one. Finally, the SINGER model has been tested with a single song at a time. It would be interesting to study its behavior when several melodies are learned.

References

[1] G. D. Sawa, "Oral transmission in arabic music, past and present," *Oral Tradition Journal*, vol. 4/1-2, pp. 254–265, 1989.

[2] A. Racette and I. Peretz, "Learning lyrics: to sing or not to sing?" *Memory & Cognition Journal*, vol. 35 (2), pp. 242–253, 2007.

[3] J. W. Stansell, "The use of music in learning languages: A review of the literature," University of Illinois at Urbana-Champaign, M.Ed., 2005.

[4] M. L. Huy, "The role of music in second language learning: A vietnamese perspective," in *Conference of the Australian Association for Research in Education and the New Zealand Association for Research in Education*, 1999.

[5] G. Mather, *Foundations of Perception*. Psychology Pr, 2006.

[6] C.-H. Chouard, *L'oreille musicienne. Les chemins de la musique de l'oreille au cerveau*. Gallimard, 2001.

[7] J. Plantinga and L. J. Trainor, "Memory for melody: infants use a relative pitch code," *Cognition Journal*, vol. 98, pp. 1–11, 2004.

[8] W. Gruhn and F. H. Rauscher, *The Neurobiology of Learning: New Approaches to Music Pedagogy Conclusions and Implications*, nova ed., ser. Neurosciences in Music Pedagogy, 2007, ch. 10, pp. 263–295.

[9] V. J. Williamson, A. D. Baddeley, and G. J. Hitch, "Music in working memory? examining the effect of pitch proximity on the recall performance of nonmusicians." *9th International Conference on Music Perception and Cognition*, pp. 1581–1590, 2006.

[10] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, "An integrated theory of the mind," *Psychological Review*, vol. 111, 136-1060, 2004.

[11] J. R. Anderson, N. A. Taatgen, and M. D. Byrne, "Learning to achieve perfect time sharing: Architectural implications of hazeltine, teague ivry (2002)," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 31, No. 4, pp. 749–761, 2005.

[12] M. D. Byrne, "Act-r/pm and menu selection: Applying a cognitive architecture to hci," *International Journal of Human-Computer Studies*, vol. 55, pp. 41–84, 2001.

[13] D. Bothell, "Act-r 6.0 reference manual," Carnegie Mellon University, Working Draft, 2004.

[14] J. R. Anderson, D. Bothell, C. Lebiere, and M. Matessa, "An integrated theory of list memory," *Journal of Memory and Language*, vol. 38, no. 4, pp. 341–380, 1998.

[15] I. Peretz and R. Zatorre, "Brain organization for music processing," *Annual Review of Psychology*, vol. 56, pp. 89–114, 2005.

[16] N. Gaab, C. Gaser, T. Zaehle, L. Jancke, and G. Schlaug, "Functional anatomy of pitch memory–an fmri study with sparse temporal sampling," *NeuroImage*, vol. 19, no. 4, p. 1417, 2003.

[17] U. Will, "Oral memory in australian song performance and the parry-kirk debate: a cognitive ethnomusicological perspective," *International Study Group on Music Archaeology*, vol. X, pp. 1–29, 2004.

[18] B. Chikhaoui and H. Pigot, "Analytical model based evaluation of human machine interfaces using cognitive modeling," *International Journal of Information Technology*, vol. 4, no. 4, pp. 252–261, 2008.

[19] B. E. John and D. D. Salvucci, "Multi-purpose prototypes for assessing user interfaces in pervasive computing systems," *IEEE pervasive computing*, vol. 4, no. 4, pp. 27–34, 2005.

[20] B. Chikhaoui and H. Pigot, "Simulation of a human machine interaction: Locate objects using a contextual assistant," in *procedeengs of the 1st International North American Simulation Technology Conference*, 2008, pp. 75–80.