MA ritgerð

Máltækni

When Strangers Meet

Collective Construction of Procedural Conversation in Embodied Conversational Agents

Stefán Ólafsson

Leiðbeinandi: Hannes Högni Vilhjálmsson
Aðstoðarleiðbeinandi: Eiríkur Rögnvaldsson

Maí 2015

HÁSKÓLI ÍSLANDS
HUGVÍSINDASVIÐ

ÍSLENSKU- OG MENNINGARDEILD

When Strangers Meet

Collaborative Construction of Procedural Conversation in Embodied Conversational Agents

Stefán Ólafsson
kt. 260985-2729

Lokaverkefni til MA gráðu Máltækni
Leiðbeinandi: Hannes Högni Vilhjálmsson (HR)
Aðstoðarleiðbeinandi: Eiríkur Rögnvaldsson (HÍ)

Íslensku- og menningardeild
Hugvísindasvið Háskóla Íslands

Í samstarfi við
Gervigreindarsetur Háskólans í Reykjavík

Maí 2015

When Strangers Meet: Collaborative Construction of Procedural Conversation in Embodied Conversational Agents

Ritgerð þessi er lokaverkefni til MA í Máltækni
og er óheimilt að afrita ritgerðina á nokkurn hátt nema með leyfi rétthafa.

Prentun: Háskólaprent
Reykjavík, Ísland, 2015

When Strangers Meet: Collaborative Construction of Procedural Conversation in Embodied Conversational Agents

Stefán Ólafsson

May 2015

Abstract

This thesis presents an application that adopts a novel method for software agents in an Icelandic language and culture training application to engage in conversation with human users and other agents. The approach allows the agents to procedurally select purpose specific conversation sections, within which they collectively construct discourse models that give rise to conversational behaviors. The theoretical foundations that make this possible are introduced, beginning with research that informs how humans conduct themselves during conversation, followed by the computational modelling of such interactions. This includes research conducted in the fields of discourse and conversation analysis, and observations from video recordings of how conversation between strangers in Icelandic unfolds. The computational modelling of communicative functions is achieved using parts of the Function Markup Language (FML) standard proposal and realized in a language learning virtual environment, involving use of Icelandic language technology. The general approach that was taken to the construction of the application is expounded, with an overview of the major components, followed by a sample run. A detailed outline of the implementation is followed by a discussion on its current capabilities is presented, as well as future work and a user study assessment proposal.

Þegar ókunnugir mætast:
samvinna vitvera við gerð samræðna í sýndarumhverfi

Stefán Ólafsson

Maí 2015

Útdráttur

Þessi ritgerð kynnir forrit sem beytir nýstárlegri aðferð til þess að veita sýndarvitverum getuna til þess að eiga í samræðum sín á milli eða við notendur í hugbúnaði sem hannaður er til kennslu á íslensku máli og menningartengdum þáttum. Aðferðin veitir vitverunum þann möguleika á keyrslutíma að velja einingar sem þjóna ákveðnum tilgangi í framvindu samræðnanna. Innan þessara samræðueininga byggja vitverurnar samræðulíkan í sameiginingu og samskiptamarkmið innan þess velda ákveðinni hegðun. Í fræðilega hluta ritgerðarinnar eru kynntir þeir þættir sem gera þetta mögulegt. Þeir eru rannsóknir á sviði orðræðu- og samræðugreiningar, sem skýra frá því hvernig mannfólk ber sig í samræðum, greining á myndbandsupptökum af því hvernig ókunnugir standa að samræðum á íslensku og kynning á reikniaðferð til myndunar samræðulíkana. Reiknilíkönin, sem samanstanda af samskiptamarkmiðum, eru mynduð með notkun hluta „Function Markup Language“ (FML) staðalsins og framkalla hegðun vitveranna í sýndarumhverfi. Máltæknihugbúnaður fyrir íslensku er nýttur í þessu samhengi. Aðferðinni sem var beitt við gerð forritsins er svo lýst, með yfirliti yfir helstu hluta þess, ásamt sýnikeyrslu á forritinu. Nákvæmari lýsing á innviðum forritsins og umræður varðandi gæði þess fylgir í kjölfarið og ritgerðinni lýkur svo með útlistingu á næstu skrefum og tillögu á notendakönnun.

Acknowledgements

There are many people to whom I owe a great deal of gratitude. First of all I am indebted to Eiríkur Rögnvaldsson, Hrafn Loftsson, and everyone involved with making the Language Technology program at the University of Iceland and Reykjavik University (RU) a possibility. The decision to join the program is easily one of the best choices I have ever made. I am especially grateful to Hannes Högni Vilhjálmsson for introducing me to the very exciting research conducted by the Socially Expressive Computing group (SECOM) at the Center for Analysis and Design of Intelligent Agents lab (CADIA) at RU. The opportunity to work on virtual agent software, getting to know the intricacies of modeling and implementing human conversational behavior has been truly inspirational and literally a dream come true. I also thank the awesome group of people I've had the pleasure of working with at CADIA: Elías, Hafdís, Valdi, Hossein, Siewart, Unnar and of course my predecessors Angelo Cafaro and Claudio Pedica.

I have to thank my fellow students in the Language Technology program: Alex Murphy, Katrín María Víðisdóttir and Hulda Óladóttir. Without each other we never would have made it out alive!

Most importantly I thank my family and in-laws. Without their unrelenting support on so many levels none of this would have been possible. I will forever be grateful to Agnes and Anna Guðrún in particular for always having my back, especially during the exceptionally hectic past two years. I cannot thank you enough, you are my everything.

Table of Contents

Abstract iv

Útdráttur v

Acknowledgements vi

List of Figures viii

List of Tables viii

1 Introduction 1

2 Theory 2

2.1 Discourse Analysis 3

2.2 Conversation Analysis 4

2.3 Communicative Functions 6

2.4 Observations of Strangers Meeting 6

2.5 Computational Modelling 7

2.6 Language Learning in Virtual Environments 8

2.7 Natural Language Support 10

3 Related Work 12

4 Approach 15

5 Implementation and Results 18

5.1 Sample Run 19

5.2 Implementation Details 20

5.3 Results and Performance 26

6 Discussion 27

6.1 Future Work 27

6.2 User Study Proposal 28

7 Conclusion 29

References 31

List of Figures

Figure 1. The ‘approach’ block 16
Figure 2. The AskInform block 17
Figure 3. The CloseInteraction block 18
Figure 4. IceLangVR ECA architecture 21
Figure 5. The FML chunks created by the Approach block’s Initiate method 25

List of Tables

Table 1. Suggested interactional function categories in FML 8

1 Introduction

This thesis describes the architecture and theoretical foundations of a system that allows software agents with conversation skills to procedurally select conversation sections during a conversation within a virtual environment. The proposed system is a component in a larger application being developed for language and culture training using the Icelandic language. This requires a look at some of the fundamental questions regarding the nature of conversation, discourse, language and interaction. By exploring research in the fields of linguistics, conversation and discourse analysis, virtual agent architecture, virtual environments and language learning, the grounds for implementing an application based on this research presents itself.

Software applications that are able to recognize speech, handwriting, images etc., have become commonplace with the advent of 'smart' devices, e.g. phones and tablets, within the past ten years. However, in order for an interaction between the human and the machine to occur, the machine's software itself must be able to respond in a manner that is in accordance with the type of interaction taking place. For example, an application like Apple's Siri involves a speech recognition interface capable of converting natural language speech to text, then processes the information and ultimately output a response using speech synthesis, i.e. converting its response to natural language using audio. However, the system that makes this particular interaction possible does not necessarily have conversational skills.

Applications such as Siri serve a particular purpose and are programmed to cope with certain domain specific requests. How would an interaction be different if such an application knew it was conversing with an acquaintance vs. a stranger? At what point does an interaction become conversational? These are the sorts of questions that this thesis deals with, particularly how to approach strangers and begin a conversation. Approaching a stranger in the street results in quite a different interaction than one where the person being approached either has knowledge of your arrival, expects you to approach, or knows you (see section 2.4).

The main contribution of the thesis is to introduce a dynamic way for participants, both human users and software agents, to shape and influence the conversation they partake in. To that end, a method was developed for allowing all participants to procedurally and collectively select what direction the conversation will take. This is accomplished by defining various kinds of sub-interactions, or *sections* (see section 2.2), that when strung together ultimately form the conversation as a whole. The sections themselves contain possible

discourse models that guide the interaction more restrictively, since not all kinds of conversation sections expect all manner of discourse to be carried out.

The motivation for this work is to enable users of a language learning application to engage in collaborative conversation with virtual agents to hone their language skills. The language learning application is an ongoing research project called *Icelandic Language and Culture Training in Virtual Reykjavík* [1] (IceLangVR) supervised by Hannes Högni Vilhjálmsson, leading the Socially Expressive Computing group (SECOM) at the Center for Analysis and Design of Intelligent Agents[2] (CADIA) at Reykjavik University (RU). The focus of this research project is to build an application that allows users to conduct interactions that are as fluid and natural as possible, supporting their acquisition of the new language, in this case Icelandic. Therefore, in addition to the novel approach to interacting with virtual agents, the resulting application will rely on various tools and databases made specifically for Icelandic to deal with natural language from the user and produce meaningful output that is in accordance with the learning objectives at any given time.

The thesis begins by introducing and defining the major concepts used throughout and laying down the theoretical foundations that it builds on. This is followed by a presentation of related work, then a description of the approach that was taken to address the problem, its implementation and results, and finally some discussion and a conclusion.

2 Theory

Research in various fields of study dealing with conversational behavior looks at human interaction from their perspective and define concepts in terms of their outlook and findings. Since many of these concepts are commonplace in English, such as ‘conversation‘, ‘agent’ and ‘discourse’, this section defines some these terms in order to distinguish them from their common and other research specific interpretations. Furthermore, this section deals with the various theoretical underpinnings that inform the work described in this thesis, and elaborates on these concepts.

Discourse and conversation analysis provide the means in which to break interactions apart and study their sub-components, thereby shedding light on the functions that underlie naturally occurring utterances. Research of such occurrences in Icelandic conversation when

[1] Grant from the Icelandic Research Fund
[2] http://cadia.ru.is/

strangers meet has been conducted, resulting in the development of a multimodal annotation scheme for use in computational modelling. The scheme draws on further research carried out for the advancement of embodied conversational agents (ECAs) and is applied to such an architecture within a language and culture training application. In addition to communicative functions, the ECAs being developed have access to various natural language processing (NLP) tools for Icelandic in order to allow users to engage in practical and genuine conversations.

2.1 Discourse Analysis

The field of Discourse Analysis (DA) has a varied focus, depending on the researcher's discipline. For instance, from the perspective of sociology, the aims of DA are different from that of psychology. In this thesis the focus is on DA from the perspective of Linguistic Pragmatics. This is the analysis of texts with the aim of describing *how* language is used within a particular context, thereby identifying the underlying function of utterances. By primarily analyzing texts, DA distinguishes itself from Conversation Analysis (CA) (see section 2.2) in at least this one respect.

Researchers have come to realize that individuals have knowledge of discourse structures, contained in a mental object called a *discourse model* (Johnson-Laird & Garnham, 1980, p. 371). As conversation progresses, a discourse model is shaped by the interacting parties; however, the speaker and the listener maintain seemingly separate discourse models:

> There is usually one context for the speaker[3] and another context for the listener … [T]he real context of an utterance consists of separate representations of the current conversation that the speaker and the listener create and maintain. (Johnson-Laird & Garnham, 1980, p. 374)

If this is in fact the case, how in the world do people conduct conversations at all? Only through shared knowledge of each other's discourse models can a speaker and listener engage in conversation. Although the speaker and listener may maintain distinct models, at least some part of each model is known to both parties and collectively crafted and kept. Each participant thus shares his model on the floor of interaction in order to allow conversation to progress.

[3] In DA, the individual who initiates a dyadic interaction is commonly referred to as 'speaker' while the other is the 'listener'.

2.2 Conversation Analysis

When humans engage in conversation with one another, the interaction is not merely an exchange of words; conversation takes place in multiple modes of interaction involving all of the senses. The field of Conversation Analysis (CA) seeks to study such interactions and detect patterns within them. CA was developed by Harvey Sacks and his colleagues Emanuel Schegloff and Gail Jefferson, arising from Sacks' lectures on the subject from 1964-1972 (Heritage, 2009, p. 302). Researchers in CA focus on naturally occurring interactions captured on video or audio recordings and meticulously annotate all exhibited behavior, from gestures to speech, in terms of the particular interaction at hand.

During the project's initial development, the focus was solely on discourse modelling. However, it soon became clear during the course of implementation that in order to procedurally allow participants to construct their own conversations, another layer of abstraction was needed to account for the type of conversation within which the discourse models would emerge. To that effect, the participants have the opportunity during an interaction to select what are known as *sections of conversation* and indicate the kind of conversation that is about to be had. Clark notes that these sections are "longer stretches of talk devoted to a single task, point of discussion, or subject matter" (Clark, 1996, p. 330). He states that discourse arises turn by turn, is managed locally, and that conversations are joint ventures shaped by all participants. In his view:

> People may have general goals on entering a conversation, but they cannot prepare specific plans to reach them. They must achieve what they do contribution by contribution. (Clark, 1996, p. 331)

Conversations are therefore emergent. Furthermore, Clark (1996) crucially identifies three *time periods* of conversation (p. 331):

1. Entry into conversation
2. Body of the conversation
3. Exit from the conversation

Although it might not be apparent at first glance, the entry and exit do constitute conversation sections in that they are devoted to at least a single task: initiating and closing the interaction, respectively. All time periods of conversation are governed by a multitude of factors, such as

the relations between the participants, their intentions and personality. At each point in time, it matters how the participants are related, what they intend to do and how personality affects their response to the other's behavior.

It is likely in most cases that the entry and exit to and from conversation are single conversation sections, dealing with initiation and closing. However, the body of the conversation is more often than not composed of multiple sections, ones that cannot be seen to form a whole beforehand. Let us assume that one participant wants some information from another. Example tasks, or goals, for the participating parties might be asking for information and giving information. In order to accomplish the information exchange, the participants are required to collaborate despite apparent differences in their task: one asks and the other informs.

A different way of framing this example is to see the asking and informing as part of the same conversation section devoted to the same task: a collaborative exchange consisting of asking and informing. Posing a question to another person puts them in the position of having to reply in some way, almost whether they like it or not. One can imagine that it was probably not their intent two seconds ago to inform anyone of anything, but the approaching individual who just posed the question "Where is the bathroom?" *must*, in some sense, be dealt with. The approaching individual had the intent to ask for the location of some place and can be seen to have imposed on the other the intent to reply.

The participants actively influence each other's communicative intents. Entering a conversation neither party could have foreseen whether they would end up giving a reply to a request, voicing an opinion or perhaps raising a question of their own. Of course the situation could arise that the participants cannot partake in the interaction at the very start or realizes at some point that she cannot continue. The reasons for discontinuation may be vast, anything from temporal (a scheduled event) to natural (hunger), but in any case these situations are mostly unforeseen and emerge in the progress of interactions.

The idea that conversation is by nature emergent, from the perspective of CA, shares a connection with the DA notion of speaker and listener discourse models. The discourse models are influenced by how the conversation unfolds, which in turn is shaped by intent (Johnson-Laird & Garnham, 1980, p. 382). Thus, a participant's intents are molded by the actions of the others on the floor of interaction as the shared discourse model emerges over the course of the exchange.

2.3 Communicative Functions

The topic of interest within DA, for the purposes of this thesis, is the elements of interaction during discourse. These elements, or *communicative functions*, are the intentions that guide behavior. They are defined as sets of tags within larger clusters encompassing deictic, speech acts, turn management, grounding, etc. Elements that have a particular function in the discourse are the pieces that make a discourse model. Identifying and annotating discourse functions in texts is the main thrust of DA and researchers have constructed theories surrounding the usage of the particular aspects, or function types, such as theories of grounding, turn-taking and speech acts.

For each of these theories, sets of communicative functions are defined in accordance with the discipline involved and may even be researcher specific. The functions are gathered in a multimodal (MM) corpus, of which a variety exists, each one designed with a specific scenario and structure in mind (Abuczki & Ghazaleh, 2013). The MM corpus used for the work in this thesis is introduced in the next section.

2.4 Observations of Strangers Meeting

Kendon in his work identified various stages of interaction, including approaches (Kendon, 1990). His work provides detailed insight into how people who are either friends or acquaintances approach and begin conversation with one another. As a result he identified various rituals performed within a segment of the interaction called a *greeting phase*. However, in stranger-to-stranger interaction the rituals of the greeting phase do not appear. This called for the analysis of what conversational behavior and discourse functions are actually exhibited in an interaction between strangers in Icelandic specifically.

The ongoing PhD work of Branislav Bédi at the University of Iceland has produced multiple video recordings of a stranger engaging another and asking for directions to a specific place (Bédi, 2015a). In his work, Bédi focuses on identifying a particular discourse function in such scenarios called a *clarification request* and in the process has amassed a multimodal annotation scheme (Bédi, 2015b). The MM corpus includes the discourse functions and behavior observed and paints a picture of the intricacies of stranger-to-stranger interaction in Icelandic.

Interestingly, the research shows a lack of greeting phase in all cases. A majority of the videos showed that the interactions are initiated by an explicit announcement. This *explicit*

announcement of presence (EAP) involves calling attention to oneself in order to initiate the approach (Kendon, 1990, p. 172). Out of 43 videos in the video corpus where a stranger approaches another, 34 showed an example of EAP. This prompted the inclusion of such a discourse function in the implementation of the application (see table 1).

The following section introduces a framework for computationally modelling communicative functions and realizing them as behavior.

2.5 Computational Modelling

The sets of functions used in this project's development are a subset of the previously noted MM corpus, which arose from the observations detailed above in addition to the efforts of an international research community. This community developed the SAIBA[4] framework for the generation of multimodal behavior in ECAs. Their aims are to unify the representation of intent planning and behavior planning to allow researchers working on ECAs to easily share their tools and findings. They define three stages of multimodal generation (Kopp et al., 2006):

1. Planning of a communicative intent
2. Planning of a multimodal realization of this intent
3. Realization of planned behaviors

The SAIBA framework community defines two representation languages that allow transition from stage (1) to (2) and from (2) to (3), these are the Function Markup Language (FML) and the Behavior Markup Language (BML), respectively. The BML standard version 1.0 has been published, while FML remains a standard proposal (Cafaro et al., 2014). Since this thesis deals with the discourse models based on intent and function of actions, it is from within this standard proposal and the MM corpus that the majority of the communicative functions used in this project are derived.

The FML standard proposal presents a unified specification. Some key terms are:

Participant – A virtual agent or user participating in an interaction and carrying out or being affected by communicative functions.

[4] Situation, Agent, Intention, Behavior and Animation

Floor – An interaction that a participant can be engaged in with others. A metaphor for the social contract that binds participants together.

FML chunk – The smallest unit of FML functions associated with a single participant and ready to be converted to BML-specific behavior.

An FML document instance is divided into a *declaration* and a *body*. The declaration contains static information, including the participants' gender and personality, and dynamic information, such as the participants on each floor. The body of the instance is divided into three *tracks*, which reflect the categorization of the communicative functions, namely *interactional*, *performative* and *mental state*. Additionally, each chunk has a temporal setting to synchronize their execution across the different tracks. Table 1 shows the suggested function categories and types of the interactional track tailored to those used within this project.

Track Type	***Function Category***	***Type***
Interactional	Initiate*	*react, initiate*
	Closing	*break-away, farewell*
	Turn-taking	*take, give, keep, request, accept*
	Speech-act*	*eap, inform, ask, request*
	Grounding*	*request-ack, ack, clarification-request, cancel*

Table 1. Suggested interactional function categories in FML and their types used in this project. The functions of the categories marked * have been altered from the original proposal. (Cafaro et al., 2014, p. 88)

In the standard proposal, the *Initiate* category also includes the types *recognize*, *salute-distant* and *salute-close*. These are greeting phase specific function types and are not dealt with here. The *Speech-act* and *Grounding* categories were expanded to include *eap* and *clarification-request*, added from the observed data in the MM corpus.

The next section discusses the merger of language learning with virtual environments and defines a few key concepts.

2.6 Language Learning in Virtual Environments

The thesis proposes an agent architecture and a conversational system for use within a language and culture training application reminiscent of a modern video game. The

interactions between the user (player) and the non-player characters (NPCs) takes place in a virtual 3D environment (VE).

In VEs, immersion enables *presence*:

> We distinguish between immersion and presence. *Immersion* is a description of a technology, and describes the extent to which the computer displays are capable of delivering an inclusive, extensive, surrounding and vivid illusion of reality to the senses of a human participant … Presence is a state of consciousness, the (psychological) sense of being in the virtual environment. (Slater & Wilbur, 1997, pp. 3-4)

The design of the overall environmental aspects, such as buildings, foliage, sounds, etc., is important to the sense of presence. No less important is the design of the NPCs or *intelligent virtual agents* (IVAs). The annual *International Conference on Intelligent Virtual Agents* defines IVAs thus:

> Intelligent virtual agents (IVAs) are interactive characters that exhibit human-like qualities and communicate with humans or with each other using natural human modalities such as facial expressions, speech and gesture. They are capable of real-time perception, cognition and action that allows them to participate in dynamic social environments. ("Fifteenth International … " , 2015)

The classical Artificial Intelligence definition of an *agent* is any software that uses sensors to perceive its environment and effectors that act upon that environment (Russell & Norvig, 2003, p. 31). An IVA is therefore software that builds on this definition with the addition of the "human-like" and "human modalities" aspects, as explored in the field of Human-Computer Interaction (HCI).

A further development of virtual agents, and the type of software agent this project employs, are *embodied conversational agents* (ECAs). Justine Cassell et al. define ECAs in terms of what they are not: only a computer interface represented by a human or animal body. Rather, ECAs are specifically conversational, exhibiting and recognizing the behavior involved during human face-to-face conversation (Justine Cassell et al., 2000, p. 29). This definition distinguishes ECAs from IVAs, which do not exhibit conversational behavior specifically.

The goal is thus to create a VE that adopt aspects of immersion to fuel presence in the user with realistic environment modelling and ECAs that can engage in natural and dynamic conversations. Striving for enabling presence in the VE is allied with immersion in the language learning sense of the term. In learning a second language, key aspects of immersion include exposing the students to the target language in the classroom and instructing them using the target language (Cummins, 1998, p. 2). Therefore, in the process of invoking a sense of presence in the users of language learning using VEs, immersion with respect to the target language should not be overlooked.

Language learning applications using ECAs strive to combine game techniques with intelligent feedback (W.L. Johnson et al., 2004). These two factors depend on lesson planning, which hinges on the overall purpose of the application, its target users and objectives. How these factors are addressed with respect to this project is discussed in section 3.

The ensuing section discusses the state of Icelandic Language Technology and introduces the speech recognition and synthesis employed by IceLangVR.

2.7 Natural Language Support

A particularly important feature of language learning software is enabling users to interact with the application using their voice. This requires the integration of a speech recognition system capable of converting Icelandic speech to text as input into the application for further conversational processing. Additionally, speech synthesis technologies (text-to-speech) are immensely beneficial in language learning applications that require the system to provide frequent natural language feedback to the user. Providing the NPCs with a voice of their own using Icelandic speech synthesis eliminates the necessity of prerecorded replies. Up until recently both of these technologies were severely lacking in quality, with respect to Icelandic; however, recent developments have allowed for the integration of conversational speech recognition and synthesis, accomplished for the first time in this project.

The status of Icelandic language technology (LT) in general is quite poor in comparison to other European languages, in fact the situation is only reasonably good with respect to basic LT tools and resources (Rögnvaldsson et al., 2012, p. 33). Due to the efforts of The Icelandic Centre for Language Technology[5] (ICLT) and Reykjavik University, a speech recognition system was developed for Icelandic in collaboration with Google. The

[5] Máltæknisetur - http://www.maltaeknisetur.is/

undertaking was accomplished through the *Almannarómur*[6] project, which collected data for an Icelandic speech corpus in 2011 and 2012. Its main aim was to create an open source speech project to enable research and development, and was well suited for acoustic modelling for speech recognition (Guðnason et al., 2012). The Google speech recognition for Icelandic was made available in late 2012, accessible to users of Android devices and the Chrome browser. In the HTML5 Speech Recognition API, JavaScript is allowed access to the browser's audio stream for conversion into text. Unfortunately this affects the usability of applications if speech recognition is to be employed, limiting them to web builds only. For this reason, and others, the Icelandic LT community continues to call for the support of ventures like *Almannarómur* in order to further the development of stand-alone open source speech recognition software for Icelandic.

The current state-of-the-art in speech synthesis for Icelandic are the voices of *Karl* and *Dóra*. This software was created by IVONA[7] for the Icelandic Organization of the Visually Impaired (BIOVI)[8] through the collaborative efforts of various organizations in the private and public sector, as well as academic institutions[9]. Due to the role that CADIA played in the creation of the speech synthesizer, the SECOM group was granted access to IVONA's services. This enabled the use of synthesized voices within IceLangVR, opening a window of opportunity for creative and unique agent dialog.

Section 6.1 discusses the incorporation of dialog management into IceLangVR. To allow for this, the system must make use of the basic, Icelandic specific, LT tools and resources available. The first of these is the IceNLP[10] toolkit, which applies sentence segmentation, tokenization, POS tagging and parsing (Loftsson & Rögnvaldsson, 2007). This tool is fundamental to any further natural language processing (NLP) performed by the system, e.g. assessing the grammatical correctness of student input, semantic analysis, basic natural language generation (NLG) on the part of the agents, etc. Basic semantic analysis is also a possibility with the use of *Íslenskt Orðanet*[11] (i.e. Icelandic Wordnet), a searchable database of semantic relations between both single- and multi-word concepts in Icelandic (Jónsson, 2012).

[6] http://almannaromur.is/
[7] http://www.ivona.com/
[8] Blindrafélaið - http://www.blind.is/
[9] http://www.blind.is/verkefni/talgervlaverkefnid/
[10] http://nlp.cs.ru.is/icenlp/
[11] http://ordanet.is/

3 Related Work

REA

The ECA architecture in IceLangVR is based on ECA research conducted at MIT's Media Lab at the turn of the century, drawing inspiration from the *REA* project in particular. With REA (J. Cassell et al., 1999) the authors pushed for realizing the – up until then – metaphor of conversation between humans and computers. To that end, they created an embodied and conversational agent that could hold up her end of the conversation by taking the user's speech and gestures as input and understand them in terms of the conversational functions they fulfill. REA stands for "Real Estate Agent" and was implemented as a humanoid displayed on a large projection screen in front of the user. Examples of behaviors she exhibited, while engaging in conversation with the user concerning real estate, are gesturing towards images of properties that appear next to her on the screen, changing eye- and head-gaze, head-nod and various facial expressions. REA therefore had a model of conversational function as the means for behavior to be generated and understood in the context of the current conversation, which influenced the design of the ECAs within IceLangVR.

TLTS

The Tactical Language Training System (TLTS) was developed by researchers at the University of Southern California (USC) for training learners in spoken Arabic and other languages (W.L. Johnson et al., 2004). The learners engage in missions that are parts of an interactive story and interact with animated characters, all the while receiving feedback and assessment from an intelligent agent coach. The authors hypothesized that the combination of intelligent feedback and gaming techniques would be motivating and lead to more rapid skill acquisition. The results from evaluating learners with limited skills showed that the system proved more effective than the traditional classroom setting. This work has evolved into the Tactical Language and Culture Training Systems (TLCTS) products of the company *Alelo*[12] and has informed the development of IceLangVR in terms of how the learner is allowed to engage in free spoken conversation, as opposed to confined to reading on-screen text (W. Lewis Johnson & Valente, 2008, p. 74). The idea of the learner playing a part in an interactive story, the lesson planning and learner assessment have also been informative to the development of IceLangVR.

[12] http://www.alelo.com/

IceLangVR

The application described in this thesis is a contribution to the ongoing Icelandic Research Fund supported project *Icelandic Language and Culture Training in Virtual Reykjavík* (IceLangVR). The principal aim of IceLangVR is to join "the international forefront of computer aided language instruction research as well as producing pioneering work in the serious games for education field" by using the latest in language and culture training technology, incorporating interactive language learning materials and language processing tools for Icelandic, and populating a 3D virtual environment modelled on downtown Reykjavík with agents that exhibit social behavior in reaction to the user (Vilhjálmsson, 2013, p. 6).

To that end, the developers of IceLangVR look to build on the success of Alelo's TLCTS by infusing expertise in teaching Icelandic as a second language and foreign languages and cultures. This role is filled by *Icelandic Online*[13] with its innovative learning methods targeted specifically at students learning Icelandic as a second language (Arnbjörnsdóttir, 2008). Its lessons plan structure informs the subject matter and dialog for the scenes in relation to the language learning aspects of the game.

Since IceLangVR's inception, the Unity3D [14] game engine has been used for development. The development platform is free to use and the current version (Unity 5.0) allows scripting in two programming languages, i.e. C# and UnityScript[15] ("Creating and Using Scripts," 2015). C# was chosen for IceLangVR as it was considered to be more robust, optimized and provided more features that the other. The 3D modelling work necessary for realizing downtown Reykjavík as a virtual environment was done with support from Páll H. Pálsson and Borgarmynd[16].

The architecture and social behavior of the ECAs in IceLangVR project has evolved over the past few years. Various technologies developed in multiple projects have added to their conversational skills in different ways, whether directly in relation to IceLangVR or not. The CADIA *Populus* social simulation platform (Pedica, 2009), which later became the *Impulsion* behavior engine, was the foundation for handling reactive social and group

[13] http://icelandiconline.is/index.html
[14] https://unity3d.com/unity
[15] Modelled after *JavaScript*
[16] http://borgarmynd.com/

autonomous behavior. The Impulsion engine was rewritten in 2013 in C# for use within Unity3D and IceLangVR.

An important development in the conversational skills of IceLangVR's ECAs was Angelo Cafaro's PhD work on *greeting agents* (Cafaro, 2014). His thesis outlined a contribution to the development of FML (see section 2.5 above) culminating in a standard proposal for the language (Cafaro et al., 2014).

These developments in autonomous social behavior and proposed standardized framework for communicative functions have paved the way for the recent advancements within the IceLangVR project. The first notable feature was the incorporation of speech recognition for Icelandic, allowing the input of natural language. Since the speech recognition was only available through the Chrome browser (see discussion in 2.7) all application demonstration builds had to be for the web, which was fortunately a simple task since web-builds are a feature of Unity3D. The addition of speech recognition to the project was implemented by Stefán Ólafsson and Elías Ingi Björgvinsson (SECOM) and first successfully demonstrated in a talk given by Hannes Högni Vilhjálmsson and Branislav Bédi at the *Nordic Seminar on IT in Language Teaching* held by the University of Iceland in February 2014. The second notable feature was the making of the ECA *Brain* architecture, developed by Jonas Braier during his involvement with SECOM in the spring of 2014. Although the current Brain structure (figure 5) has changed from the original implementation, most of the original components are still in use.

The next features developed for IceLangVR were the modelling of non-verbal dominance and submissiveness in virtual agents[17] in research conducted by Hafdís Erla Helgadóttir (SECOM) and social interaction management using parts of the FML standard proposal (see section 2.5) conducted by Stefán Ólafsson (SECOM). These features were worked on during the summer of 2014, culminating in a workshop paper (H. H. Vilhjálmsson et al., 2014), poster and demonstration (H. Vilhjálmsson et al., 2014) at IVA 2014 in Boston. This allowed for modelling agent interaction based on personality features using communicative functions.

An addition to the project that is currently under development is the IceLangVR Scripting Engine, made by Siewart van Wingerden (SECOM). The engine allows for the design of new environments with a set of goals and is made up of multiple components that handle various tasks, such as creating new agents and an events system. The event system

[17] http://www.havethis.info/poster

will allow for the scheduling of specific tasks to be performed by the agents and integration of lesson planning for the language learning component, directing the delegation of assignments for the user.

The following section outlines the approach taken to providing the means for the agents to frame their conversation within task specific sections.

4 Approach

The purpose of procedurally selecting conversation sections is to allow the participants of the conversation to affect its course. There are a two main reasons for doing this. Firstly, the attempt is to maintain presence, the authenticity of the interaction, by giving the user a feeling of natural conversation that they are a part of. Secondly, this provides the user as a language learning student the opportunity to repeatedly engage in the same kind of conversation, for example to ask questions however many times they feel is needed. Since repetition is an integral part of language learning, the importance of this point should not be overlooked. The computer does not lose patience with the student.

Clark's *conversation sections* are central in the approach to this problem. In the code, the conversation sections are called *blocks* and contain methods that drive the interaction forward. The idea is to segment an interaction into purpose specific components. Examples of such blocks are the initiation of the interaction (approach), ending the interaction (closing), and any other purpose specific segment occurring in between, e.g. asking for directions. In an approach for instance, particular conversational functions have been identified as reoccurring elements (see section 2.5). It is therefore possible to construct mechanisms that produce discourse models based on these observations, ones that can be used regardless of what kind of conversation will follow. Only when the interaction is over is it possible to realize what course the conversation took, its topics and so forth, by analyzing the blocks in hindsight. Natural language conversation between people is an emergent activity that can only loosely be determined beforehand (Clark, 1996, p. 331).

They key to accomplishing procedural and collaborative creation of discourse models by all participants at certain moments in the conversation is the fusion of blocks and discourse functions. Together they allow participants to create discourse models by taking note of factors such as their personalities, intentions, relations, etc. Moreover, the blocks that allow entry and exit into and from the conversation have certain features that distinguish them from

blocks that would appear in the body of the conversation, in that they contain the discourse functions that are of a more ritualistic nature. In contrast, the blocks that form the body of the conversation have a more varied kind of discourse functions, ones that are appropriate for the particular task of the section. That being said, although it is not as rigorous as the approach or closing, blocks that appear in conversation's body still have a structure that can be realized.

Going back to the example of an approach, there are factors that define what kind of an approach is to be made. One of the more obvious factors that matters a great deal is relations between agents, particularly whether the individual being approached is a stranger or not. It is quite apparent from observational data on strangers (see section 2.4) that the behavior exhibited by the participant making the approach is quite different from the approach behavior of acquaintances and friends as described by Kendon (Kendon, 1990). In the latter, there is an elaborate greeting phase that takes place in various stages, while no such phase is present in the former. The video annotation work provided the basis for deciding important systematic development choices and the structures of the blocks and the methods within them were determined by these observations.

Each state within a block contains methods that perform specific tasks that progress the interaction along a non-predetermined path. At every state the participants' intents with respect to the conversation may change, influenced by the emergent actions of others.

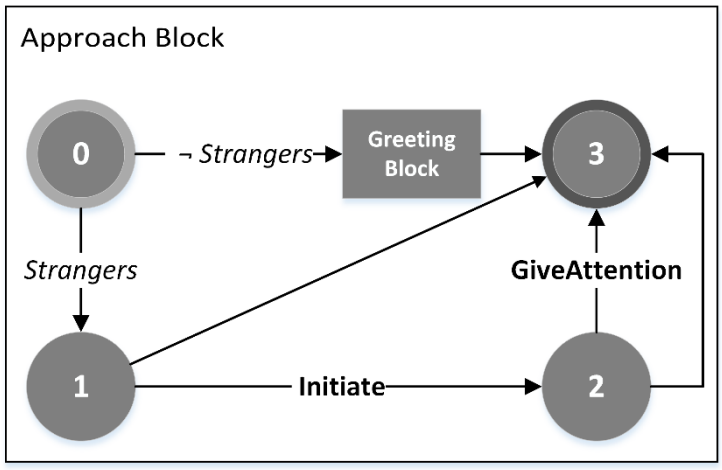

Figure 1. The 'approach' block's state machine propels the conversation using methods (Initiate and GiveAttention, shown in bold) that piece together the discourse functions relative to the agents' intent. The initial state checks for relations and moves to either a greeting phase or a 'stranger specific' initiation of conversation. States (1) and (2) allow for 'inaction', resulting in the approach coming to an abrupt end in the final state (3).

The actions taken in a state are methods (shown in bold in figure 1) that produce FML bodies that contain the discourse functions relevant to those situations in the conversation. If action

is taken in state (2) the agents form a group, signifying the beginning of conversation proper and binding the participants together in the physical space. In some cases, inaction is relevant, such as when an agent's intent changes in mid-conversation, resulting in the approach coming to a halt and possibly the conversation as a whole.

After going through an approach a body block may be selected with respect to the agents' intent and personality traits. One such body block allows the agents to progress through states where they can ask for and receive information, shown in figure 2.

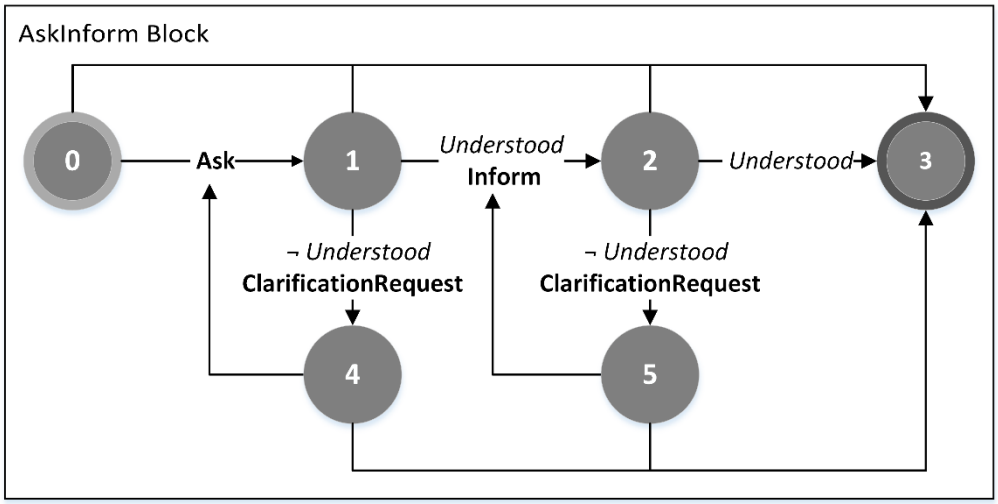

Figure 2. The AskInform block's state machine progresses through the states, adding discourse functions relevant to the agents' intent. The methods that produce the functions are shown in bold: Ask, Inform and ClarificationRequest. The italicized text is the outcome of a truth check for whether the agents' utterances were understood by the other.

From the initial state, the agent who has the turn may act by performing the *Ask* action leading to state (1) or not act leading to the end state (3). In state (1) the agent whose turn it is checks whether she 'understands' the dialog produced by the other agent. This involves using NLP tools to validate certain features of the dialog in question, e.g. grammatical correctness, semantic coherence, etc. If the sentence is not understood, the agent performs a *ClarificationRequest* action, leading to state (4) and may involve the agent simply producing a sound of bewilderment or to ask the other to repeat herself. In either case, she requests of the other to *clarify* the last utterance. This allows the agent who first performed the *Ask* action to ask again, as figure 2 shows. The same applies to the transition from state (1) to (2), except the agent who asked will not be the one who informs.

When the body of a conversation has been completed, the interaction may be closed. Figure 3 shows a state machine that creates the FML bodies necessary for realizing the behavior associated with such a conversation section.

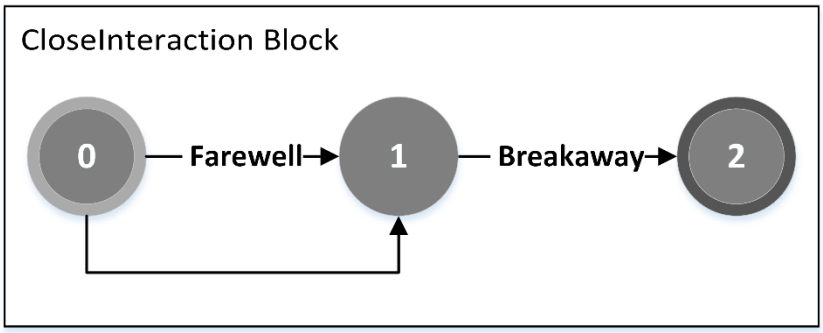

Figure 3. The CloseInteraction block's state machine. The methods, in bold, create the communicative functions that underlie the behavior associated with closing an interaction. In some circumstances farewells are not appropriate, resulting in an immediate break away from the interaction.

In addition to the methods that allow for the creation of FML bodies to be realized as behaviors, the closing block removes any extraneous components relating to the interaction. It became apparent after observing the video data (see section 2.4) that the participants actually begin moving away from one other in the middle of thanking or saying farewell. Therefore, in order to reflect these observations, the initial state (0) takes care of removing the group that bound the agents in conversation.

After the creation of FML bodies in the methods shown in each of the figures above, they are combined with the static information attributed to each interaction, namely the floor and its participants, to form an FML instance. At that point the instance can be sent to any component that is made to plan and realize behavior given communicative functions.

The next section shows a sample run of the application, followed by details on the implementation and results.

5 Implementation and Results

This section begins with a sample run of the application providing an example output given two agents with particular dominance and intent settings. Both agents in the example are NPCs; however, as revealed in section 5.2, the interaction would not be much different if the user was involved. Following the sample run is a detailed description of the components in the code that make the interaction possible and comments on the results and performance.

5.1 Sample Run

Two agents, Peter and Bjorn, are instantiated in the vicinity of one another in a virtual environment. Peter has a high dominance setting and an intent to get information about the location of a specific place in downtown Reykjavík. Bjorn on the other hand has low dominance and no specific intent. When Peter moves closer to Bjorn, their respective perception systems perceive the other and their reasoning faculties check their intentions and decide whether to act on them. While Bjorn has no interest in initiating a conversation, Peter's intent for getting the information prompts him to begin an interaction with Bjorn. A discourse manager is initiated and a floor of interaction created with Peter and Bjorn as participants. The floor's state is changed to an executing setting as it seeks to call for the next action from the current block, but finds that no current block is available. The first block is therefore established by looking at both participants' intentions and dominance levels and in this case an approach block is selected.

In the block's initial state, Peter and Bjorn's relationship is checked and as it turns out they are strangers. The state of the floor becomes 'ready', the discourse manager calls for its current block's next action and the floor's state is switched to execution once again. Inside the block, the participants are now in state 1 where Peter creates an FML body with the following communicative functions: react, take-turn, request acknowledge-ment, explicit announcement of presence, an utterance of "Afsakið" (i.e. *excuse me*) and give-turn. Bjorn adds to the FML body a react and a grounding acknowledgement function. The FML body is turned into an FML instance and sent to each agent for behavior generation of their respective functions.

The floor state changes and the next action is called for as before and Peter and Bjorn find themselves in state 2 of the approach. Now a group object is created in the virtual environment that binds the participants physically in the conversation. Bjorn now makes the following functions: take-turn, request acknowledgement, acknowledgement speech act, an utterance of "Já" (i.e. *yes*) and give-turn. Peter contributes to this FML body with a grounding acknowledgement. The FML body is then dealt with and the floor state changes in the same manner as described above. However, now the approach block has reached an end state and the floor must select a new block.

By evaluating the participants' dominance and intent, it is determined that Peter's block suggestion is appropriate and a block for asking for and giving information (AskInform as in figure 2) becomes the current block. Peter now in the initial state makes the functions: take-

turn, request acknowledgement, ask speech act, an utterance of "Hvar er Hitt Húsið?" (i.e. *Where is Hitt Húsið?*) and give-turn. Bjorn adds a grounding acknowledgement function. The functions are sent to be generated and the block state becomes progresses to state 1. Now Bjorn takes the turn, requests acknowledgement, has a speech act of inform, makes an utterance of "Það er á horni Pósthússtrætis og Austurstrætis" (i.e. *It is on the corner of Pósthússtræti and Austurstræti*) and gives the turn. Peter grounds with an acknowledgement. Now the AskInform block has reached an end and Peter's intent calls for a block that closes the interaction.

The closing block begins by eliminating the group that bound them together. Now Peter makes the same functions as before, except this time there is a farewell function and the speech act is a ritual with an utterance of "Takk" (i.e. *Thanks*). The nature of the ritual speech act ensures that Bjorn replies in kind and the system makes sure that Bjorn's functions are realized right after Peter's acts. The closing block concludes with a break-away function performed by both participants and now neither participant has the intention to continue with the conversation. Therefore, the floor informs the discourse manager that the interaction is over.

This concludes the sample run of the application and the next section will provide details concerning the implementation that make the sample run possible.

5.2 Implementation Details

The following is a detailed description of the individual components that make the output shown in the sample run possible. This includes implemented classes and relevant variables that provide the functionality for conducting an interaction.

Agent

The IceLangVR ECAs have a multitude of components necessary for conducting an interaction, most important of which are `Impulsion` and the `Brain`. The Impulsion library and classes are the basis for coordinating agents in a group and handling instinctive behavior. The IceLangVR agents inherit from the `ImpulsionAgent` base class. This allows the agents to take full advantage of Impulsion's functionality, such as to perceive other agents in their environment and form social groups for interactions such as conversation.

An important feature introduced in the development of the IceLangVR Scriptable Engine, was the ability for the user to "possess" or take over the controls of an otherwise

normal IceLangVR agent. This is a result of the design principle that the user's embodiment in the virtual environment should not be a traditional puppet-like avatar that is vastly different from any other virtual human in the scene. The user thus takes control of the agent's motor skills, controlling the walking animation with the keyboard and head-look with the mouse. As will become evident from the text that follows, there is a need for the agents to make their intents known. For the user, this is a matter of pressing a button on the keyboard, e.g. 'T' for 'talk', which changes the `IntentState` variable in the agent. Thus, when the user's agent approaches another, an interaction much like the one described above may take place, though the dialog is managed differently, as described below.

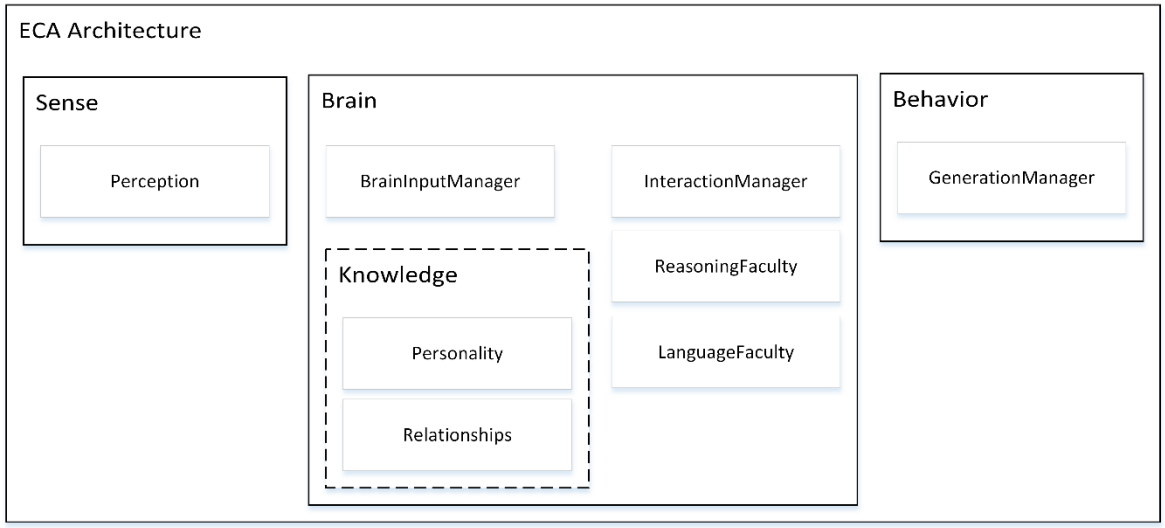

Figure 4. IceLangVR ECA architecture. The *Perception* component is as the agent's sensor and communicates with the *BrainInputManager*. The Brain's components work together with the discourse system in making communicative functions to be sent to the *GenerationManager* for behavior realization.

Brain

The user possessing an agent approaches another agent. Both of the agents perceive one another via Impulsion's `Perception` script and may perceive multiple agents in their "neighborhood". However, only the closest one is relayed by `Perception` to the agents' `BrainInputManager` (BIM), which has several functions that handle different kinds of perception. The only functioning perception mechanism for the time being is the means of detecting the closest agent. This object is then sent for interpretation to the Brain's `ReasoningFaculty` (RF). An agent's RF interprets the other agent's proximity in a particular way; at a certain distance (3.6 meters) the agents are within one another's social zone (Cafaro, 2014, p. 21). There are other conditions that are checked at this point. The first

is the `InteractionManager`'s (IM) `SituationState` and the second is its `IntentState`. To ensure that the agent who has the intent to interact will be the one who initiates the interaction going forward, the `SituationState` is set to `None` and the `IntentState` is set to `GetInfo`. If the other agent does not intend to interact, she gets the opportunity to opt out after the attempt to initiate has occurred. At this point the RF changes the `SituationState` to `Initiation`, ensuring that the agent initiating an interaction does not engage right away with any other agent that might enter its social zone. Most importantly, the initiating agent's RF now calls the IM's `StartInteraction` method and passes the other agent as a parameter. `StartInteraction` calls the `CreateFloor` method in the singleton class `DiscourseManager` (DM), passing as a parameter a list of the two agents involved in the interaction.

DiscourseManager

Now the discourse system takes over. The DM serves as the keeper of all floors in the environment, allowing for multiple interactions occurring anywhere in the scene. It is implemented as a singleton class allowing any agent to call the `CreateFloor` method without creating new instances of it just for that particular floor alone. This setup ensures the progression of all interactions taking place on floors of interaction. The DM's `CreateFloor` method creates a new `Floor` and adds the two agents[18] to it as `Participants`. The floor is then added to a list of active floors in the DM. The DM calls a method called `CheckActiveFloors` every 500 milliseconds. This method makes sure that the DM actually has at least one active floor and then goes through the list of active floors, checks if their `FloorState` enumeration is set to `Ready`, and then calls their `ExecuteNextAction` method.

Floors and Block Selection

The `Floor` class contains methods and variables needed for containing and progressing a particular interaction. Some important variables include lists of `Participant` objects, past blocks and `FmlInstance` objects, an integer `id` and a simple dialog history implemented as a list of string `KeyValuePair` objects. The `ExecuteNextAction` method, mentioned above, is key to an interaction's progression. Once called, the `FloorState` is changed to

[18] The system implementation will be extended to support multi-agent interactions.

Executing, barring the DM's CheckActiveFloors from calling the ExecuteNextAction method while the particulars of any given point in the interaction are being worked out. Then the NextAction method in the current Block instance is called, passing a list of the participants as a parameter and returning either a true or false value depending on whether the block's state machine has reached an end state or not. If it returns true, it has finished its execution and therefore it's safe to change the FloorState back to Ready. If it returns false, however, it means that the internal state machine of the current block has reached an end state and the participants must collaboratively select a new block for the interaction to continue.

The Floor's DecideNextBlock method is called and either returns null or a new block. If the new block is returned it is added to a list of past blocks for that floor and the FloorState is thus switched to Ready once again, allowing the DM to propel the interaction forward in the same manner as described above. If the DecideNextBlock returns null, however, the interaction as a whole is over, the floor's execution is halted, its FloorState is switched to Finished and the DM adds the floor to a list of past floors.

The selection of a new block can be implemented in numerous ways. The approach here was to make use of the participant's personality, namely their dominance. The agent with the highest dominance personality setting decides the next block. To begin the interaction, the Approach block must be selected. As before, the ExecuteNextAction method calls the DecideNextBlock when it finds there is no current block to call. Within DecideNextBlock, the number of past blocks is checked. If there are none then, if there is a participant with a SituationState of Initiation, the Approach block is returned and the participant's SituationState is changed to Conversation. In order to accomplish the selection of consecutive blocks, the DecideNextBlock method wades through the participants for the one with the highest dominance setting. That participant's ReasoningFaculty contains a BlockSuggestion method which is called at this point.

The BlockSuggestion method looks at the particular agent's intents in order to return her block suggestion to the floor, since specific intentions call for particular blocks. For example, if an agent's intent is to get information, i.e. has the IntentState set to GetInfo, the method switches the IntentState to Ask and then returns a block to the floor that will allow the intent to be realized, i.e. an AskInform block. If the agent has a particular intent, like the intent to go somewhere, the IntentState would be set to Go. In that case the method makes no changes to her intent and returns a block to the floor that will

end the conversation, i.e. a `CloseInteraction` block. If the `IntentState` has no specific setting, the function returns null as an indicator that the interaction is over. The `BlockSuggestion` method, therefore, returns a suggested block from each agent to the floor that is appropriate given their intents.

Blocks

As previously mentioned, the `ExecuteNextAction` method in the `Floor` class calls the `NextAction` method in the current block. Each block inherits from a super-class called `Block` which contains the definition for the abstract method `NextAction`, which takes a list of participants as an argument. The `Block` class also contains a list variable of `FmlBody` objects and maintains a count of the number of `FmlBody` objects it has created. As mentioned above, the classes that inherit `Block` are `Approach`, `AskInform` and `CloseInteraction`. These sub-classes implement the abstract `NextAction` function, which encapsulates a state-machine that progresses the interaction forward. The implementation allows for the creation of more blocks.

Each state performs a specific task. For the purposes of providing details on how the state machine of a block is traversed, the following example follows that of the `Approach` block (figure 2). In the initial state (0), the relationships of the participants are checked to see whether they are strangers or not. If they are not strangers, they are either categorized as acquaintances or friends and the interaction moves to a `Greeting`, a sub-block within the approach. The intricacies of greetings have not been implemented for this project and will therefore not be detailed here. In the case where they are in fact strangers, the interaction is moved to state (1) where the participant with the intent to `GetInfo` initiates the conversation. Before going into details concerning the `Initiate` method that makes this possible, if the intent of the initiating participant has changed to a value such as `Go`, by some outside force such as the IceLangVR Scriptable Engine's events system, the initiating participant moves the interaction to the end state (3), essentially through inaction, and a new block is collectively selected, most likely `CloseInteraction` in this case.

If the participant's intent allows for an initiation to occur, the `Initiate` method in state (1) starts the creation of an `FmlInstance`. It begins by creating an `FmlBody` object with the initiating participant as its maker. Next it makes four `FmlChunk` objects, each containing a list of FML functions, a `TrackType` and a timing setting.

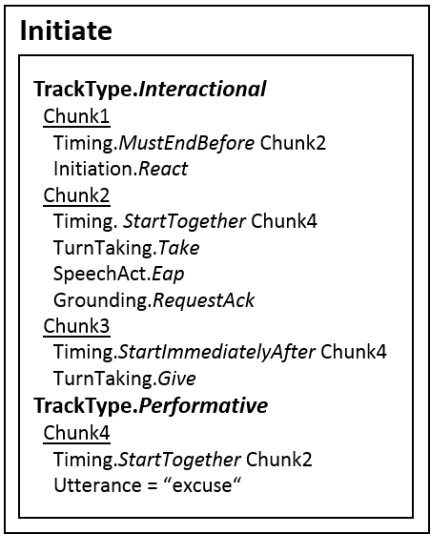

Figure 5. The FML chunks created by the Approach block's Initiate method. The chunks form the equivalent of an FML document body. Chunks within separate tracks may be executed together, depending on the Timing setting. FmlChunk objects have utterance variables that are initially set to the 'type' of utterance that should occur in each case and is later realized as a sentence in the agents' LanguageFaculty component.

The example `FmlBody` object setting in figure 5 is a typical example of the functionality of the methods in the states. After such an object is created, the participants who are not its maker have the opportunity to add functions and chunks to it. The `FmlBody` object has a method catching certain types of functions that call for immediate feedback from the other participants to be executed simultaneously. These functions include grounding requests for acknowledgement, initial reaction, speech act rituals and breaking away from the conversation. This concurrent feedback is provided by the participants who do not have the turn, adding to the fluidity and authenticity of the interaction.

Utterances in the `Performative` track are realized as sentences in the agents' `LanguageFaculty` via the `MakeUtterance` method. As of yet, the method is given a sentence 'type' and simply assigns a value to the relevant chunk's `Utterance` variable using a switch statement. Further plans for dealing with dialog management is discussed in section 6.1. An important function of the `MakeUtterance` method is to check whether the agent is possessed by the user, i.e. whether the agent is in fact the user's avatar. If that is the case, then this is the point at which the speech recognition software is triggered to allow the user to provide natural language input.

The next step after all participants contribute to the `FmlBody` is to create an `FmlInstance` object. The `Floor` class contains a method that joins the `FmlBody` with the current `Floor` instance and ships it off to the `GenerationManager` of the maker of the body and all those participants who made contributions to it. At this point the communicative

function mechanism has completed its task and behavior generation and realization takes over. In the case of IceLangVR, behavior is realized through a combination of utilizing Impulsion and custom animations in Unity3D using Mecanim[19]. Since the ECAs are given a voice via the IVONA speech synthesis system (see section 2.7), part of realizing their behavior is turning the performative chunk's text to speech. This does not apply when the user is possessing an agent; however, all other behaviors, such as movement of the mouth, pointing with hands, head gestures, etc., will be realized by the possessed agent.

Once all behavior for a particular `FmlInstance` has been executed, the `FloorState` is switched from `Executing` back to `Ready`, allowing the `DiscourseManager` once again to call that floor. This completes the cycle of collectively creating discourse models and realizing them as behaviors. The following section comments on the current capabilities of the system and concerns regarding its implementation.

5.3 Results and Performance

The system now provides the ECAs with a framework for producing communicative functions within task specific conversational sections called blocks. The agents' intentions inform the selection of blocks, which are proposed by each agent on the shared floor of interaction. One block is collectively selected and represents the new direction that the conversation is taking. Within each block, the participants progress through situations, or states, by taking actions that result in discourse models appropriate for the given state. The model produced is shared with the other participant[20] on the floor, allowing her to contribute communicative functions to it. A particular state may change a participant's intent, which will influence the selection of the next block, perhaps one that would not otherwise have been selected. These interactions are therefore collectively constructed and emergent conversations.

A weakness of the current implementation that needs to be addressed is that the selection of blocks on the floor is too dependent on a simplistic portrayal of personality involving only the agents' dominance levels. The agents individually suggest a block to the floor in accordance with their intentions, but the final selection depends solely on their

[19] http://docs.unity3d.com/Manual/AnimationOverview.html
[20] The interactions are currently dyadic. See section 6.1 for future work on multi-agent interaction.

dominance. The full extent of how corruptive this is to the users' experience is not clear, which calls for an overall assessment of the application in a user study (see section 6.2).

The state machine implementation ensures that the selection of blocks and their traversal happens quickly. The `FloorState` variable (see section 5.2) has the role of controlling the progression through the states and only allows advancement once the functions have been realized as behavior. Of course, behavior planning is a separate process that may take some time, as does the actual execution of the behaviors. Therefore, at the communicative function (FML) level there is room for actual goal-based search and the block implementation would be beneficial for a fast search by limiting the size of the state space, confining it to the current block (see section 6.1).

The project's current implementation is extendible in most respects. It is possible to create new blocks for some purpose specific task, complete with methods that produce discourse models. The code base is not overly complex, but in order to fully integrate a new block into the system one does need to spend some time getting familiar with it.

Section 6 provides a discussion of future work for this project and proposes a user study for quality assessment.

6 Discussion

6.1 Future Work

In the coming months the development of IceLangVR will continue, which calls for the incorporation of the proposed conversation system in this thesis with the implemented components of the larger project. Integration with the Scriptable Engine (see section 3) is a priority and will play an important role in introducing dialog management and language learning lesson planning to the system. Accomplishing this calls for the application of the tools and resources for Icelandic LT (see section 2.7). The agents' language component (the `LanguageFaculty` discussed in 5.2) should manage their dialog and NLP needs, of which syntactic and basic semantic analysis can be implemented using the tools available today.

The Scriptable Engine's event system can be used to direct tasks within lesson plans to agents in the virtual environment via a scheduling mechanism. The events could be made to influence the agents' intentions, which affects the selection of blocks and actions taken within them.

Additionally, since the agents in IceLangVR can form groups, the implementation of the conversation system has to move beyond dyadic interactions. The video data (see section 2.4) shows that a different interaction arises when the subject approaches a group as opposed to individuals, mainly in the way communicative functions are directed towards specific agents within the group.

Currently, the discourse models that arise during the course of conversation is more of a reflection of what has been seen to occur in human conversations rather than the result of reactive planning by the agents. The block structure allows for the implementation of sub-goal formulation and search for a solution that fulfills the goal in a relatively small state space, i.e. without searching outside the confines of the current block. This could allow the agents to anticipate certain behaviors, strive towards maintaining or losing a particular intent, and plan accordingly. Planning could also be performed at the block level, since the agents do have a sense of what they want and particular blocks may be able to meet their needs.

6.2 User Study Proposal

The following is a user study proposal to be conducted in the near future. The main goal of the study would be to gain a better understanding of how the system facilitates language learning, if that is indeed the case. The first goal of the study would be to on the one hand reveal how preoccupied the subjects are with the environmental attributes, such as the textures in the scene, the appearance of ECAs, etc., and on the other survey their feeling of presence when engaging in conversations. To the second point, the feeling of presence may stem mostly from the environmental attributes or the lesson planning and language learning aspect, or both. It is therefore important to be able to distinguish between these factors that may invoke a sense of presence in order to prioritize the aspects of IceLangVR that need further development.

The second goal would be to find out how the procedural emergence of the discourse models affects language learning. This could be accomplished by having the subjects go through an interaction where the procedural selection system is either not in effect or up and running. The former setting would mean having the conversational system static from beginning to end, resulting in the subject having more or less the same conversation with multiple agents. The latter would allow for the subject to engage in varied conversation from agent to agent, given that the agents do not all have the same intents and personality.

Additionally, the question of how conversational behavior on the part of the ECAs affects the subjects could be conducted in the same manner as above.

These experiments can be both qualitatively and quantitatively assessed. The qualitative assessment would involve surveying the subjects on their experience by posing various questions relating to particular aspects of the environment and the conversations. This assessment is especially important with respect to the first goal, gaining insight into what factors contribute to the subject's sense of presence. The first quantitative assessment involves following any change in the grades that the subjects receive after completing a lesson. Finding little or no overall improvement in the subjects' scores merits a refinement of the conversational and lesson planning. The second quantitative assessment is gauging system failures and break-downs in interaction, e.g. the conversation coming to an abrupt end or not taking place when they should. Such an evaluation is paramount to the refinement of the final product's quality.

In carrying out the user study, a between subjects design is appropriate. For testing the effects of ECA behavior on the subjects, for example, one group would interact with the version where the agents exhibit all behaviors within their capacity and the other group interacts with agents showing little or no behavior. Both groups would undergo the qualitative and quantitative assessments.

7 Conclusion

This thesis proposes a solution for embodied conversational agents to engage in conversation with other agents and human users, in a language and culture training environment. The proposed solution is geared specifically towards how strangers go about initiating and conducting interactions and involves enabling the agents to procedurally select purpose specific conversation sections that give rise to discourse models. Sharing these models on a floor of interaction allows all participants to collaborate and "join in" on the conversation.

This design is the result of keeping with the emergent nature of conversations, in that they can only be foreseen up to a point and the actions of others may impact our intentions and consequently our actions. The research that inspired the design was the work within discourse and conversational analysis, most notably the work of Herbert H. Clark (Clark, 1996) that introduced task oriented sections of conversation. In order to understand how strangers initiate and conduct conversations in Icelandic, video recordings of a stranger

walking up to another on the street and asking for directions in Icelandic were analyzed with respect to communicative functions and behavior.

This allowed for the computational modelling of the functions that underlie the observed behavior using parts of the FML standard proposal (Cafaro et al., 2014). The result is an application that provides the framework for ECAs in a virtual language learning environment to engage and collaborate with others in conversation, involving the use of current Icelandic language technology systems.

This solution contributes to the conversational skills that the IceLangVR ECAs already possess (see section 3), which is the ability to interact using communicative functions that result in behavior, such as turn-taking, coupled with personality traits. The novelty here is confining such interactions to task specific sections wherein the agents work together and once the task is completed they both have a say in what the next one will be. The ways in which this is accomplished involves using a combination of the agents' personalities and their intentions. The intentions, however, are not static like personality and may change over the course of the conversation. This relates to conversations as being emergent, since the participants may influence each other's intentions during the interaction and the selection of conversation sections cannot be predetermined.

The work also contributes to Icelandic LT in the following ways. The system described here has successfully, for the first time, implemented both Icelandic speech recognition and synthesis, used for user input and as the agents' voice, respectively. In the near future, more Icelandic specific NLP tools will be incorporated into the IceLangVR application as the language learning, understanding and generation aspects of the project expand. Modelling conversational functions and behavior based on interactions conducted by native speakers has an element of language preservation to it. The various reflexes and rituals that come across in mundane conversations can be particular to a culture and modelling them results in a digital preservation of language use conversational behavior. Not only does this work add to the collection of Icelandic LT applications, it also has a part to play in its preservation.

References

Abuczki, Á., & Ghazaleh, E. B. (2013). An overview of multimodal corpora, annotation tools and schemes. *Augmentum, 9*, 86-98.

Arnbjörnsdóttir, B. (2008). Kennsla tungumála á netinu: hugmyndafræði og þróun Icelandic online. *Hrafnaþing*, 7-31.

Bédi, B. (2015a). *A Corpus Collection of Video Data for Virtual Reykjavik.* Unpublished raw data. Part of a PhD-thesis. University of Iceland.

Bédi, B. (2015b). *Multimodal Annotation Scheme for Virtual Reykjavik*. Unpublished manuscript. Part of a PhD-thesis. University of Iceland.

Cafaro, A. (2014). *First Impressions in Human-Agent Virtual Encounters.* (PhD), Reykjavik University.

Cafaro, A., Vilhjálmsson, H. H., Bickmore, T., Heylen, D., & Pelachaud, C. (2014). Representing Communicative Functions in SAIBA with a Unified Function Markup Language *Intelligent Virtual Agents* (pp. 81-94): Springer.

Cassell, J., Bickmore, T., Billinghurst, M., Campbell, L., Chang, K., Vilhj, H., . . . Yan, H. (1999). *Embodiment in conversational interfaces: Rea*. In the Proceedings of the SIGCHI conference on Human Factors in Computing Systems, Pittsburgh, Pennsylvania, USA. http://dl.acm.org/citation.cfm?doid=302979.303150

Cassell, J., Bickmore, T., Campbell, L., Vilhjálmsson, H., & Yan, H. (2000). Human Conversation as a System Framework. In J. Cassell, J. Sullivan, S. Prevost & E. Churchill (Eds.), *Embodied Conversational Agents* (pp. 29-63). Cambridge MA: Massachusettes Institute of Technology.

Clark, H. H. (1996). *Using Language*: Cambridge University Press.

Creating and Using Scripts. (2015). Retrieved April 5, 2015, from http://docs.unity3d.com/Manual/CreatingAndUsingScripts.html

Cummins, J. (1998). *Immersion Education for the Millennium: What We Have Learned from 30 Years of Research on Second Language Immersion*.

Fifteenth International Conference on Intelligent Virtual Agents (IVA 2015), August 26-28, Delft. (2015). *International Conference on Intelligent Virtual Agents.* Retrieved March 18th 2015, 2015, from http://iva2015.tudelft.nl/

Guðnason, J., Kjartansson, O., Jóhannsson, J., Carstensdóttir, E., Vilhjálmsson, H. H., Loftsson, H., . . . Rögnvaldsson, E. (2012). *Almannarómur: an open icelandic speech corpus.* In Proceedings of the SLTU.

Heritage, J. (2009). Conversation Analysis as Social Theory. *The New Blackwell Companion to Social Theory.* (pp. 300-320): Wiley-Blackwell.

Johnson-Laird, P. N., & Garnham, A. (1980). Descriptions and discourse models. *Linguistics and Philosophy, 3*(3), 371 - 393.

Johnson, W. L., Marsella, S., Mote, N., Viljhalmsson, H., Narayanan, S., & Choi, S. (2004). *Tactical Language Training System: Supporting the rapid acquisition of foreign language and cultural skills*. In Proceedings of the STIL/ICALL Symposium, Venice, Italy.

Johnson, W. L., & Valente, A. (2008). *Tactical language and culture training systems: using artificial intelligence to teach foreign languages and cultures*. In the Proceedings of the 20th national conference on Innovative applications of artificial intelligence - Volume 3, Chicago, Illinois.

Jónsson, J. H. (2012). Íslenskt orðanet. Retrieved April 10, 2015, from http://ordanet.is/

Kendon, A. (1990). *Conducting interaction: Patterns of behavior in focused encounters*.

Kopp, S., Krenn, B., Marsella, S., Marshall, A. N., Pelachaud, C., Pirker, H., . . . Vilhjalmsson, H. (2006). *Towards a common framework for multimodal generation: the behavior markup language.* In Proceedings of the 6th international conference on Intelligent Virtual Agents, Marina Del Rey, CA.

Loftsson, H., & Rögnvaldsson, E. (2007). *IceNLP: a natural language processing toolkit for icelandic.* In Proceedings of INTERSPEECH.

Pedica, C. (2009). *Spontaneous Avatar Behavior for Human Territoriality.* In Proceedings of the 9th Invernational Conference on Intelligent Virtual Agents.

Russell, S. J., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*: Pearson Education.

Rögnvaldsson, E., Jóhannsdóttir, K. M., Helgadóttir, S., & Steingrímsson, S. (2012). Íslensk tunga á stafrænni öld - The Icelandic Language in the Digital Age. In G. Rehm & H. Uszkoreit (Eds.), *META-NET White Paper Series*.

Slater, M., & Wilbur, S. (1997). A Framework for Immersive Virtual Environments (FIVE) - Speculations on the Role of Presence in Virtual Environments. *Presence: Teleoperators and Virtual Environments, 6*(6), 603-616. doi: citeulike-article-id:4678276

Vilhjálmsson, H., Björgvinsson, E., Helgadóttir, H., Kristinsson, K., Ólafsson, S., Cafaro, A., . . . Bédi, B. (2014). *Social Gatherings in Virtual Reykjavik*. Demo and Poster at the 14th International Conference on Intelligent Virtual Agents, Boston.

Vilhjálmsson, H. H. (2013). Icelandic Language and Culture Training in Virtual Reykjavík: RANNÍS. Research Funt Grant Proposal.

Vilhjálmsson, H. H., Björgvinsson, E. I., Helgadóttir, H. E., & Ólafsson, S. (2014). *We Never Stop Behaving: The Challenge of Specifying and Integrating Continuous Behavior*. In Proceedings of the Workshop on Architectures and Standards for IVAs at the 14th International Conference on Intelligent Virtual Agents, Boston.