

Augmenting Online Conversation through Automated Discourse Tagging

Hannes Högni Vilhjálmsón
 USC Information Sciences Institute¹
 4676 Admiralty Way, Suite 1001
 Marina del Rey, CA 90292
 hannes@isi.edu

Abstract

In face-to-face communication, the communicative function of the spoken text is clarified through supporting verbal and nonverbal discourse devices. In computer-mediated communication, the mediating channel may not be able to carry all those devices. To ensure the original intent gets communicated effectively, discourse tags can be embedded in a message to encode the communicative function of text given the context in which it was produced. The receiving client can then generate its own supporting discourse devices from the received tags, taking into account the receiver's context. Spark is a synchronous CMC architecture based on this concept of a message transformation, where an outgoing text message gets automatically annotated with discourse function markup that is then rendered as nonverbal discourse cues by a graphical avatar agent on the receiver side. A user study performed on a derived application for collaborative route planning demonstrates the strength of the approach.

1. Introduction

A helpful way to look at communication is to distinguish between what is being communicated and the means by which it is communicated. In written discourse, words and their arrangement provide the means while in face-to-face conversation nonverbal behavior such as gesture, gaze, and tone of voice, also play an important role. An idea can be communicated in a variety of ways that depend on the devices available to us. At the level of an idea or communicative intent, we can speak of the need to carry out certain *discourse functions*, while at the level of realization we can speak of the *discourse devices* that perform the actual work.

An example of a discourse function is topic change. In written discourse, the use of headings, paragraphs and phrases such as “in other news” serve as discourse devices that accomplish topic change. In face-to-face conversation, a pause, a change in posture, the picking up

of a new object and words such as “anyway” are discourse devices that accomplish that same function.

Not only does the function versus device distinction provide an interesting insight when studying discourse, but it can also be a powerful tool when building communication systems. At the design stage we can think about the technology in terms of what discourse functions can generally be supported by the range of discourse devices we make available. At run-time, we can look at the discourse devices that are employed by the users of our systems to derive their communicative intent and use that knowledge to provide further support.

Spark is an architecture for synchronous, i.e. real-time, online messaging that is informed by this discourse theory as well as by studies of face-to-face conversational behavior. In essence, Spark aims to augment regular text chat by adding animated avatars that provide supporting face-to-face discourse devices. In meeting that objective, Spark also demonstrates a powerful mechanism for transforming messages based on the mapping from discourse function to discourse devices.

The next section will review some of the related work and the following section will outline the architecture. Sections 4 through 6 will discuss details of implementation and finally sections 7 and 8 will provide evaluation and conclusions.

2. Related Work

2.1. Discourse Functions

The process of conversation has been widely studied and many different discourse functions cataloged, but two categories seem to stand out as being crucial for a successful exchange: *interactional* functions and *propositional* functions [1]. These roughly correspond to what some have termed coordination of process and content [2] and others interactional and transactional functions [3] respectively.

Interactional functions deal with managing the conduct itself. This includes negotiating when to start or end a conversation [4], establishing mutual focus of attention

¹ The research described in this paper was performed by the author at the MIT Media Lab

[5], making sure that turns are exchanged in an orderly fashion [6, 7] and maintaining evidence of continued attention through backchannel feedback [8]. Basically these functions establish and maintain an open channel of communication between participants.

Propositional functions on the other hand deal with how information gets packaged and shared across an already open channel. Each utterance can be regarded as an instruction from a speaker to hearers on how to update the pool of shared knowledge, a structure often referred to as a *discourse model* [9]. As the things being talked about, referred to as *discourse entities*, enter this model, the speaker can make use of their saliency to communicate more efficiently. A good example is the use of pronouns shortly after someone has been named.

According to the theory of *information structure* [10], only a part of each utterance, the *rhematic* part, serves the function of updating the discourse model, while the remaining *thematic* part anchors the contribution in the ongoing discourse. For example, in the response to the question “who are you?” the rhematic part of “I am your friend” would be “your friend”.

Even though a speaker successfully produces an utterance it is not guaranteed that listeners successfully decode it and update the discourse model correctly. Therefore a process known as *grounding* often takes place during conversation, where speakers and listeners verify the successful transmission of information [11].

The organization of utterances into topics and sub-topics has been described as *discourse structure* [12]. This structure is important because it provides a context for interpreting what is being said and is therefore often made explicit through special discourse functions that manipulate it.

2.2. Discourse Devices

When conversation is conducted face-to-face, the interactional and propositional functions get support from the available nonverbal communication channels. For the most part, people are not conscious of these behaviors, but evolution has provided us with mechanisms that are quite effective.

Establishing and maintaining participation in a conversation is largely dependent on appropriate body orientation and gaze direction. To engage people in a conversation, one has to show them visual attention beyond what would be considered a passing glance according to [4, 13]. Subject to the other people’s reciprocal action and acceptance, salutations are exchanged. Finally it is possible to move closer and everyone re-orient themselves such that they have clear access to each other’s field of attention [5].

Requesting a turn in conversation, typically involves breaking eye contact with the speaker and raising the hands in preparation for gesturing. A speaker can give

the turn by going silent and looking at the person who is meant to get the next turn [5, 6, 14]. Speakers often request backchannel feedback by looking at the listener and raising the eyebrows [15]. To request a more involved feedback, this behavior can be supplemented with pointing the head towards the listener or a series of low amplitude head nods ending in a raising head [14].

As for propositional functions, several types of behaviors help listeners decode information packaged in utterances. Typically, the rhematic part of an utterance is underlined with intonation or with increased gestural activity. Particularly important discourse entities are emphasized with pitch accents, head nods, eyebrow raising or precisely timed gesture strokes called beats. Discourse entities in the shared visual context can simply be referred to by pointing. Some entities have a salient visual feature that an illustrative gesture can elaborate on when the entity is mentioned [16].

Grounding is often accomplished through a purely nonverbal exchange. Positive evidence of understanding include head nods, facial expressions and expected actions such when following instructions. Negative evidence can be in the form of expressions of puzzlement or in the failure to react at all to requests of feedback [14, 15].

Discourse structure and the transitions within it are clearly reflected in the accompanying nonverbal stream [15, 16]. Behaviors typically involve motion and a number of body parts proportional to the impact of the shift on the ongoing discourse [17]. For example, changing the topic of the conversation altogether is usually preceded by a change in overall posture, whereas a digression from a main point is often accompanied by a slight gesture to the side [18].

2.3. Online Conversation

People attribute to synchronous online communication qualities of face-to-face conversation because of how relatively responsive and spontaneous the medium is [19, 20], but the reduced range of available discourse devices makes the medium unsuitable for many of the tasks traditionally solved face-to-face.

Beyond the perhaps obvious limitation that typing speed imposes on the pace of text based conversation, some of the reported difficulties of using chat include difficulty of recognizing and engaging participants [21, 22], limited turn negotiation mechanism [19, 23, 24], overlapping topic threads [19, 21, 22, 24-26], lack of feedback, leaving those not actively messaging practically invisible [22, 27], and no way of visually establishing referents or focus of attention [28].

Veteran users of synchronized CMC systems have adapted to the medium and created a number of textual conventions, essentially new discourse devices, that try to overcome these limitations [19, 29]. But forcing new or

casual users to learn these conventions can lead to frustration and a low acceptance of the medium.

Adding video for directly transmitting nonverbal behavior has met with less success than many predicted, in part because discourse devices are projected from one visual context into another without an attempt to preserve the original function. For example selecting the next speaker with gaze is impossible with standard web cam setup [30]. Attempts have been made to correct for this but they either focus on a narrow range of behavior (for example head orientation) [31, 32], or they require an elaborate setup out of reach for most home users [33].

Avatars can provide participants with virtual bodies that share a visual context. Most graphical chat systems require their users to explicitly control the nonverbal behavior exhibited by their avatar. But since most of the discourse devices discussed here are produced unconsciously, they are therefore not re-produced in these avatars. BodyChat addressed this issue by automating communicative behavior in avatars, focusing on animating gaze algorithmically to serve interactional functions [34]. More recent work has demonstrated the power of this approach by showing that algorithmic control of avatar gaze outperforms random gaze or no behavior at all [35-37]. No one has attempted to automate the range of nonverbal cues needed to cover both interactional and propositional discourse functions.

3. The Spark Architecture

Spark is an architecture for synchronous online communication based on the idea that if a transmitted message contains a description at the level of discourse function, any discourse devices that didn't fit through the transmission channel could be recreated, or at least replaced with an equivalent device, on the receiving end.

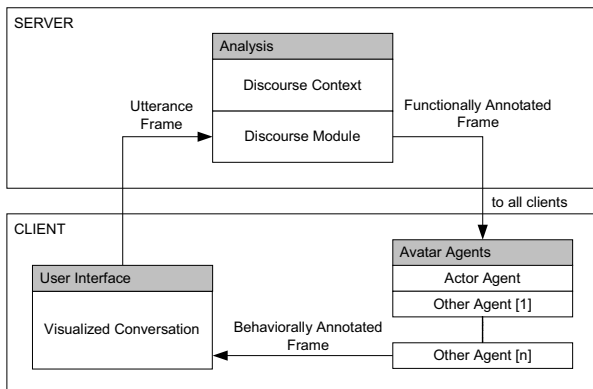


Figure 1: An overview of the Spark architecture

The architecture is an XML message pipeline where the original message called a *frame* is first passed through an analyzer that annotates the message with discourse function tags. The annotated frame is then broadcast to

all clients, where an avatar agent, essentially a proxy, applies a series of transformation rules to the functional markup resulting in a new set of markup, called behavioral markup, that represents supporting discourse devices. Finally the behaviorally marked up frame is rendered as an animated performance synchronized with the delivery of the original message. The pipeline is built on the BEAT framework for behavior generation [38], but departs most significantly from BEAT by making a clear distinction between function and behavior, and by addressing multi-party conversation.

The next three sections will discuss three key components of the Spark architecture. The first is the Discourse Module that is responsible for adding the functional markup. It is impossible to infer much about discourse without a rich representation of context. Another key component is therefore the database that keeps track of the Discourse Context. The last component discussed here is the Avatar Agent.

4. Discourse Module

The Discourse Module uses the Discourse Context (see Section 5) to add discourse function tags to a message, wrapped into an XML structure called an utterance frame. An example of this annotation is given in Figure 2.

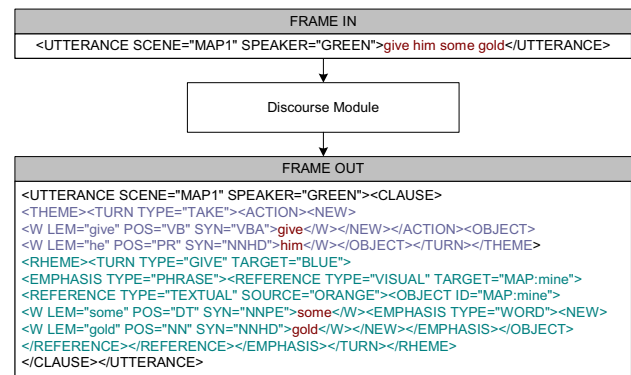


Figure 2: A text message before and after discourse function tagging

Before actual discourse processing, the incoming utterance frames need to be broken into basic units such as words, phrases and clauses. A tokenizer encloses words and punctuation in W tags. It then consults a part-of-speech tagger (The EngLite tagger from Conexor) to fill in attributes for each word. The first attribute is the actual part-of-speech (POS), such as noun or a verb. The second attribute is the lemma of the word (LEM), i.e. the basic root form of the word. The third attribute is a light syntax identifier that describes where the word stands in relation to the words around it. This generally marks words as either as the head of a phrase, such as a noun

phrase or a verb phrase, or modifiers to the head. Next a chunker groups the words together into phrases and clauses based on punctuation and word classes. Noun phrases get marked with an OBJECT tag, verb phrases with an ACTION tag and clauses with a CLAUSE tag.

The discourse processing is handled by a number of annotation methods, each applied in turn to the utterance. These methods use the discourse context, existing annotation and heuristics drawn from the literature to progressively enrich the description. What follows is a summary of each of the currently implemented methods, in the order they are applied. The summaries start with a short description of the discourse phenomenon. Then “Uses” lists the tags and attributes that need to be present (the format is comma separated tag names immediately followed by any needed attributes identified with “@”). “Creates” lists the tags that get inserted into the text and finally the algorithm itself is described.

4.1. Mark New

Function: Lexical givenness. Whether a certain word has been seen before in the current discourse.

Uses: W @POS @LEM

Creates: NEW

Method: Enclose every W element whose POS attribute indicates an adjective, noun or a verb (words belonging to any *open class* except the adverb class) and whose LEM attribute is not identical to the LEM attribute of any W element in the discourse history.

4.2. Mark Topic Shifts

Function: Movement within the discourse structure. Seeing the discourse structure as a stack of topics, where topics can be pushed onto the stack and popped off later.

Uses: CLAUSE, W @LEM @SYN

Creates: TOPICSHIFT @TYPE=(NEXT | PUSH | POP | DIGRESS | RETURN)

Method: Tag the first W of a CLAUSE (skipping to the second if the first W is a connective) if its LEM attribute matches any of the following topic shift discourse markers adopted from [3] :

Next - and, but, then, speaking of that, that reminds me, one more thing, before I forget

Push - now, like

Pop - anyway, but anyway, so, as I was saying

Digress - incidentally, by the way

Return - anyway, what were we saying

4.3. Mark Information Structure

Function: The thematic and rhematic components of a clause.

Uses: CLAUSE, OBJECT, ACTION, NEW

Creates: THEME, RHEME

Method: Groups together all OBJECTs that occur before the first ACTION in a CLAUSE, calling that the pre-verbal group. Similarly the group of any OBJECTs or ACTIONs occurring after that first ACTION gets called the post-verbal group. If a group or the ACTION contains a NEW element, it is marked as focused. If the pre-verbal group is the only focused group or element, it gets tagged as RHEME and the post-verbal group as THEME, otherwise the post-verbal group gets the RHEME tag and the pre-verbal the THEME tag. If there is only one group, it gets tagged as a RHEME regardless of focus status. If the post-verbal group is focused, the ACTION gets counted with the pre-verbal group, otherwise the post-verbal. This follows the heuristics described in [39].

4.4. Mark Emphasis

Function: Particular attention should be drawn to this word or part of utterance.

Uses: RHEME, ACTION, OBJECT, NEW

Creates: EMPHASIS @TYPE=(PHRASE | WORD)

Method: All numbers get tagged (TYPE = WORD). Every ACTION or OBJECT within a RHEME that contains a NEW element gets tagged (TYPE = PHRASE) and all the NEW word elements also get tagged (TYPE = WORD).

4.5. Mark Contrast

Function: Two or more items are being contrasted with each other (currently only antonyms).

Uses: @POS @LEM

Creates: CONTRAST @ID

Method: For each W that is an adjective, tag if its LEM attribute equals the lemma of any antonym or any synonym of that antonym of an earlier adjective W (using WordNet). If a match is found within the current utterance, both W elements get tagged and get an ID number identifying the pair.

4.6. Identify Clauses

Function: The general communicative purpose of the clause. Essentially speech act category, but currently limited to what punctuation reveals.

Uses: CLAUSE, W @SYN

Creates: CLAUSE @TYPE = (EXCLAMATION | QUESTION)

Method: All clauses ending in a question mark get TYPE = QUESTION and all clauses ending in an exclamation mark get TYPE = EXCLAMATION.

4.7. Identify Objects

Function: Label mentioned discourse entities.
Uses: UTTERANCE @SCENE, OBJECT, W @LEM
Creates: OBJECT @ID
Method: For all OBJECTs try to find a match in the set of instances listed in the domain knowledge base (KB) and in the discourse history. If a match is found in the KB, then the OBJECT gets the unique ID of the matched instance. If not found, the discourse history is searched in case the OBJECT was created during preceding discourse. If a match is then found, the ID of that OBJECT is used. If no match is found, a new unique ID is assigned to the OBJECT. A match score between an OBJECT and an instance in the KB is the number of instance features that are identical to any W LEM attributes contained in the OBJECT. A match score between two OBJECTs is calculated as the number W LEM attributes they contain that are identical. The match that scores the highest is picked as the match. If there is a tie, no match is reported.

4.8. Identify Actions

Function: Actions may have features that lend themselves well to visual illustration. Here, verb phrases are linked to action descriptions in the knowledge base.
Uses: ACTION, W @POS @LEM
Creates: ACTION @ID
Method: For all ACTIONs, try to find a match in the set of action descriptions listed in the KB. It is a match if the lemma of the head verb in the ACTION's verb phrase is identical to an action description identifier. If no match is found then the search is repeated with the set of all hypernyms of the head verb. Any matching identifier is used as the ID value of the ACTIONs. The ID is left blank if no match is found.

4.9. Mark Reference

Function: Discourse entity is not new but evoked through a textual or visual reference [40].
Uses: UTTERANCE @SCENE, OBJECT
Creates: REFERENCE @TYPE=(VISUAL | TEXTUAL) @TARGET @SOURCE
Method: Every OBJECT that matches any of the instances listed in the scene description is tagged and the TYPE set to VISUAL and the ID to the instance ID. Every OBJECT that matches any of the OBJECTs in the discourse history is tagged and the TYPE set to TEXTUAL, the ID set to the matched OBJECT's ID and the SOURCE set to the ID of the person who last contributed the OBJECT to the discussion.

4.10. Mark Illustration

Function: Indicate a feature of a discourse entity that should be emphasized through illustration.
Uses: OBJECT, ACTION
Creates: ILLUSTRATE @DESCRIPTION
Method: Every OBJECT within a RHEME and that contains a NEW element gets checked against the KB using the object ID. If this instance of an object has an unusual value assigned to an object feature, as determined by the definition of a typical instance in the KB, a description of the atypical feature and value are assigned to DESCRIPTION as a string. Every ACTION within a RHEME and that contains a NEW element gets checked against the KB using the action ID. If a description of the action, or any of its hypernyms (a more generic verb) as shown by WordNet, is found in the KB, that description is assigned to DESCRIPTION.

4.11. Mark Interaction Structure

Function: Currently marks addressee.
Uses: UTTERANCE @SPEAKER @SCENE
Creates: UTTERANCE @HEARER
Method: If the HEARER attribute of an UTTERANCE is not already set, first all OBJECTs in the UTTERANCE are examined to see if there is a match with any instance of a person in the set of participants for the scene identified in the SCENE attribute. If a match is found, that person's ID is set as HEARER. If no match is found, then HEARER is set to the person who was the last speaker. If there was not last speaker (this is the first utterance of a conversation), HEARER is left undefined.

4.12. Mark Turn Taking

Function: Floor negotiation.
Uses: THEME, RHEME
Creates: TURN @TYPE=(TAKE | KEEP | GIVE), @TARGET
Method: Tag all RHEMES that are at the end of an utterance with TURN of TYPE GIVE and TARGET set to HEARER. Tag all THEMES that are at the beginning of an utterance with TURN of TYPE TAKE. If the THEME is not at the beginning of an utterance, tag it 70% of the time with TURN of TYPE KEEP. This, in conjunction with 4.13 implements the algorithm presented in [41]

4.13. Mark Grounding

Function: Requests for backchannel feedback from listeners (other types of grounding are not automated).
Uses: RHEME

Creates: GROUNDING @TYPE = REQUEST
@TARGET

Method: If a RHEME is not at the end of an utterance, tag it 70% of the time with GROUNDING of TYPE REQUEST and set TARGET to HEARER.

5. Discourse Context

5.1. Domain Knowledge Base

The Domain Knowledge Base (KB), essentially an ontology, is an XML file that describes the set of discourse entities and actions likely to enter the conversation. The entries in the KB are of three different types: object type, object instance, and feature description.

Type definitions associate features and their typical values with generic object types. These object types serve as templates for specific object instances that can be recognized in the discourse. An example of a type definition would be:

```
<TYPE NAME="TREE" CLASS="OBJECT">
<NUMFEATURE NAME="HEIGHT" TYPICAL="30-90" />
<SYMFATURE NAME="SHAPE" TYPICAL="STRAIGHT" />
</TYPE>
```

An instance of a certain type defines a discourse entity with a unique ID. An instance describes the features of the particular entity, possibly flagging an unusual trait that could be exploited if an illustration is called for. An example of an instance definition would be:

```
<INSTANCE OF="TREE" ID="TREE1" HEIGHT="35"
SHAPE="CROOKED" />
```

For feature values or actions that need to be illustrated, their description can be looked up in the KB by the name of the value or action.

5.2. Participation Framework

The participation framework describes everyone's role in the conversation. Participation status can currently be any of HEARER (ratified), ADDRESSEE (focus of speaker's attention) or SPEAKER. When no one is speaking, a HEARER status is assumed for everyone.

When the Discourse Module sets the status of a participant, the structure automatically updates the status of the other members if necessary. In particular, if person A is currently a SPEAKER and person B gets SPEAKER status, then the person A gets ADDRESSEE status if a new addressee was not named, otherwise a HEARER status. This implements the turn taking rule from BodyChat.

5.3. Discourse Model

The discourse model is the part of the discourse context that keeps track of the dynamic state of the overall discourse through a discourse history and a visual scene description.

There are two parts to the discourse history. The first part is simply a list of all tagged utterance frames processed so far. Leaving them tagged allows the history to be searched both by lexical items and discourse function. The second part is a recency list of discourse entities. This is a list of discourse entities that have been created during the course of the discourse, with the most recently referred to entity on the top. Only one instance of each entity is allowed in the list, so when an entity is referred to a second time for example, it gets promoted to the top. The scene description simply enumerates participants and any object instances that are currently visible to everyone.

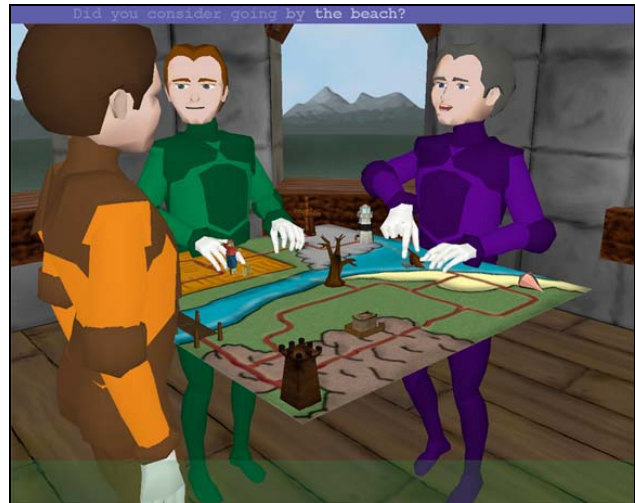


Figure 3: The Avatar Agent on the right delivers a message to the person across the table

6. Avatar Agent

After the Discourse Module has processed and annotated an utterance frame with functional markup and distributed it to all connected clients, an Avatar Agent representing the sender of the message gets to add supporting discourse devices (see Figure 3). It does this by applying a series of transformation rules on the functional markup generating new XML tags describing the supporting behaviors.

Discourse Function Tag	Discourse Device Tag (Behavior)	Listener Reaction Tag
EMPHASIS @TYPE=WORD	HEADNOD	
	GESTURE @TYPE=BEAT	
EMPHASIS @TYPE=PHRASE	EYEBROWS @TYPE=RAISE	
GROUNDING @TYPE=REQUEST	GLANCE @TARGET=ADDRESSEE	GLANCE @TARGET=SPEAKER
	EYEBROWS @TYPE=RAISE	HEADNOD
TURN @TYPE=GIVE	LOOK @TARGET=ADDRESSEE	LOOK @TARGET=ADDRESSEE
TURN @TYPE=TAKE	GLANCE @TARGET=AWAY	LOOK @TARGET=SPEAKER
TOPICSHIFT	POSTURESHIFT	
REFERENCE @TYPE=TEXTUAL	GLANCE @TARGET=SOURCE	
REFERENCE @TYPE=VISUAL	GLANCE @TARGET=OBJECT	GLANCE @TARGET=OBJECT
	GESTURE @TYPE=POINT @TARGET=OBJECT	
CONTRAST	GESTURE @TYPE=CONTRAST	
ILLUSTRATE @DESCR.=X	GESTURE @TYPE=ICONIC @DESCR.=X	

Table 1: The basic set of function to device transformation rules executed by the speaker and listener Avatar Agents, resulting in an animated face-to-face performance

Each transformation stands for a rule that associates a discourse device with a discourse function as described in 2.2 and summarized in Table 1. Finally the frame is passed around to all the other Avatar Agents, representing the rest of the participants, which then get a chance to add any programmed reactions.

A transformation can be written in an XML transformation language such as XSLT. Here is an example of a simple rule to generate head nods for emphasis:

```
<!-- Nod head on word emphasis. -->
<xsl:template match="EMPHASIS [@TYPE='WORD'] "
priority="10">
<HEADNOD>
  <xsl:copy>
    <xsl:apply-templates select="*|node()" />
  </xsl:copy>
</HEADNOD>
</xsl:template>
```

This generation rule looks for any tag with the name EMPHASIS and of TYPE WORD (see the highlighted “match” expression) and then surrounds that tag with a new HEADNOD tag (see the highlighted tags). The discourse function EMPHASIS is therefore getting supported here through the precisely placed HEADNOD behavior. Transformation rules in Spark can also be written in C++, which is useful when the transformation requires any computation beyond pattern matching.

7. Evaluation

The Spark approach was evaluated by implementing a scenario where three users had to get together in a virtual map room after being told they were prisoners in an enemy stronghold and had to plan their escape. This scenario was chosen because it involved conversation about a complex visual object, the map, and because it involved collaboration, which could provide some insight

into whether the avatar behaviors contributed to the conversation process and a successful outcome.

First, to evaluate how well the discourse devices chosen by the avatars mirrored those observed in real people, a physical mockup of the scene was created and videotaped with three live subjects performing the task. The utterances from a random 40-second segment of the video were fed through the Spark architecture and the resulting avatar behavior compared to that of the real people, which was annotated using Anvil [42] (see Figure 4). The analysis involved emphasis gestures, pointing gestures, gaze direction and head movements. Overall Spark did well, making exact predictions of behavior more than half the time, with the wrong predictions mainly due to excessive emphasis and backchannel feedback generation. That makes sense, because the avatars in Spark generate a behavior every time a rule indicates a logical opportunity. If Spark took into account factors such as personality or affect, it might be able to use that as a principled way of reducing overall avatar liveliness. This is something to consider as future research.

50 subjects were signed up for 2 sessions each with the Spark system, where a session involved a group picking what they believed to be the quickest escape route on the map in front of them. Each subject was briefed on a different set of helpful facts about the map prior to a session to ensure they needed to work together. In half of the sessions the subjects would just see the room and the map, and receive each other’s messages without the avatars. In the other half, everything would be the same except they would see each other as animated avatars standing around the map (see Figure 3). In both kinds of sessions the subjects could highlight parts of the map to indicate their choice of path.

Speech	So if we go this way, then we will have a choice through the swamp or choose it through..or we can stay in the tent									
Head		Nod		Nod		Nod				Shake
Gesture	Mountain		Tent to Swamp						Tent	Beat
Gaze	Map									
Head			Nod		Nod		Nod			Nod
Gesture			Beat		Swamp		Beat			Tent
Gaze	At addressee	Map	At addressee	Map		At addressee	Map			

Figure 4: An example of nonverbal behaviors observed when a person spoke an utterance (top) compared to the behaviors generated by Spark for the same utterance (bottom)

Two other conditions, crossed with the avatar versus no-avatar conditions, were the use of synthesized speech versus teletype style text. Apart from noting that people typically didn't like the synthesized voices, this part of the study won't be discussed further here.

The fact that each subject was assigned to 2 instead of all conditions (although balanced for order effects and only assigned to adjacent cells) of the 4 conditions in this 2x2 design, made the analysis of the data more difficult and contributed to lower power than with standard within-subject experiments, which suggests a simpler design for future studies. Nevertheless, some clear results emerged.

The 14 subjects that tried both an avatar system and a system without avatars were asked to compare the systems on a 9 point likert scale from a high preference for no avatars to a high preference for avatars along 6 dimensions including which system was "more useful", "more fun", "more personal", "easier to use", "more efficient" and "allowed easier communication". One-tailed t-tests showed that the preference for avatars was significant ($p < 0.05$) for all but the "easier to use" question where no significant preference either way was found. These subjective results clearly indicate that people find the avatars compelling and helpful.

To test the hypothesis that the avatars providing supporting discourse devices would improve the overall process of conversation, compared to text-only messaging, 11 different measures of quality of conversation process were taken. Seven were objective behavioral measures from the chat logs, including the portion of utterances without explicit grounding (i.e. verbal verification of reception), portion of utterances that got requested replies, portion of non-overlapping utterances and portion of on-task utterances. Four were subjective likert scale questionnaire measures, including sense of ability to communicate and sense of control over conversation. All but one measure was found higher in the avatar condition and a t-test of the grand mean (across all 11 normalized measures) showed that indeed it was significantly higher ($p < 0.02$) in the avatar condition than in the non-avatar condition, supporting the hypothesis.

To test the hypothesis that the avatars providing supporting discourse devices would improve the actual outcome of the collaboration, compared to text-only messaging, 8 different measures of the quality of task outcome were taken. Two were objective measures, one being the quality of the escape route that the subjects chose together and the other being the completion time (which ranged from 5 to 40 minutes). Six were subjective likert scale questionnaire measures including "How well did you think the group performed on the task?", "How strong do you think the group's consensus is about the final solution?" and "How much do you think you contributed to the final solution?". Again, all but one measure was higher in the avatar condition, and again, a t-test of the grand mean (across all 8 normalized measures) showed that it was significantly higher ($p < 0.02$) in the avatar condition than in the non-avatar condition, supporting this hypothesis as well.

8. Conclusions

Spark is able to produce a visualization of synchronous online conversation that mimicks the appearance of face-to-face interaction, based on an approach informed by discourse analysis and conversational behavior research. By constructing the visualization in a principled way with regard to what discourse functions are supported and to the role of each discourse device, benefits to the conversational process can be expected.

While the experiment indicates an overall benefit, the statistical power was not great enough to draw conclusions about many of the individual measures. A follow-up study of a simpler design should operationalize and investigate each discourse function in depth to provide evidence that the supporting discourse devices are doing their job.

The current Spark implementation generates discourse devices based on a model of face-to-face conversation. However, other models are also possible since the Avatar Agents can be viewed as general social proxies [43] capable of taking any desired form. For

example, an abstract representation, such as the one used in [44], would sometimes make more sense.

Because the transformation from function to devices takes place on each client, it is possible for clients to render different views of the conversation. For example, they could localize behavior mapping to reflect the culture of the recipient, or a note-taking client could use the functional markup to produce well organized notes.

The main limitation of the approach is the reliance on the difficult task of inferring discourse function from text. This is an area of continuous improvement and Spark makes it easy to plug in new or improved methods as they become available. Another related limitation is that the discourse processing introduces a messaging lag that could undermine actual benefits of the approach. This lag is hardly noticeable with very short utterances, but longer utterances (more than 10 words) can take several seconds. This problem will hopefully go away as technology improves.

Acknowledgements

The author would especially like to thank Justine Cassell for her guidance throughout the project and Amy Bruckman, Cynthia Breazeal, Bruce Blumberg and Dan Ariely for feedback that helped shape this work. The author is grateful to Ian Gouldstone for beautiful artwork and to the entire GNL team and Deepa Iyengar for input along the way. Last but not least, the author would like to thank the many sponsors of Digital Life at the MIT Media Lab for their support.

References

- [1] J. Cassell, T. Bickmore, L. Campbell, H. Vilhjalmsson, and H. Yan, "More than just a pretty face: conversational protocols and the affordances of embodiment," *Knowledge-Based Systems*, vol. 14, pp. 55-64, 2001.
- [2] B. O'Connell and S. Whittaker, "Characterizing, Predicting, and Measuring Video-Mediated Communication: A Conversational Approach," in *Video-Mediated Communication*, K. Finn, A. Sellen, and S. Wilbur, Eds.: Lawrence Erlbaum Associates, Inc., 1997, pp. 23-49.
- [3] H. H. Clark, *Using Language*: Cambridge University Press, 1996.
- [4] E. Goffman, *Behavior in public places; notes on the social organization of gatherings*. [New York]: Free Press of Glencoe, 1963.
- [5] A. Kendon, *Conducting Interaction: Patterns of behavior in focused encounters*. New York: Cambridge University Press, 1990.
- [6] S. Duncan, "On the structure of speaker-auditor interaction during speaking turns," *Language in Society*, vol. 3, pp. 161-180, 1974.
- [7] C. Goodwin, *Conversational Organization: Interaction between speakers and hearers*. New York: Academic Press, 1981.
- [8] R. M. Krauss and S. R. Fussell, "Constructing Shared Communicative Environments," in *Perspectives on Socially Shared Cognition*, L. B. Resnick, J. M. Levine, and S. D. Teasley, Eds. Washington: American Psychological Association, 1991, pp. 172-200.
- [9] J. Allen, *Natural Language Understanding*. Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc., 1995.
- [10] G. Brown and G. Yule, *Discourse Analysis*. Cambridge: Cambridge University Press, 1983.
- [11] H. H. Clark and S. E. Brennan, "Grounding in Communication," in *Perspectives on Socially Shared Cognition*, L. B. Resnick, J. M. Levine, and S. D. Teasley, Eds. Washington: American Psychological Association, 1991, pp. 127-149.
- [12] L. Polanyi, "A Formal Model of the Structure of Discourse," *Journal of Pragmatics*, vol. 12, pp. 601-638, 1988.
- [13] M. S. Cary, "The Role of Gaze in the Initiation of Conversation," *Social Psychology*, vol. 41, pp. 269-271, 1978.
- [14] H. M. Rosenfeld, "Conversational Control Functions of Nonverbal Behavior," in *Nonverbal Behavior and Communication*, A. W. Siegman and S. Feldstein, Eds., 2nd ed. Hillsdale: Lawrence Erlbaum Associates, Inc., 1987, pp. 563-601.
- [15] N. Chovil, "Discourse-Oriented Facial Displays in Conversation," *Research on Language and Social Interaction*, vol. 25, pp. 163-194, 1991.
- [16] D. McNeill, *Hand and Mind*. Chicago and London: The University of Chicago Press, 1992.
- [17] A. Kendon, "The negotiation of context in face-to-face interaction," in *Rethinking context: language as interactive phenomenon*, A. Duranti and C. Goodwin, Eds. New York: Cambridge University Press, 1990, pp. 323-334.
- [18] J. B. Bavelas, N. Chovil, L. Coates, and L. Roe, "Gestures Specialized for Dialogue," *Personality and Social Psychology*, vol. 21, pp. 394-405, 1995.
- [19] C. C. Werry, "Linguistic and Interactional Features of Internet Relay Chat," in *Computer-Mediated Communication: Linguistic, Social, and Cross-Cultural Perspective*, S. C. Herring, Ed. Amsterdam: John Benjamins, 1996, pp. 47-63.

- [20] A. Garcia and J. B. Jacobs, "The Interactional Organization of Computer Mediated Communication in the College Classroom," *Qualitative Sociology*, vol. 21, pp. 299-317, 1998.
- [21] D. Vronay, M. Smith, and S. M. Drucker, "Alternative Interfaces for Chat," presented at UIST'99, Asheville, NC, 1999.
- [22] M. Smith, J. J. Cadiz, and B. Burkhalter, "Conversation trees and threaded chats," presented at Computer Supported Cooperative Work, 2000.
- [23] J. McCarthy, Miles, V., Monk, A., Harrison, M., Dix, A., Wright, P., "Text-based on-line conferencing: a conceptual and empirical analysis using a minimal prototype," *Human-Computer Interaction*, vol. 8, pp. 147-183, 1993.
- [24] M. C. Russell and C. G. Halcomb, "Bringing the Chat Room to the Classroom," *Usability News*, vol. 4, 2002.
- [25] L. Cherny, *Conversation and community : chat in a virtual world*. Stanford, Calif.: CSLI Publications, 1999.
- [26] D. D. Suthers, "Collaborative Representations: Supporting Face to Face and Online Knowledge-building Discourse," presented at 34th Hawai'i International Conference on the System Sciences, Maui, Hawai'i, 2001.
- [27] J. Donath, "A Semantic Approach To Visualizing Online Conversations," *Communications of the ACM*, vol. 45, pp. 45-49, 2002.
- [28] L. Damianos, J. Drury, T. Fanderclai, L. Hirschman, J. Kurtz, and B. Oshika, "Evaluating Multi-party Multi-modal Systems," MITRE, Technical October 2000 2000.
- [29] L. Cherny, "The MUD Register: Conversational Modes of Action in a Text-Based Virtual Reality," in *Linguistics: Stanford University*, 1995.
- [30] S. Whittaker and B. O'Conaill, "The Role of Vision in Face-to-Face and Mediated Communication," in *Video-Mediated Communication*, K. Finn, A. Sellen, and S. Wilbur, Eds.: Lawrence Erlbaum Associates, Inc., 1997, pp. 23-49.
- [31] R. Vertegaal, "The GAZE Groupware System: Mediating Joint Attention in Multiparty Communication and Collaboration," presented at CHI'99, Pittsburgh, PA, 1999.
- [32] M. J. Taylor and S. M. Rowe, "Gaze Communication using Semantically Consistent Spaces," presented at CHI 2000, The Hague, The Netherlands, 2000.
- [33] M. Kobayashi and H. Ishii, "ClearBoard: A Novel Shared Drawing Medium that Supports Gaze Awareness in Remote Collaboration," *IEICE Transactions on Communications*, vol. E76-B, pp. 609-617, 1993.
- [34] J. Cassell and H. Vilhjalmsson, "Fully Embodied Conversational Avatars: Making Communicative Behaviors Autonomous," *Autonomous Agents and Multi-Agent Systems*, vol. 2, pp. 45-64, 1999.
- [35] A. R. Colburn, M. F. Cohen, S. M. Drucker, S. LeeTiernan, and A. Gupta, "Graphical Enhancements for Voice Only Conference Calls," Microsoft Corporation, Redmond, WA, Technical Report MSR-TR-2001-95, October 1, 2001 2001.
- [36] M. Garau, M. Slater, S. Bee, and M. Angela Sasse, "The Impact of Eye Gaze on Communication using Humanoid Avatars," presented at CHI 2001, Seattle, WA, 2001.
- [37] R. Vertegaal and Y. Ding, "Explaining Effects of Eye Gaze on Mediated Group Conversations: Amount or Synchronization," presented at CSCW 2002, New Orleans, LA, 2002.
- [38] J. Cassell, H. Vilhjalmsson, and T. Bickmore, "BEAT: the Behavior Expression Animation Toolkit," presented at SIGGRAPH01, Los Angeles, CA, 2001.
- [39] L. Hiyakumoto, S. Prevost, and J. Cassell, "Semantic and Discourse Information for Text-to-Speech Intonation," presented at ACL Workshop on Concept-to-Speech Technology, 1997.
- [40] E. P. Prince, "Toward a Taxonomy of Given-New Information," in *Radical Pragmatics*, Cole, Ed.: Academic Press, 1981, pp. 223-255.
- [41] O. E. Torres, J. Cassell, and S. Prevost, "Modeling Gaze Behavior as a Function of Discourse Structure," presented at First International Workshop on Human-Computer Conversation, 1997.
- [42] M. Kipp, "Anvil - A Generic Annotation Tool for Multimodal Dialogue," presented at Eurospeech, Aalborg, 2001.
- [43] T. Erickson and W. A. Kellogg, "Social Translucence: An Approach to Designing Systems that Support Social Process," *ACM Transactions on Computer-Human Interaction*, vol. 7, pp. 59-83, 2000.
- [44] F. Viegas and J. Donath, "Chat Circles," *Proceedings of CHI'99*, pp. 9-16, 1999.