Representing Coordination and Non-Coordination in an American Sign Language Animation

Matt Huenerfauth
Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
+1-215-898-8630

matt@huenerfauth.com

ABSTRACT

While strings and syntax trees are used by the Natural Language Processing community to represent the structure of spoken languages, these encodings are difficult to adapt to a signed language like American Sign Language (ASL). In particular, the multichannel nature of an ASL performance makes it difficult to encode in a linear single-channel string. This paper will introduce the Partition/Constitute (P/C) Formalism, a new method of computationally representing a linguistic signal containing multiple channels. The formalism allows coordination and non-coordination relationships to be encoded between different portions of a signal. The P/C formalism will be compared to representations used in related research in gesture animation. The way in which P/C is used by this project to build an English-to-ASL machine translation system will also be discussed.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *language generation, machine translation*; K.4.2 [**Computers and Society**]: Social Issues – *assistive technologies for persons with disabilities.*

General Terms

Design.

Keywords

American Sign Language, Multimodal Generation, Accessibility Technology for the Deaf, Gesture Generation.

1. INTRODUCTION

American Sign Language (ASL) is a full natural language – with a linguistic structure distinct from English – used as the primary means of communication for approximately one half million deaf people in the United States [14] [11] [15]. Due to limited exposure to spoken language during childhood, many deaf people find it difficult to read English text. In fact, the majority of deaf U.S. high school graduates (age 18) have only a fourth-grade (age 10) English reading level [6]. Technology for the deaf rarely addresses this literacy issue. Software for translating English text into animations of a computer-generated character performing ASL can make a variety of English text sources accessible to the deaf, including: TV closed captioning, teletype telephones, and computer interfaces [9]. English-to-ASL machine translation (MT) software can also be used in educational software for deaf children to improve their literacy skills. Instead of onscreen English text, the software would produce an animation of a human character performing ASL (specifically, it would translate English text into a detailed script to control a 3D character previously built by graphics researchers). Unfortunately, few Natural Language Processing (NLP) researchers have tried to build English-to-ASL systems [7] [9]. The visual/spatial properties of ASL make it difficult to encode using traditional NLP software.

1.1 Writing Systems and NLP Research

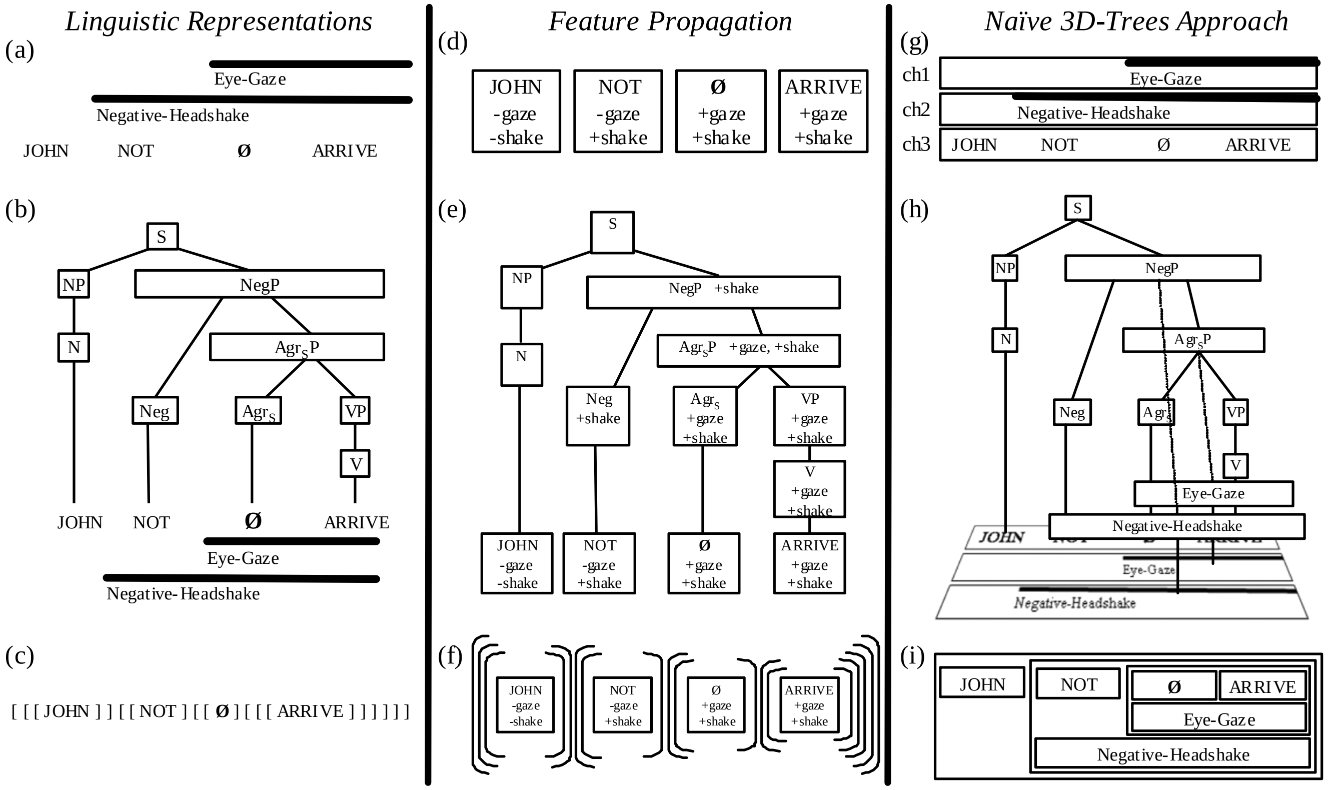
A language performance (even for a spoken language) contains several parallel streams of information: in addition to the spoken words, the signal includes facial expression, hand gestures, eye gaze, and other vocal data (prosody, volume, and pitch). These channels of the signal are time-coordinated data streams; each is a set of values which change over time [8]. The writing system for English does not record most channels of this signal, and so text-based NLP systems can take advantage of their users' literacy skills to simplify their work. Instead of specifying a full performance (with gestures, prosody, facial expression, etc.), they only need to generate a string in a writing system.

Because ASL is a language without a conventional writing system, an ASL generator cannot make this simplification. With no written form, the generator must specify the values for each channel of an ASL performance: hand locations, hand shapes, hand orientations, eye gaze, head-tilt, shoulder-tilt, body posture, and facial expression (all of which convey meaning in ASL). Inventing an ASL writing system doesn't solve this problem – without users trained in this writing system, the generator could not use it as output. It would still need to build a full animation.

This paper will show how there are also problems with using string-like representations inside of an ASL NLP system – even if we eventually convert the representation into animation before showing it to users. Specifically, the single-channel nature of strings tends to over-synchronize an ASL animation specification. Thus, we will propose a representation that encodes parallel channels of an ASL performance: the Partition/Constitute (P/C) Formalism. This formalism was

ASSETS'05, October 9–12, 2005, Baltimore, Maryland, USA.

Figure 1: Traditional linguistic representations of an ASL sentence (a-c) and two problematic NLP encodings of it (d-f, g-i).



developed as part of an English-to-ASL machine translation project [7] [8] [9]. P/C is the formal underpinning of the representations used in our software.

1.2 Organization of this Paper

An example ASL sentence will be shown using the string-like and tree-like notations used by ASL linguists. Two naïve ways to encode the sentence will be attempted, and failings of each will be discussed. The P/C formalism will be introduced as a solution to these problems, and subtleties in its use will be outlined. Since an important part of our project is the generation of some ASL phenomena called classifier predicates, a P/C representation of these is shown toward the end of the paper. Finally, P/C will be compared to some previous representations.

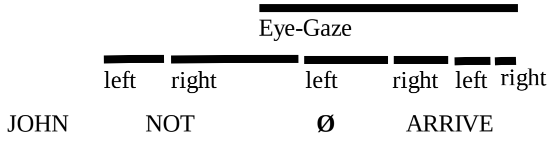
2. HOW LINGUISTS REPRESENT ASL

Linguists studying spoken languages often use string notations, but ASL linguists generally prefer a notation that records the multichannel nature of the signal. Figure 1(a) is a sentence written in the "decorated string" notation used by ASL researchers. During this sentence, signers produce three signs with their hands: JOHN, NOT, and ARRIVE. They shake their head in a negative manner during the portion of the sentence under the "Negative-Headshake" bar. In ASL, objects under discussion can be associated with locations in space around the signer. During the part of the sentence under the "Eye-Gaze" bar, the signer looks at a location previously associated with JOHN. (Eye gaze can indicate subject agreement for intransitive verbs [15].)

The glosses (words) in the figure are not a writing system – ASL has no written form. The glosses are used by linguists to record the activity of an ASL signer's hands, and dark bars represent information not conveyed by the hands – non-manual signals (NMS). The bars "decorate" the string. The notation uses a "null symbol" (Ø) to act as a placeholder for a linguistic unit that does not produce any performance with the signer's hands. In this example, the use of the Ø indicates that the Eye-Gaze starts a moment before the beginning of the sign ARRIVE.

In the example, there is non-coordination between the headshake and the string of words: A headshake consists of a series of individual left and right head movements (see Figure 2). The notation does not need to encode how each movement should be coordinated with each sign in the sentence performance. In fact, minor variations in the timing of these individual left and right movements (relative to the manual signs) would not produce different meanings.

Figure 2: The Left and Right Movements of the Head



Since we would like to develop a representation of the internal structure of an ASL signal, it is useful to consider ASL syntactic tree representations used in the linguistic literature. Consider the syntax notation for this sentence in Figure 1(b).

(This figure contains a simplified version of an analysis in the style of [15]; the linguistic details are not important for the way the example is used in this paper.) The tree explains the arrangement of the manual signs, but it doesn't indicate how the NMS bars are linked to them. Since trees are a graphical way to represent a nested structure for a text string, consider how the tree can be represented as a (one-dimensional) bracketing structure in Figure 1(c). The NMS bars would extend beyond the constituents in the brackets – it's not clear how the bars fit into the notation. While researchers have used decorated strings and syntax trees to make great strides in ASL linguistics [15], an ASL animation system needs a more precise representation of the coordination relationships.

3. STRINGS & FEATURE PROPOGATION

Previous ASL NLP systems have internally encoded the ASL signal as a string of glosses, which represent the individual signs to perform [7]. Most also use a traditional grammar modification called "feature propagation" – they associate values with nodes in the syntax tree, and these "features" spread their value from parents down to children to "propagate" information through the tree. In Figure 1(d-f), one feature [+shake] indicates headshake is occurring and another [+gaze] indicates eye-gaze. These features are passed down to the individual signs at the leaves of the tree. Note how the AgrSP node passes [+shake] from its parent to its children and adds [+gaze] to all its descendents.

There are problems with this approach – it splits the representation of the NMS into individual events for each sign. The ASL performance is treated like a string of beads on a chain – the boundary between each sign acts like a global synchronization point across all of the ASL channels. The notation doesn't accurately represent the coordination relationships in the signal. We discussed above how the Negative-Headshake movements are not coordinated with the boundaries of individual signs, yet this notation implies that there are individual headshake events coordinated with each sign. While the animation output system could merge these headshake features together into a single multi-sign event, the representation doesn't indicate when it's allowed to merge features across signs (and when it cannot). By over-coordinating the signal, these systems produce an overly constrained specification script of the movements required of the animated virtual human character. Introducing unnecessary constraints into this script could make the character's already-difficult requirements too hard to be performed successfully.

4. A NAÏVE APPROACH: 3D TREES

We would like a representation that does not break apart NMS events into small pieces, yet we would like to link the NMS events to the tree structure. One way to do this is to represent each channel of the signal as its own string, see Figure 1 (g-i). To represent the structure of all three strings in parallel, we must use a three-dimensional tree – Figure 1(h). Some branches in the tree move out of the page toward the reader – the dotted lines in the figure. In fact, the nodes at the top of the tree (S, AgrP, NegP) are not two dimensional as they might appear. They should be visualized as 3D "blocks" that cover several channels.

The tree image itself is less important than the bracketing information it captures. When viewed from above, the 3D-tree looks like the two-dimensional bracketing structure in Figure 1(i). Time is shown in the horizontal dimension, and the channels are represented in the vertical dimension. (Since they're easier to read, bracket diagrams will be used for the rest of this paper.) The entire sentence is contained inside of a single rectangle that corresponds to the S-node in the tree. It spans the entire sentence left-to-right and specifies the sentence performance across all of the channels (top-to-bottom). To the right of the JOHN box, there is a large rectangle containing the rest of the sentence; this is the NegP node. When nodes covering several channels split into children, each child can cover a subset of the channels covered by their parent. For instance, the NegP node assigns its AgrSP child to the top two channels and its Negative-Headshake child to the bottom channel.

There has been previous theoretical work on the definition of tree structures that can branch in multiple dimensions [2] and grammars to generate them [13] [17]. These grammars have been used to specify the structure of visual languages [13], a term used to refer to systematic 2D diagrams that communicate information: flow charts, state diagrams, process diagrams, etc. (While ASL is a human language that is quite visual, it is not what is meant by this term.) These grammars are multidimensional; they can produce structures are not just linear (one-dimensional) strings. The rules in these grammars allow nodes to break into multisets of unordered sub-nodes with constraints between them. [1]

In this paper, we will propose a linguistically motivated version of such a two-dimensional grammar that encodes a human language signal, not just an artificial language. Our grammar uses one dimension to represent time and the other to represent the channels in a signal. This novel application of multidimensional grammar to human language (and this use of a temporal and a channel dimension) yields a formalism that can encode the structure of a variety of multichannel language signals.

4.1 Problems with this Naïve Approach

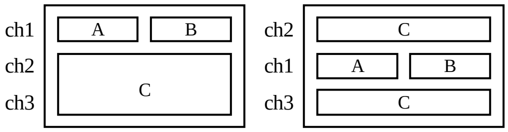
The naïve approach will be the basis of our new formalism; so, we carefully examine it here. Let's note what it does well. Figure 1(i) records the signal's multi-channel nature, represents how nodes break into children, and shows how responsibility for channels can be delegated to children of a node. Much like a traditional tree, the bracketing diagram breaks a signal into nested, non-overlapping components. These children nodes may divide their parent into left-to-right temporally sequential constituents and they may also assume responsibility for a subset of the top-to-bottom channels of the signal that are covered by their parent.

One problem with this approach is that it implies a sorting on the channels of the signal. Since the channels are laid out in a top-to-bottom fashion, the notation seems to imply that a total order has been defined between channels. This is not a linguistic claim that our formalism should force us to make. If the top-to-bottom layout of channels is arbitrary, then the notation must

[1] P/C (presented later) can be formulated as a special 2D instance of these grammars. Constituting rules could use constraints to enforce that their sub-nodes are ordered and adjacent; partitioning rules could enforce sub-nodes to cover their parent's channels in a non-overlapping way.

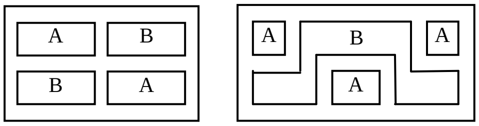
allow non-contiguous pieces of structure to belong to a single child (Figure 3). (Throughout this paper, we will manage to arrange the channels in our diagrams to avoid producing images that contain non-contiguous nodes; however, it is possible in this 3D tree approach for a single node to be non-contiguous in the top-to-bottom "channel" dimension of a diagram.)

Figure 3: Equivalent Diagrams with Channels Reordered



Just because some non-contiguous structures may be needed, we don't necessarily want a formalism which will allow us to encode bizarrely-shaped nodes (i.e. nodes which represent linguistically implausible assignments of portions of the output channels to child nodes). For example, consider the unusual structures in Figure 4: these are not decompositions we need in our ASL system.

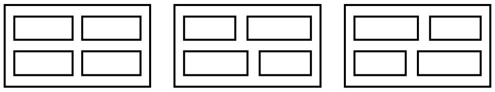
Figure 4: Linguistically Implausible Bracketing Diagrams



The bracketing structure in Figure 1(i) left some portions of the signal unspecified. (No node was assigned to some portion of some channels: consider the space below the JOHN node.) A better way to represent an unspecified part of the output signal would be to use a special null node (**Ø**). In this way, when a parent branches into children nodes, the children will completely cover the range of the parent – even if we have to insert some **Ø** nodes in order to do this.

Bracketing diagrams are not generally meant to indicate precise timing information (for instance, drawing a rectangle 3 millimeters to the right of another shouldn't indicate that one event happens 3 seconds after another). The diagrams are only meant to indicate linear ordering of phenomena and their nested structure. So, when a rectangle in a bracketing diagram breaks into children in both the left-to-right and top-to-bottom directions at the same time, then there may be a cross-channel temporal relationship that is left unspecified (see Figure 5). In this case, there are four children of the parent node: two on one channel and two on another. Unfortunately, it's not clear whether or not variations in the way we draw the diagram should be interpreted as specifying a temporal relationship between the "breaks" on each of the two channels. We can draw the break on the top channel to the left, to the right, or vertically aligned with the break on the bottom channel. Since we don't want the precise location of rectangles to indicate performance timings, then it seems awkward to interpret the left-to-right position of these breaks as meaningful. Further, how would we indicate that we don't care about the precise timing coordination of two changes on two different channels? Here, no matter how we draw the boxes, we seem to claim some temporal relationship.

Figure 5: Should these diagrams be interpreted differently?



We would prefer a multichannel representation that did not over-specify cross-channel coordination relationships (as this approach seems to do). If forced to specify temporal relationships that we don't really care about, then we may put too many artificial requirements on the performance of the ASL animation output. Such overspecification reduces the flexibility of the final graphics animation output module of our ASL system. We may produce a specification that is too difficult for the animated human character to perform. We would prefer a formalism that allows us to *optionally* specify the coordination relationships between events on different channels – so that we only specify temporal relationships we care about – and we avoid such problems.

5. P/C: A MULTICHANNEL FORMALISM

While the naïve approach captured the multichannel nature of the ASL signal, there were problems. It allowed us to leave portions of a diagram empty; we should use **Ø** nodes instead. We would like to avoid oddly-shaped child nodes (rectangles are preferred), but because channels may be arbitrarily ordered top-to-bottom in our diagram, we have to allow children nodes to be: *rectangles that have been sliced into horizontally parallel and identical-length pieces.* Finally, we would like an optional way to specify coordination relationships across channels of the signal.

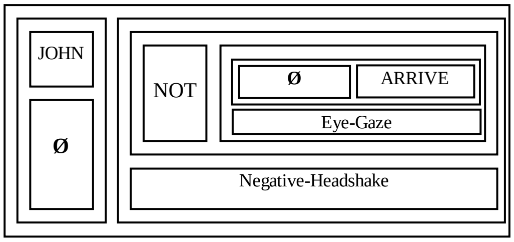
The naïve approach gives us too much flexibility in the possible structures it can describe. Our formalism should enforce more restrictions. We now require each rectangle to split in only one direction at a time. Further, we will require that the children of a rectangle "cover" all of the time (in the left-to-right dimension) and all of the channels (in the top-to-bottom dimension) of their parent in a non-overlapping way. This is our new multichannel ASL representation: The Partition/Constitute Formalism (P/C).

Since rectangles in the bracketing diagram are analogous to nodes in a 3D-tree structure, let's replace our "rectangle" terminology with "nodes" instead. When a node branches left-to-right, we will call it a constituting node, and we say that it has broken into constituents. The left-to-right ordering of constituents should be interpreted as specifying a temporal sequence for the sub-phenomena that compose their parent. Children cover the entire time range of their parent in a non-overlapping way. Constituting nodes are just like nodes in a traditional syntax tree, like Figure 1(b), where nodes break into sequential children.

Nodes branching top-to-bottom in the bracketing diagram (i.e. branching into-the-page and out-of-the-page in the 3D tree image) are called partitioning nodes, and we say that they have broken into partitions. A partitioning node breaking into children indicates a delegation of responsibility from the parent to each of its children. The set of channels covered by the parent is partitioned among all of the children in a non-overlapping manner. Each child is only allowed to specify/control those channels which it has been assigned by its parent. Since the order of channels in our diagrams is arbitrary (they are ordered to optimize readability), then the order of a partitioning node's children in a grammar rule should not be interpreted as meaningful. Further, if a child spans multiple channels that are not adjacent in the way the diagram was drawn, then it may not appear as a single contiguous rectangle. (We have managed to order the channels in our diagrams to avoid such nodes.)

A new P/C Formalism bracketing diagram of our sentence is shown in Figure 6; each rectangle splits left-to-right or top-to-bottom (but not both). Note that we could have also restricted the formalism so that nodes could only binary-branch. For now, we'll allow multi-branching nodes, but it is interesting to note that binary-branching would imply trivially that a node can only partition or only constitute – with only two children, it could only split in one direction.

Figure 6: A Partition/Constitute Bracketing Diagram



5.1 When to Partition? When to Constitute?

When generating a representation of an ASL animation, then we can constitute or partition at any step of the derivation. How do we decide when to partition and when to constitute? Why prefer one tree to another? We could trivially partition all of the channels at the root of the tree, or we could do all of the partitioning at the leaves. However, the resulting trees would not capture the conceptual decomposition of the multichannel signal that motivated this formalism. If we do all of the partitioning at the leaves, then we'd produce a structure that looks just like the single-channel "Feature Propagation" tree in Figure 1(e). If we do all of the partitioning at the root, then we'd have completely independent and unrelated tree structures for every channel of the signal. Clearly, channels should be related in some way during a signal; so, there must be a middle ground. Constituting and partitioning nodes should be interspersed throughout the tree.

Guideline 1: To break a phenomenon into sub-phenomena that occur in a temporal sequence, we use a constituting node. Just like nodes in traditional syntax trees break phrases into sub-phrases, a constituting node is broken into temporally sequential children that produce their parent.

Guideline 2: If information on two different channels shows coordination in stopping/starting/intermediate timing, then we should first constitute and then partition (to produce a structure like Figure 7(a). This produces gaps in the figure between horizontally adjacent rectangles (called coordination breaks), which serve as "mile-posts" or cross-channel synchronization points in the representation. For example, Figure 7(a) represents a two-channel signal with phenomena on each channel that begin, change, and end at the same time. We use a coordination break to capture the simultaneity between the changes in the two phenomena.

Figure 7: Specifying Coordination and Non-Coordination

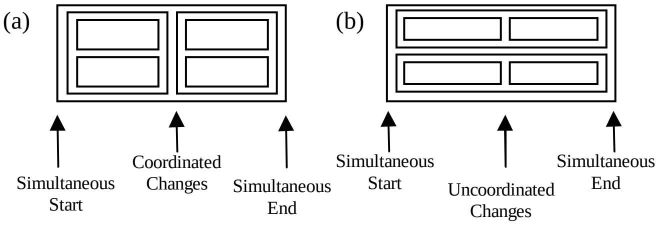


Figure 7(b) also represents a two-channel signal with phenomena that begin, change, and end; however, here the changes are not necessarily simultaneous. (The beginning and end are still coordinated.) While the "uncoordinated changes" are drawn such that they align horizontally, the diagram could have equivalently been drawn such that the boxes did not line up. Since the two channels do not have a coordination break across them, the changes may or may not align temporally during a performance. Since the relationship is not specified, the figure encodes several possible performances. We may not care to specify this relationship; it may not affect the meaning of the output.

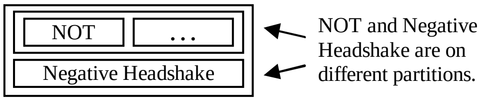
Guideline 3: When the timing between a signal on two channels is uncoordinated, arbitrary, or unspecified, then they should be assigned to different children of a partitioning node. Partitioning establishes a *coordination independence* between two channels of a signal. After partitioning, there is no guarantee that boundaries will align between two channels. During generation, if a portion of two channels have been assigned to different partitions, then the nodes lower in the tree structure should not need to know information (especially timing information) from nodes on a different partition to make generation choices or to produce output. By partitioning two channels before further decomposing them, we can encode that there is non-coordination between those two channels of the output. Thus, a syntax tree for a written string is just a special case of P/C notation that never divides a signal into partitions – the tree contains only constituting nodes. No coordination independence assumptions are made in the decomposition.

5.2 Determining the Channels in the Signal

While the previous section explored how to decide when to partition and when to constitute during the generation of a multichannel signal, but it did not explain how to determine the set of channels to best represent the signal. (This issue is important during the initial design of a generation system.) In a complex linguistic signal, like American Sign Language, parts of the body may be used to convey many different kinds of information. Selecting the best channel breakdown for the signal is not as easy as merely assigning the specification for each body part to a different channel. The linguistic literature is a good starting point for deciding how to represent a signal in a multichannel fashion, but we'll see that attempting to encode samples of the signal in P/C can suggest more fine-grained channel decompositions.

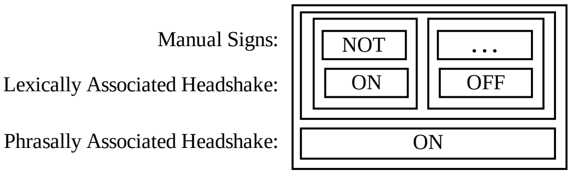
The ASL linguistic literature suggests ways to break the signal into channels. For instance, based on linguistic analyses of ASL, the headshake and head-tilt were assigned to different channels in Figure 6. However, there's a problem: a human signer will often use a stronger intensity of Negative Headshake during the part of the sentence that co-occurs with the ASL sign NOT. The headshake intensity decreases further away from the sign. The problem is that we placed Negative Headshake on a different partition from NOT (Figure 8). Since they were placed on different partitions of the signal, the timing of NOT should not be allowed to affect the intensity of Negative Headshake.

Figure 8: Relevant Sub-Structure from Figure 6



In many ASL sentences, the intensity of a non-manual signal (NMS) is affected by the timing of a sign (often the NMS is strongest during some sign). Since the individual left and right head shakes of the signer are not coordinated to the individual signs, we'd like to partition the signs and the NMS. So, how can we explain the intensity change? One solution is to split the headshake into two channels: one for the negative headshake associated with phrases and another for the headshake associated with individual signs (Figure 9). The change in headshake intensity is thus explained as an additive effect of combining the ON value of the Lexical and Phrasal channels. After the Lexical channel is OFF, the intensity fades to a lower level.

Figure 9: A More Fine-Grained Decomposition of Channels


Manual Signs:
Lexically Associated Headshake:
Phrasally Associated Headshake:
NOT
. . .
ON
OFF
ON

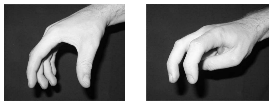
Thus, our attempt to represent the ASL sentence in the P/C notation has suggested a new way to break apart the channels of the signal. While this breakdown facilitates our building an ASL generation system, we do not claim that is has theoretical linguistic implications for ASL. Human ASL signers may or may not represent language this way, but it is a useful way to represent ASL in software. Indeed, the P/C notation and its breakdown of a signal into channels and partitions (with coordination independence assumptions between them) has been designed from an engineering perspective, not a theoretical linguistic one. While it may be useful to linguists, further study of human-produced ASL data would be required to determine if the formalism is necessary/sufficient for the representation of the linguistic structure of actual human-produced ASL sentences.

6. ASL CLASSIFIER PREDICATES

This English-to-ASL MT project also generates animations of ASL phenomena called classifier predicates. These phenomena are an important part of ASL, yet they have received little attention from NLP researchers [7] [9], partially because they have been difficult to represent. During a classifier predicate, signers use their hands to position, move, trace, or orient imaginary objects in front of them in space to indicate location, movement, shape, contour, or size of a corresponding real world entity. Signers use a meaningful ASL handshape chosen from a finite set of shapes based on characteristics of the entity described (whether it is a vehicle, human, animal, etc.) and what aspect of the entity is described (position, motion, etc). A classifier predicate is often preceded by a noun phrase indicating the entity whose motion will be depicted.

For example, the sentence "the cat sat next to the house" can be expressed using two classifier predicates (with a noun phrase preceding each). After performing the sign HOUSE, signers move their non-dominant hand in a "downC" handshape forward and slightly downward to a point in space in front of their torso where a miniature house is envisioned. Next, after making the ASL sign CAT, signers use their dominant hand in a "bentV" handshape to indicate a location where a miniature cat is envisioned. Generally, downC handshapes are used to indicate boxy objects, and bentV, stationary animals (Figure 10). Since the sign CAT only requires one hand, signers may choose to hold their non-dominant hand (in the downC handshape) at the house location during the performance of the sign CAT and the bentV classifier predicate for the cat's location.

Figure 10: The downC and bentV ASL Handshapes.



The way our MT software translates English sentences describing spatial layout/movement of objects into ASL performances (like the one described above) is discussed in [9]. Scene visualization software is used to produce a 3D model of the position of the objects discussed in the English sentence, and this data is used to select locations and motion paths for the signer's arms during the performance. The system builds a plan of individual arm motions, which are decomposed into smaller motions during generation. Before P/C, the project lacked a good representation of the complex coordination relationships in this ASL animation.

Figure 11: P/C Representation of ASL Classifier Predicates

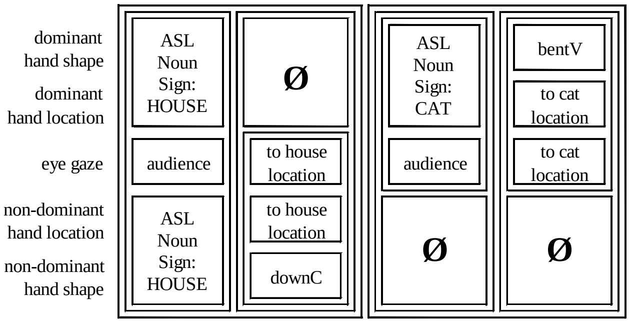


Figure 11 is a P/C representation of the classifier predicate performance described above. Eye-gaze, hand locations, and handshapes are represented as channels in the diagram. The HOUSE and CAT portions of the performance are different constituents (within which the classifier predicate and the noun are sub-constituents). While the start/end of eye and hand movements should aligned to the start/end of each classifier predicate, we don't care how these movements correspond to each other during the middle of a predicate. So, they are on different partitions inside of each classifier predicate constituent. To represent how we could optionally hold the non-dominant downC hand during the CAT performance, note how the CAT part of the signal has only **Ø** nodes for the non-dominant hand. The animation output module can be designed to optionally hold the last hand position if the subsequent specification is only **Ø**.

7. P/C VERSUS OTHER FORMALISMS

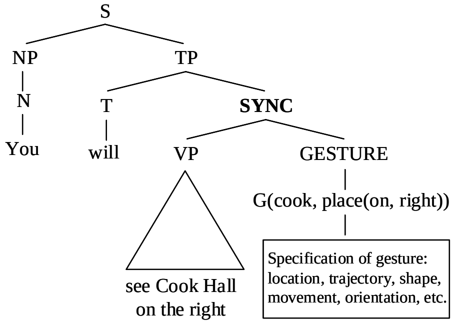
Sign language animation researchers have proposed various encodings of the script controlling the animated human [5] [7]. While the encodings record the character's movements, they do not record the hierarchical linguistic structure of the sentence nor how larger constituents in the sentence account for phenomena across channels. They are thus not useful as a sentence-level representation during generation. Other representations have been proposed for movements of actual humans during a performance. Signstream is a notation (and software tool) for recording parallel elements of an American Sign Language performance [16]. FORM [13], based on the Annotation Graph formalism [2], is a similar notation for recording non-linguistic gestures during a videotaped performance. Unfortunately, both were designed as annotation schemes, not as data structures to be used during natural language generation. They record precise timings of events; for example, they may record that someone raised an eyebrow 3.45 seconds after the start of a sentence. They don't specify which timing relationships must be coordinated and which were coincidental. They record the details of a single performance, not a set of all possible performances that would be grammatical output.

There are similarities between generating an ASL animation and generating an animation of a speaking character that performs gestures [8]. Both are instances of the same problem: generating a linguistic signal distributed across multiple channels. Gesture researchers have developed representations for coordinating gesture and speech; so, we should determine if they are useful for ASL. In gesture generation, the speech communicates most information to the user, and gestures convey additional content. During ASL generation, there is no audio channel; information is conveyed by the body only. To facilitate generation, the movements are broken into more fine-grained channels: the shape, orientation, and movement of each hand; the direction of head tilt, eye gaze, and shoulders; the shaking of the head; the eyebrow position; and other channels.

The NUMACK [10], REA [4], and BEAT [3] projects all used a similar formalism to coordinate gesture and speech: a tree with nodes representing gestures that should occur during the speech output. An example of such a tree structure for the sentence "You will see Cook Hall on the right" is shown in Figure 12 (this figure is adapted from one in [10]). While most of the branches in the tree indicate the temporal sequence of the words of the speech output, the SYNC node is special. It indicates that its two children should be performed at the same time. In this case, a gesture "G" will co-occur with the speech output of the phrase "see Cook Hall on the right." On first glance, SYNC looks like a partitioning node in the P/C formalism.

Unfortunately, SYNC trees are not expressive enough to encode a multichannel ASL animation. They do not record the internal structure of gestures; the "G" node cannot branch further into children. While some gestures may be simple enough to represent in this way, the movements needed for ASL animation are more complex. To represent the hierarchical structure of each channel, P/C thus allows partition and constituting nodes to be interleaved in a tree.

Figure 12: Speech/Gesture SYNC Tree (adapted from [10])


S
NP
TP
N
T
SYNC
You
will
VP
GESTURE
G(cook, place(on, right))
see Cook Hall on the right
Specification of gesture: location, trajectory, shape, movement, orientation, etc.

Another problem with using SYNC trees to encode ASL is that they do not model the signal channels at a sufficient level of detail. There is only a separation between the speech (the words) and gesture channels. If this were scaled up to represent the many channels of an ASL signal (by nesting SYNC nodes inside of each other or by allowing SYNC nodes to split into multiple children), then it would be difficult to ensure that no two nodes in the tree would try to modify the value of a channel at the same time. SYNC trees do not record which channels are covered by a particular node nor how responsibility for channels is delegated to children in a non-overlapping way. The P/C formalism prevents such conflicts; no two children of a partitioning node control the same channel of the signal.

While SYNC trees have proven to be a sufficient representation for the gestures produced by the REA, NUMACK, and BEAT projects, they are insufficient for ASL. The P/C formalism is more appropriate. Further, P/C may be of interest to future gesture projects that wish to produce more complex gestures (with internal structure or a more detailed decomposition of the animation channels).

8. CONCLUSIONS AND FUTURE WORK

The P/C formalism is an encoding scheme which allows a linguistic signal to be represented as several parallel streams of hierarchically-structured information. The formalism uses a two-dimensional grammar (with one dimension representing time and the other representing the channels of the signal), and it has the ability to record both temporal coordination and non-coordination relationships between portions of the signal across channels. These properties give the formalism greater expressivity than string-like encodings of complex linguistic signals (like those used by previous ASL MT systems), which tend to introduce extraneous temporal coordination relationships into the signal.

One limit on the expressivity of the formalism is that a node may either partition or constitute (but not both) inside of a P/C tree structure. Further, the children of a partitioning node are assumed to have no temporal coordination relationships between them. These properties of the formalism would simplify the design of any later animation module that must process a P/C tree structure and produce an animation of an ASL signer. The output module would not need to synchronize animation events on different channels of the signal that have been placed on different partitions of the tree. Animation representations which allow the encoding of parallel events whose internal sub-events are not temporally coordinated is a familiar design approach to computer graphics animation researchers; however, P/C's explicit representation of coordination and non-coordination for a multichannel linguistic signal (like an ASL animation) is novel.

P/C's significance from an accessibility perspective is that it helps bridge NLP technology (on written/spoken languages) to sign languages (like ASL) so that we can build new linguistic tools for deaf users. Specifically, it provides a way to encode languages that cannot be represented felicitously using a single-channel string-based approach. It uses tree-like data structures familiar to the NLP community to account for an ASL multichannel signal. P/C better encodes the temporal relationships in a signal than do previous representations available to NLP researchers. While P/C was developed for American Sign Language, the formalism should also be useful for representing the structure of other sign languages.

While this paper has shown how P/C better encodes an ASL linguistic signal than do simpler string-like representations, this paper has not claimed that the P/C formalism is suitable for use by linguists studying the structure of ASL. The formalism has been designed to be a better engineering approximation of the temporal relationships in an ASL animation than previously used approaches, and the design of the formalism has been motivated by both ASL-linguistic and animation considerations. While properties of the formalism may be of interest to linguists studying how to represent and analyze human-produced ASL, further study of ASL data would be required to determine if the formalism is sufficiently expressive to encode the structure of ASL produced by human signers.

The implementation of our English-to-ASL MT project is currently underway, and an evaluation study of a prototype version of the classifier predicate generation module is scheduled for the end of 2005. During the study, deaf native ASL signers will be asked to evaluate the ASL animations produced by the system to determine the quality of the ASL output our software design can achieve.

9. ACKNOWLEDGMENTS

This project is supported by NSF Universal Access Program Award #0520798 "SGER: Generating Animations of American Sign Language Classifier Predicates" and an NSF Graduate Research Fellowship to this author. I would like to thank my advisors Dr. Mitch Marcus and Dr. Martha Palmer for their guidance, discussion, and revisions during the preparation of this work. I would also like to thank Dr. Norman Badler for his help in obtaining funding for this research.

10. REFERENCES

[1] Baldwin, W., & Strawn, G. 1991. Multidimensional Trees. Theoretical Computer Science 84:2, pp293-311.

[2] Bird, S., & Lieberman, M. 2000. A Formal Framework for Linguistic Annotation. Speech Communication 33(1,2), 23-60.

[3] Cassell, J., Vilhjálmsson, H., Bickmore, T. 2001. BEAT: the Behavior Expression Animation Toolkit. SIGGRAPH '01, Los Angeles, CA, USA.

[4] Cassell, J., Stone, M, and Yan, H. 2000. Coordination and Context-Dependence in the Generation of Embodied Conversation. International Natural Language Generation Conference, Mitzpe Ramon, Israel.

[5] Elliott, R., Glauert, J., Jennings, V., and Kennaway, J. 2004. An Overview of the SiGML Notation and SiGML Signing Software System. Workshop on the Representation and Processing of Signed Languages, 4th Int'l Conf. on Language Resources and Evaluation.

[6] Holt, J. 1991. Demographic, Stanford Achievement Test - 8th Edition for Deaf and Hard of Hearing Students: Reading Comprehension Subgroup Results.

[7] Huenerfauth, M. 2003. Survey and Critique of ASL Natural Language Generation and Machine Translation Systems. Technical Report MS-CIS-03-32, Computer and Information Science, University of Pennsylvania.

[8] Huenerfauth, M. 2005. American Sign Language Generation: Multimodal NLG with Multiple Linguistic Channels. Association for Computational Linguistics, 43rd Annual Meeting, Student Research Workshop, Ann Arbor, MI, USA.

[9] Huenerfauth, M. 2005. American Sign Language Spatial Representations for an Accessible User-Interface. 3rd International Conference on Universal Access in Human-Computer Interaction. Las Vegas, NV, USA.

[10] Kopp, S., Tepper, P., and Cassell, J. 2004. Towards Integrated Microplanning of Language and Iconic Gesture for Multimodal Output. Int'l Conference on Multimodal Interfaces, State College, PA, USA.

[11] Liddell, S. 2003. Grammar, Gesture, and Meaning in American Sign Language. UK: Cambridge U. Press.

[12] Marriot, K. & Meyer, B. 1996. Towards a Hierarchy of Visual Languages. AVI'96 Workshop on the Theory of Visual Languages and Computing, 2:311-331.

[13] Martell, C. 2002. Form: An extensible, kinematically-based gesture annotation scheme. 3rd International Conference on Language Resources and Evaluation.

[14] Mitchell, R. 2004. How many deaf people are there in the United States. Gallaudet Research Institute, Graduate School and Professional Programs, Gallaudet University. June 28, 2004. http://gri.gallaudet.edu/Demographics/deaf-US.php

[15] Neidle, C., Kegl, J., MacLaughlin, D., Bahan, B., and Lee R.G. 2000. The Syntax of American Sign Language: Functional Categories and Hierarchical Structure. Cambridge, MA: The MIT Press.

[16] Neidle, C., Sclaroff, S., and Athitsos, V. 2001. SignStream™: A Tool for Linguistic and Computer Vision Research on Visual-Gestural Language Data. In Behavior Research Methods, Instruments, and Computers 33:3, 311-320

[17] Tucci, M., Vitiello, G., Costagliola, G. 1994. Parsing Nonlinear Languages. IEEE Trans. Software Eng., 20:9.