Spatial and Planning Models of ASL Classifier Predicates for Machine Translation

Matt Huenerfauth
Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104 USA
matt@huenerfauth.com

Abstract

A difficult aspect of the translation of English text into American Sign Language (ASL) animation has been the production of ASL phenomena called "classifier predicates." The complex way these expressions use the 3D space around the signer challenges traditional MT approaches. This paper presents new models for classifier predicates based on a 3D spatial representation and an animation planning formalism that facilitate a translation approach and are compatible with current linguistic accounts of these phenomena. This design can be incorporated into a multi-path architecture to build English-to-ASL MT systems capable of producing classifier predicates.

1. Introduction and Background

Although Deaf students in the U.S. and Canada are taught written English, their inability to hear spoken English results in most Deaf U.S. high school graduates (18+ years old) reading at a fourth-grade (10 year old) level (Holt, 1991). Unfortunately, many Deaf accessibility aids, like television closed captioning or teletype telephones, assume the user has strong English literacy skills. Many Deaf people with English reading difficulty are fluent in American Sign Language (ASL), and so an English-to-ASL MT system can make information and services accessible when English captioning text is at too high a reading level or a live interpreter is unavailable.

ASL is a natural language used by the half million Deaf people in the United States and Canada. The structure of ASL is quite different than English, and its visual modality allows it to use phenomena not seen in spoken language yet argued to be linguistic (Neidle et al., 2000; Liddell, 2003). In addition to using hands, facial expression, eye gaze, head tilt, and body posture to convey meaning, an ASL signer can use the surrounding space for communicative purposes. For example, signers can assign discourse entities locations in space and later refer to them by pointing to these locations. The locations are not meaningful topologically, i.e. positioning an entity to the left of another in space doesn't mean it is to the left of the other in the real world.

1.1. Classifier Predicates: A Spatially Complex Phenomena

Other ASL expressions are more complex in their use of space and position invisible objects around the signer to topologically indicate the layout of entities in a 3D scene being discussed. Constructions called "classifier predicates" allow signers to use their hands to position, move, trace, or re-orient an imaginary object in the space in front of them to indicate the location, movement, shape, contour, physical dimension, or some other property of a corresponding real world entity. This paper will focus on classifier predicates of movement and location (CPMLs) of entities in a 3D scene. CPMLs consist of a semantically meaningful handshape and a 3D arm movement path. A classifier handshape is chosen from a closed set 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 CPML is often preceded by a noun phrase indicating the entity whose locomotion will be depicted. (While not illustrated in this paper, a CPML's 3D path can be linguistically conventional rather than visually representational (Liddell, 2003); such predicates can also be handled by the linguistic models and MT approach discussed in section 4.)

For example, the sentence "the car parked between the cat and the house" can be expressed using a set of three CPMLs. After making the ASL sign *CAT*, a signer would move a hand in a "bent V" handshape (see Figure 1) forward and slightly downward to a point in space in front of his or her torso where an imaginary miniature cat could be envisioned. Next, after making the ASL sign *HOUSE*, the signer would make a similar motion with a "downward C" handshape to a place where a house could be envisioned. Finally, after signing *CAR*, the signer would place their dominant hand in a "sideways 3" handshape and trace a path in space to indicate the route taken by the vehicle. At the end of the motion (at a location between the 'cat' and the 'house'), the signer would position the open palm of their non-dominant hand. The dominant hand would end its motion by coming to rest atop the platform produced by the non-dominant hand. Generally, "bent V" handshapes are the classifier for stationary animals, "downward C" for boxy objects, and "sideways 3" for vehicles. As the example suggests, translation into an ASL classifier predicate is complex because of the productive and space-depicting nature of these expressions.

1.2. Previous Direct and Transfer ASL MT Architectures

Since ASL has no written form, there are currently insufficient parallel English-ASL corpora for stochastic MT approaches; so, English-to-ASL MT systems have used non-stochastic direct and transfer MT architectures. Graphics software is used to animate 3D virtual human characters to perform the signing output (generally a script written in a basic animation instruction set controls the characters' movements); therefore, these systems convert English text into a script directing the characters how to perform ASL. Direct systems have used word-to-sign dictionaries to produce Signed English (a non-ASL English-like form of signing), and transfer systems have handled more divergences to produce actual (but limited) ASL output (Huenerfauth, 2003).

Because these systems employed only traditional lexical and grammatical resources and because they made no attempt to model the spatial arrangement of objects in the 3D scene being discussed, they were unable to produce classifier predicates from spatial English text. Omitting these phenomena from the coverage of an English-to-ASL MT system is unsatisfactory for several reasons: (1) many English concepts lack a fluent ASL translation without classifier predicates, (2) these phenomena are common in native signing (ASL signers produce classifiers 1 to 17 times per minute, depending on genre) (Morford and MacFarlane, 2003), and (3) English/ASL translation pairs involving classifier predicates are quite structurally divergent (and would thus be particularly useful to translate for a Deaf user with limited English literacy skills).

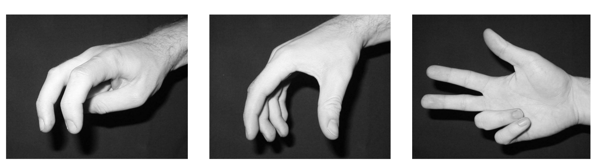

Figure 1: ASL Classifier Predicate Handshapes: "Bent V," "Downward C," and "Sideways 3"

2. A Multi-Path MT Architecture

This project's goal is to design an English-to-ASL MT system that can generate animations of classifier predicates of movement and location (CPMLs) from English sentences describing 3D scenes. There are two facets to this work: (1) a novel ASL representation and processing scheme that can generate CPMLs and (2) an overall MT software architecture design that allows this CPML-generator to be part of a complete English-to-ASL translation system that handles a variety of sentence types. The shape of this overall architecture will be described in this section, and the CPML generation component will be the focus of the remainder of this paper.

The discussion of classifier predicates above suggested that traditional direct and transfer MT approaches are insufficient for handling these phenomena. An MT approach with richer spatial and knowledge resources is needed to generate CPMLs, and this paper will later propose an interlingual translation design involving a 3D virtual reality representation of the entities under discussion. However, section 1.2 indicated that most non-classifier English-to-ASL sentence pairs can be translated using a simpler transfer design. For this reason, an English-to-ASL MT architecture has been proposed containing multiple processing pathways: interlingual, transfer, and direct (Huenerfauth, 2004a). The pathway for English inputs producing CPMLs includes the virtual reality software, but the pathway for other inputs uses a transfer MT approach (broader coverage and easier to implement). Finally, since many Deaf signers are somewhat familiar with non-ASL English-like forms of signing, this design also includes a direct pathway: a word-for-sign substitution process that produces a Signed English output. The system will process an input sentence using the most sophisticated pathway for which sufficient linguistic resources exist and "falls back" on simpler pathways as needed. This architecture for blending deep-knowledge and broad-coverage MT approaches in a single system should also be useful for other language pairs: especially for building hybrid stochastic/linguistic systems or for translating texts in which certain sentences require special processing (Huenerfauth, 2004a).

3. A Representation of 3D Locomotion for English-to-ASL MT

The design uses virtual reality software to calculate and store a model of the 3D coordinates of entities discussed in a spatially descriptive English text. The animated ASL signing character later uses this 3D coordinate data to calculate the arm movements needed to produce a classifier predicate to describe each entity's 3D location or movement. The 3D model is analogous to the miniature invisible objects one can imagine floating in space around ASL signers when they use classifier predicates to describe a 3D scene. To decide the locations and movement paths of the objects in the model based on the English input text, this design will use the AnimNL system (Bindiganavale et al., 2000). This software accepts English text containing instructions for a set of 3D virtual reality objects to follow, and it moves the objects in the 3D scene to obey the English input sentences. AnimNL has been implemented for military training and equipment repair domains, and it can be extended to new domains by augmenting its library of Parameterized Action Representations (PARs), to cover additional English spatial verbs.

PARs are feature-value structures that record the details of a locomotion event necessary for the AnimNL planning algorithm to generate a virtual reality animation; they have slots specifying: what agent is moving, the path/manner and translational/rotational nature of this motion, speed and timing data, terminating conditions, and planning operator slots like preconditions, effects, etc. The system stores a database of PAR templates that represent prototypical actions that entities can perform. These templates are missing particular details (some of their slots aren't filled in) about the position of the agent or other entities in the environment that would affect how the animation action should really be performed in particular situations. By parameterizing PARs on the 3D coordinates of the objects participating in the movement, the system can produce animations specific to particular 3D scene configurations and reuse animation code.

AnimNL operates by first analyzing an English input sentence, then it uses lexical and semantic features from the analysis to select and partially fill a PAR template from the database, and finally the PAR serves as an initial operator to a hierarchical animation planning process. A single locomotion event may contain several sub-movements or sub-events, and for this reason, PARs may be defined in a hierarchical manner. AnimNL's planner calculates the preconditions and effects of particular animation movements in order to produce a realistic virtual reality scene. This virtual reality serves as an intermediary between English sentences about 3D locations and movement and the ASL CPMLs they produce; in fact, it can be regarded as an interlingua for texts in this 3D locomotion domain (Huenerfauth, 2004a).

4. Models for CPML Generation

After calculating the 3D layout of the entities discussed in an English text, an approach is needed to generate animations of CPMLs describing this scene. We have argued (Huenerfauth, 2004b) that a recent linguistic model of classifier predicate generation (Liddell 2003) can serve as a starting point for developing such an approach. In Liddell's model, signers have a mental image of a scene to be discussed (much like a virtual reality) which they map (or "blend") onto the space around their body, and they use 3D information from this scene to select and fill templates for a classifier predicate from a template lexicon. Unfortunately, Liddell (2003) does not provide detail about the internal structure of these templates nor the exact selection/filling process. The remainder of this paper describes new computational models for classifier predicate generation within an English-to-ASL MT system that formalize and implement this linguistic account.

4.1. A Spatial Model of CPML Semantics: "Ghosts"

The way in which this system implements the blending of the virtual reality scene onto the space around the ASL signing character is by instantiating miniature floating invisible versions of the objects in the virtual reality scene in front of the torso of the signer. The *ghosts* in this invisible world model topologically correspond to the placement of the objects in the virtual reality; however, the ghosts don't need to be precise visual depictions of the entities discussed in the original English text – they are always invisible. The ghosts merely serve as rough placeholders for the 3D locations and orientations over time of the objects they represent; they also record the set of appropriate ASL classifier handshapes that could be used to refer to those entities. Within this model, we can regard the purpose of an ASL signer producing CPMLs as an attempt to convey information about what the invisible ghosts are doing in space. The 3D model of these ghosts over time can thus serve as a loose semantic representation of a set of CPMLs; in this light, the three CPMLs in section 1.1 can be thought of as conveying that a box-like object and a stationary animal occupied two points in space and a vehicle object came to rest in between them.

4.2. A Spatial Model of CPML Phonology: "Articulators"

To convey information about the invisible ghosts in space, the signer manipulates various visible *articulators* in that same region of space. These articulators include the dominant hand, the non-dominant hand, the eye-gaze, the head-tilt, facial expression, and shoulder tilt. (These last two are not discussed in this paper.) Because this CPML generation system will overlay a 3D coordinate system in the space near the signer's torso in order to position the ghosts, we can take advantage of this coordinate system to efficiently model the articulators of a CPML, as described below.

All phonological models of ASL record how the handshape, hand location, hand orientation, movement, and non-manual elements of a signing performance change over time (Coulter, 1993); however, current models are not well-suited to the representation of CPMLs. They typically record hand location relative to other parts of a signer's body; so, signs like CPMLs that occur in the "neutral space" in front of the torso require precise 3D coordinates of location in order to be modeled successfully. Unfortunately, it's not clear from these models how a generation process would produce the stream of 3D coordinates over time needed to specify a CPML. Hand orientation is often modeled as a set of cardinal or body-relative directions (which is insufficient for the variety of hand orientations seen in CPMLs), and even those models that do specify orientation more precisely do not account for how a generation process for CPMLs could calculate this information. Finally, these models specify handshape information at a finer granularity than necessary for CPMLs, which typically involve only a small set of handshapes.

At the other end of the spectrum, a non-linguistic (or perhaps a phonetic) representation of an animation of a 3D virtual human character performing a CPML would need to record a tremendous number of parameters over time: all of the joint angles for the face, eyes, neck, shoulders, elbows, wrists, fingers, etc. If we had to generate classifier predicates while considering all of these values, the task would be quite difficult. The goal of this model is to reduce the number of independent parameters needed to be specified by the generation process while still allowing us to produce a complete animation, and so we model CPMLs as a complex, coordinated movement (or dance) of a small set of 3D objects in the space in front of the signer.

Specifically, eye-gaze and head-tilt can be represented as a pair of 3D points in space at which these articulators are aimed. Since these points will often track a ghost in the signing space (or may aim at the audience), the model has a method of calculating their values. We can make this simplification because what is semantically meaningful in a CPML about eye-gaze and head-tilt is the point at which they are aimed, not the exact details of neck or eyeball angles. Fortunately, the animation software to be used by this system can calculate head/eye positions for a virtual character given a 3D point in space; so, this model is sufficient for producing an animation.

In a CPML, the position of the hand (not the whole arm) is semantically meaningful; so, the model can make another simplification. The location in space of the dominant and non-dominant hand are recorded as another pair of 3D coordinates. (We also record for each hand the 3D orientation and which of the classifier handshapes should be used.) Given hand location and orientation values, there are algorithms for calculating realistic elbow/shoulder angles for a 3D virtual human character (Liu, 2003); so, the model is again sufficient for generating animation.

4.3. A Planning-Based Model of CPML Templates: "CP-PARs"

In the previous sections, we have specified semantic and phonological models for CPMLs; in doing so, we have implicitly framed the CPML-generation problem as the task of creating a 3D movement script for the articulators based on the state of the invisible ghost model and the original English sentence. Continuing our implementation of Liddell's (2003) linguistic model, the next computational representation needed is for his classifier predicate "template lexicon." This lexicon will be implemented using a novel ASL representation based on the PAR animation planning formalism (Section 3). PARs were used previously by the AnimNL software, and the motivation for basing this new representation on them is that the problem of planning the movements of CPML articulators is analogous to AnimNL's problem of planning the movements of 3D virtual reality objects from an English input text.

Just as AnimNL stored a database of PAR templates of potential locomotion actions of 3D virtual reality objects, this MT system will store a database of PAR-like planning operators specifying potential CPML arm movements. The AnimNL system demonstrated that PARs are a robust mechanism for 3D animation planning; in fact, they represent information (e.g. about timing, manner, and purpose) that is more sophisticated than is required for planning CPMLs. Therefore, a reduced form of PARs will be implemented called "classifier predicate PARs" or *CP-PARs* that include those PAR fields that are relevant for planning CPML articulator movements. Just as AnimNL used lexical, semantic, and argument structure information from the analyzed English sentence to select and fill PAR templates, this system will link English motion verbs to particular CP-PAR templates and use the analyzed English sentence to select and fill a CP-PAR template.

Just like PARs, these CP-PARs are parameterized on the 3D coordinates of entities in a 3D scene – in this case, on 3D coordinates of the ghosts or other articulators. For example, a CP-PAR may be associated with the English verb "park" and may specify the articulator movements required to indicate a vehicle is parking (using the "sideways 3" handshape and non-dominant hand platform described in section 1.1). However, the CP-PAR would be parameterized on two pieces of information: the initial and final 3D locations of the ghost whose parking is being conveyed. So, the same CP-PAR template could be used to generate "parking" CPMLs at many 3D coordinates. Since AnimNL would have already calculated the 3D layout of the ghosts, the CP-PAR can simply use these 3D ghost coordinates when selecting the 3D locations of the articulators.

Finally, the filled-in CP-PARs will be processed by a hierarchical planning algorithm (much like AnimNL) to produce a set of detailed animation specifications for the articulators. The planner can consider: the locations of ghosts and articulators, lexical/semantic information from the English sentence, and the CPML discourse model (see next section). The goals, preconditions, and effects of the CP-PAR planning operators will allow them to trigger or modify additional classifier predicates that may be required to satisfy ASL grammatical constraints. Because CP-PARs (like PARs) are hierarchical planning operators, a single CP-PAR could be decomposed into sub-plans with temporal relationships between them. This mechanism can facilitate the coordination of multiple articulators during the performance of a single classifier predicate, and it enables this model to optionally pre-compile the syntax of complex interactions between multiple classifier predicates and store them as a sub-plan graph of a large multi-predicate CP-PAR.

While it is easier to develop classifier predicate templates using the concise CP-PAR formalism than the more verbose/robust PAR formalism, it is anticipated that the system will actually convert the CP-PARs into standard PARs prior to run-time so that the PAR-planning algorithm already implemented within the AnimNL software could be re-used as the processing engine for CPML generation. Since CP-PARs would typically contain less information than PARs, this conversion process should usually be rather simple. However, there is one regard in which CP-PARs will actually need to be a richer representation than PARs – CP-PARs must allow a developer to specify the location and orientation of the hand articulators over time in a more abstract manner. Generally, animation movement details in a PAR are not recorded in a symbolic fashion inside the attribute-value feature structure of the representation; instead, low-level animation programming code is associated with the PAR detailing how a 3D animation path is calculated based on the parameters inside the feature structure.

To facilitate the creation of CP-PARs by developers with linguistic (rather than graphics) expertise, it is important that the hand locations/orientations be easier to specify. Since the hand articulator typically tracks a ghost in the scene, a basic set of location/orientation calculation functions will be defined (that base their calculations on the 3D positions, paths, and orientations of a particular ghost over time). When CP-PARs are converted to PARs, the 3D animation code for the PAR would be created based on the 3D coordinates of the ghost over time and the calculation function that was chosen by the linguistic developer. For example, to specify a CP-PAR for "parking a car," the linguistic developer would just need to specify that the dominant hand should follow the 3D location of the ghost-car it is describing, and the front of the "sideways 3" handshape should be pointed in the direction of the motion path along which the car is traveling. To accommodate classifier predicates with unusual movement patterns, the system should also allow developers to program animation code directly if so desired.

4.4. A Model of CPML Discourse: "Identified(), Positioned(), Topicalized()"

During the analysis of the English input text, a reference resolution algorithm will track the entities mentioned in the text and maintain a list of the English text discourse referents. All of the entities that are assigned locations in the virtual reality model by the AnimNL software will also be added to a CPML discourse model. In addition to maintaining a list of these ghosts involved in the 3D scene, this CPML discourse model records whether certain discourse features are true for each entity. For example, one entity at a time can have the feature **topicalized(**entity**)** set to 'true'. Two other discourse features are more CPML-specific: **identified()** and **positioned()**. When an entity has been explicitly named using a noun phrase immediately prior to the performance of a CPML, then the feature **identified(**entity**)** will be set to 'true'. Performing an entity's noun phrase will also temporarily make the feature **topicalized(**entity**)** 'true'. When a previous classifier predicate has explicitly indicated a 3D coordinate in space for an entity, then the feature **positioned(**entity**)** is set 'true', and it remains 'true' until AnimNL moves that ghost in the virtual reality scene.

These discourse features are used during the CPML planning process; specifically, a precondition of most CPMLs is that the main ghost described satisfy the condition: **((identified(**entity**) and positioned(**entity**)) or topicalized(**entity**))**. The effect of this precondition is that when the signer performs a classifier predicate, it is clear which entity is being discussed. Either the entity is the current topic, or since a 3D location has explicitly been assigned to the entity in space, it is clear what entity is being referred to by a classifier predicate that begins at that 3D spatial coordinate. Most CPMLs will also require any other ghost involved in the predicate to satisfy the following: **(identified(**entity**) and positioned(**entity**))**. For example, we'll see in the next section that the classifier predicate for parking a car will require that any ghosts that the car is positioned *relative to* in space have been clearly identified and positioned by previous classifier predicates.

5. An Example of the CPML Models Used for Machine Translation

We will now illustrate how the system would process the English input sentence from Section 1.1: "The car parked between the cat and the house." The analyzer would parse the sentence, assign a word sense to the verb (e.g. park-23), and build a predicate argument structure with "the car" listed as the agent argument and "between the cat and the house" listed as a location adjunct. The system would identify the following list of discourse referents: (1) the car, (2) the cat, (3) the house. Finally, the AnimNL software would process this English sentence and build a 3D virtual reality scene portraying the information. It would select 3D locations for the cat and the house, calculate a location "between" them, select initial and final locations for the car, and create a 3D animation of the car's movement path. Since all three discourse entities were assigned locations in the virtual reality, all of them are added to the CPML discourse model. Initially, the discourse predicates **identified()** and **positioned()** would be 'false' for all three of the entities in the scene.

The initial goal for our CPML planning process is to express the semantics of the English sentence; so, the PARKING-VEHICLE CP-PAR (Figure 2) is triggered via the express-semantics predicate in its "Effects" field. This maps "the car" ghost to the variable g0 inside this CP-PAR, and the CP-PAR checks the restriction that g0 is a vehicle. The noun phrases inside the locative adjunct ("the cat" and "the house") are mapped to g1 and g2, respectively. Before performing PARKING-VEHICLE, the preconditions must be satisfied; so, they are placed on the list of goals. These **positioned(**g1**)** and **positioned(**g2**)** goals will trigger two additional CPMLs to precede PARKING-VEHICLE, namely LOCATE-STATIONARY-ANIMAL and LOCATE-BOXY-OBJECT. The **identified()** and **topicalized()** preconditions of all three of these CPMLs will trigger an appropriate instance of MAKE-NOUN-SIGN to precede each of them.

The CP-PARs in Figure 2 demonstrate how the location and orientation of the articulators can be calculated from the coordinates of ghosts using a set of 3D functions (note the functions inside the "Location" and "Orientation" fields of several of the CP-PARs). We can also see how the handshape of hand articulators can be specified symbolically using constants like "Sideways 3" or "Bent V." Finally, the example illustrates how parameters can be passed to CP-PARs (see the calls to PLATFORM and EYETRACK inside of PARKING-VEHICLE) and how multiple CP-PARs can be coordinated in sequence or concurrently. In this case, the control of the dominant hand, non-dominant hand, and eye-gaze articulators of the 'parking car' CPML is divided across three different CP-PARs that are performed simultaneously during the animation output.

At the end of the planning process, a sequence of CPMLs (with intervening ASL noun phrases) would be generated that is identical to the ASL output described at the end of Section 1.1.

PARKING-VEHICLE
Parameters: g0 *(ghost car parking)*, g1..gN *(other ghosts)*
Restrictions: g0 is a vehicle
Preconditions: topic(g0) or (ident(g0) and positioned(g0))
for g=g1..gN: (ident(g) and positioned(g))
Articulator: dom *(the dominant hand)*
Location: dom.loc = follow_location_of(g0)
Orientation: dom.ori = direction_of_motion_path(g0)
Handshape: dom.hs = "Sideways 3"
Effects: positioned(g0), topic(g0), express_semantics(
[park-23 agent=g0 locative_NPs=g1..gN])
Concurrently: PLATFORM(g0.loc.final), EYETRACK(g0)

PLATFORM
Parameters: loc0 *(the location at which to position upturned palm of non-dominant hand)*
Articulator: ndom *(the non-dominant hand)*
Location: ndom.loc = loc0
Orientation: ndom.ori = palm_upward()
Handshape: ndom.hs = "B" *(an open palm)*
This CP-PAR produces a horizontal platform with the non-dominant hand. In this example, it is performed concurrently with the PARKING-VEHICLE CP-PAR to produce a complete 'parking' classifier predicate.

LOCATE-STATIONARY-ANIMAL
Parameters: g0 *(the ghost to position in space)*
Restrictions: g0 is an animal
Preconditions: topic(g0) and ident(g0)
Articulator: dom *(the dominant hand)*
Location: dom.loc = move_to_location_of(g0)
Orientation: dom.ori = g0.ori
Handshape: dom.hs = "Bent V"
Effects: topic(g0), positioned(g0)
There is also a LOCATE-BOXY-OBJECT CP-PAR just like this one but using the "Downward C" handshape.

EYETRACK
Parameters: g0 *(ghost to follow with eye gaze)*
Articulator: eg *(the eye gaze articulator)*
Location: eg.loc = follow_location_of(g0)

MAKE-NOUN-SIGN
Parameters: g0 *(ghost whose noun sign to perform)*
Effects: topic(g0), ident(g0)
System looks up ASL sign(s) for the noun phrase that refers to g0, and it performs the sign(s).

Figure 2: Pseudo-code for the CP-PARs mentioned in the translation example of Section 5.

6. Discussion

This paper has illustrated how studying MT issues for ASL can push the boundaries of current MT methodologies and inspire linguistic understanding of ASL. Both the special difficulty in translating CPMLs and the familiarity many ASL signers have with Signed English motivated this system's exploration of a multi-pathway architecture for MT. The spatial nature of CPMLs motivated the integration of virtual reality software as an intermediary MT representation. The capabilities and requirements of the AnimNL software and the PAR planning formalism have motivated new computational models of CPML discourse, semantics (ghosts), templates (capturing morpho-syntactic composition of these predicates), and phonology (articulators). These MT models can serve as starting-points for further linguistic analyses of these phenomena.

This project is currently finishing specifying the CPML models, the CPML-generation approach (Huenerfauth, 2004b), and the multi-pathway machine translation architecture in which it will be situated (Huenerfauth, 2004a). Other research topics include: defining evaluation metrics for an MT system that produces ASL animation containing classifier predicates, finding ways of representing additional ASL phenomena using spatial or planning-based frameworks, planning the software implementation of the MT design, and beginning the construction of an initial CP-PAR lexicon of classifier predicate templates. This project will also consider ways of extending the current CP-PAR specification to address more complex classifier predicate generation issues, such as timing and duration of CPMLs, instantiation and placement of new objects in a scene, selection of scene perspective and scale, handling of manner/adverbials, and further specification of facial expression, body posture, hand location, and orientation within the CP-PAR framework.

7. Acknowledgements

I would like to thank my advisors Mitch Marcus and Martha Palmer for their guidance, discussion, and revisions during the preparation of this work.

8. References

Bindiganavale, Rama, William Schuler, Jan Allbeck, Norman Badler, Aravind Joshi, & Martha Palmer: 2000, 'Dynamically altering agent behaviors using natural language instructions.' In *4th International Conference on Autonomous Agents*.

Coulter, Geoffrey, ed.: 1993, *Phonetics and Phonology*. Vol. 3: Current Issues in American Sign Language Phonology. New York, San Francisco, London: Academic Press.

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

Huenerfauth, Matt: 2003, 'A survey and critique of American Sign Language natural language generation and machine translation systems.' Technical Report MS-CIS-03-32, Computer and Information Science, University of Pennsylvania.

Huenerfauth, Matt: 2004, 'A multi-path architecture for machine translation of English text into American Sign Language animation', in *Proceedings of the Student Workshop of the Human Language Technologies conference / North American chapter of the Association for Computational Linguistics annual meeting: HLT/NAACL 2004*, Boston, MA.

Huenerfauth, Matt: 2004, 'Spatial representation of classifier predicates for machine translation into American Sign Language', in *Workshop on the Representation and Processing of Signed Languages, 4th International Conference on Language Resources and Evaluation: LREC 2004*, Lisbon, Portugal.

Liddell, Scott: 2003, *Grammar, gesture, and meaning in American Sign Language:* Cambridge: Cambridge University Press.

Liu, Ying: 2003, 'Interactive reach planning for animated characters using hardware acceleration', Doctoral Dissertation, University of Pennsylvania.

Morford, Jill, & James MacFarlane: 2003, 'Frequency characteristics of American Sign Language', *Sign Language Studies*, 3:2.

Neidle, Carol, Judy Kegl, Dawn MacLaughlin, Benjamin Bahan, & Robert Lee: 2000, *The syntax of American Sign Language: Functional categories and hierarchical structure:* Cambridge, MA: The MIT Press.