Sutton-Slevinski Collaboration
Internet-Draft
Intended status: Informational
Expires: November 9, 2014

S. Slevinski
SignPuddle

May 8, 2014

# The SignPuddle Standard for SignWriting Text
## draft-slevinski-signwriting-text-03

Abstract

For concreteness, because the universal character set is not yet universal, and because an international standard for the internet community should be documented and stable, this I-D has been released with the intention of producing an RFC to document the character use and naming conventions of the SignWriting community on the Internet.

The SignWriting Script is an international standard for writing sign languages by hand or with computers. From education to research, from entertainment to religion, SignWriting has proven useful because people are using it to write signed languages. The SignWriting Script has two major families: Block Printing for the reader and Handwriting for the writer.

The SignWriting Text encoding model defines the structures of SignWriting Block Printing. The plain-text mathematical names are explained with tokens and regular expressions patterns. The visual image is supported with SVG and PNG generated by a SignWriting Icon Server. An experimental TrueType Font is available.

Formal SignWriting strings define a lite ASCII markup to name each sign logogram. The text is defined with regular expressions. The included query language defines several productive searching possibilities. The transformation from query language to regular expression is defined.

For Unicode, the current use of the Private Use Area font characters is documented. A character proposal for plane 1 is included that is isomorphic with the characters that are currently used by the community.

Three appendices discuss additional topics to the standard. The first discusses the Modern SignWriting theory and example document, stable since January 12, 2012. The second discusses the symbol encoding of the International SignWriting Alphabet 2010. The third discusses the SignPuddle Standards: licences, infrastructure, and compatibility.

This memo concretely defines a conceptual character encoding map for the Internet community. It is published for reference, examination, implementation, and evaluation. Distribution of this memo is unlimited.

## Status of This Memo

This Internet-Draft will expire on November 9, 2014.

## Copyright Notice

## Table of Contents

## 1. SignWriting Script

The SignWriting Script is the universal and complete solution for written sign language. It has been applied to a wide and deep international community of sign languages including: American Sign Language, Arabian Sign Languages, Australian Sign Language, Bolivian Sign Language, Brazilian Sign Language, British Sign Language, Catalan Sign Language, Colombian Sign Language, Czech Sign Language, Danish Sign Language, Dutch Sign Language, Ethiopian Sign Language, Finnish Sign Language, Flemish Sign Language, French-Belgian Sign Language, French Sign Language, German Sign Language, Greek Sign Language, Irish Sign Language, Italian Sign Language, Japanese Sign Language, Malawi Sign Language, Malaysian Sign Language, Maltese Sign

Language, Mexican Sign Language, Nepalese Sign Language, New Zealand Sign Language, Nicaraguan Sign Language, Norwegian Sign Language, Peruvian Sign Language, Philippines Sign Language, Polish Sign Language, Portugese Sign Language, Quebec Sign Language, South African Sign Language, Spanish Sign Language, Swedish Sign Language, Swiss Sign Language, Taiwanese Sign Language, and Tunisian Sign Language.

Sign language is vastly different than spoken language. Instead of the sequential sounds of the voice, there is a 3 dimensional space with simultaneous action. The SignWriting Script creates 2-dimensional writing that is visually icon and full of featural information. This is true on the symbol level and on the sign level. A symbol represents phonemic information and is full of featural information to better understand the phonemes of the symbols. A sign is a 2-dimensional arrangement of symbols and is full of featural information to better understand the morphemes of the signs.

The SignWriting Script is an international standard for writing sign languages by hand or with computers. From education to research, from entertainment to religion, SignWriting has proven useful because people are using it to write signed languages.

Initially developed in 1974, the script was written exclusively by hand for 12 years. Since then the script has spread around the world and continues to be written on paper and chalkboard.

In 1981, SignWriting Publishing rapidly evolved with Block Printing. In 1986, computerization of the SignWriting Block Printing began. The current symbol encoding of the ISWA 2010 has been stable since the font release on October 20th, 2010. The larger character encoding model has been stable since the initial release of Modern SignWriting on January 12th, 2012.

The 2 families of the SignWriting Script are Handwriting for the writer and Block Printing for the reader. Block Printing uses more features and Handwriting often uses less. Block printing is used in education, publishing, and is the basis of the computerized model.

### 1.1. 2-Dimensional Signs

A sign is a variably-size logographic word. It is a 2-dimensional combination of symbols inside of a signbox with a tight bounding box and an explicit center. The size of the signbox varies with the symbols written inside.

Inside of a 2-dimensional signbox, the symbols are placed in a freeform, 2-dimensional arrangement. This feature of the script expresses spatial relation directly.

Writing based on vision uses two viewpoints: receptive and expressive. The receptive viewpoint is based on the idea of receiving an image. For the receptive viewpoint, the right hand of a signer will be written on the left side of the signbox. When SignWriting is used for transcription, the receptive view is most often used. The related writing systems of DanceWriting and MovementWriting normally use the receptive viewpoint.

The expressive viewpoint is based on the idea of expressing a concept. For the expressive viewpoint, the right hand of a signer will be written on the right side of the signbox. When SignWriting is used for authorship, the expressive view is most often used.

The are two main writing planes: the front wall (Frontal Plane) and the floor (Transverse Plane). The choice of writing plane can affect the shape of the symbols, such as the fill pattern for the hand shape palms or the tail for the movement arrows.

There are two perspectives: front and top. The front perspective is a straight on view of/from the signer. The top perspective is a top-down view of the signer. Usually, a sign will be written from a single perspective.

## 1.2. Punctuation and Text

Logographic signs are mixed with punctuation to form text.

Punctuation is a single symbol and separates a series of signs into structured sentences. A punctuation symbol is always used alone and should not be used in a sign. Line breaks should not occur before punctuation.

When written vertically, SignWriting can use 3 different lanes: left, middle, and right. The middle lane is the default lane and punctuation is always used in the middle lane. No matter the lane, the center of a sign is aligned with the center of the lane.

For body weight shifts to one side or the other, the center of the sign is aligned with a fixed horizontal offset from the middle lane into either the left or right lane.

The left and right lanes are used to represent body weight shifts and are represented by a horizontal offset from the middle lane. Body weight shifts are important to the grammar of sign languages, used

for two different grammatical aspects: 1) role shifting during sign language storytelling, and 2) spatial comparisons of two items under discussion. One "role" or "item" is placed on the right side of the body (right lane), and the other on the left side of the body (left lane), and the weight shifts back and forth between the two, with the narrator in the middle (middle lane).

### 1.3. Block Printing

Valerie Sutton writes, "SignWriting Printing is easy to read. It is designed for the reader. The Printing can be written by hand as well as by computer. If I am writing a letter to a friend in ASL, I write the letter in SignWriting Printing, taking the time to make sure that my handwritten-symbols are easy and clear to read. I try to write as clearly as if I were using a computer. Of course it is slower, but it is worth it, knowing that my friend will be able to read my letter!"

With Block Printing, a sign is a cluster of several symbols arranged in 2-dimensions space. Each symbol has a definite appearance and understanding within an established symbol set. The exact form of each symbol is structured, standardized, and highly featural.

Each symbol has two aspects. The first is the line that defines the positive shape of the symbol. The second aspect is the fill (or negative space) of the symbol that is sometimes used inside the lines for palm facing, and inside some arrow heads and tails. Not every symbol has fill. Fill matters when symbols overlap. The negative space of the symbol on top will cover part of the symbol underneath.

The Block Printing family is aimed at the needs of the reader and the publisher. The Block Printing family is ready to standardize with a fully developed model.

### 1.4. Handwriting

Valerie Sutton writes, "SignWriting Handwriting is easier to write by hand, than the Printing. It is designed for the writer. There are several variations of Handwriting, and since most of the time, the writer is only writing for private notes, some writers create their own shortcuts that work just for them...and that is fine!"

The purpose is not to recreate the iconic symbols of the International SignWriting Alphabet exactly by hand, but the purpose is to enable the writer to quickly write notes on paper or chalkboard. Handwriting often drops features of the SignWriting Script for efficiency and speed. If too many features are dropped,

the writing may loose it's clarity over time as the writer is distanced from the writing. This is common for Shorthand.

A popular form of SignWriting is cursive. It can be shared among a groups of writers or it can be individualized and personal. Cursive writing is designed to have fluid marks and a natural flow. Cursive writing may use fewer features than the iconic symbols, but should be related to an iconic symbol in appearance and meaning. Once developed, this style of writing is great for taking notes in a class.

Shorthand is a skill of the proficient writer [1]. In 1982, Sign Language Stenographers could record sign language with SignWriting Shorthand at normal signing speed [2]. Time tests proved practice and special training were required. The marks they write are personal style of quick and efficient strokes with a highly developed reception to what signifies meaning. They understand the iconic symbols of the SignWriting Script, but their marks are personal reminders rather than a fully developed text.

The shorthand in and of itself is often an incomplete representation of the gestures that were experienced. The shorthand writing can be thought of as a short-term memory device. Often shorthand notes must be revised and extended at a later time, the sooner the better.

## 2. SignWriting Text

SignWriting Text uses plain text that is diagrammatic. It defines relationships with simple structures. It clarifies likenesses that are topologically similar.

SignWriting Text is grammatically correct because it supports 2-dimensional arrangement and writing with lanes. Mathematically sized logograms are named with plain text strings based on patterns. Simple HTML and CSS are used for proper vertical layout.

This model separates visual display from layout issues. It is compatible with TrueType Fonts and server generated images, either SVG or PNG.

### 2.1. 2-Dimensional Space

Each logographic sign exists on its own 2-dimensional signbox. Each point on the signbox is identified with an X and a Y coordinate. Each signbox has a defined center. Formal numbers range from -250 to 249. Informal number have no limit.

```
              Y Axis
                | -
                |
                |
                |
                |
                |
   X Axis       |
     -----------+------------
     -          |           +
                |
                |
                |
                |
                |
                | +
```

Symbols are placed on the signbox with coordinates that represent the top-left of the symbol image. Symbol images may overlap.

2-dimensional space does not have a normative 1-dimensional order. When symbols overlap, the relative order of the overlapping symbols is important. Otherwise, the exact string order of the spatial symbols is unpredictable.

Each signbox is an unordered list of symbols in 2-dimensional relationships that can be represented with an ASCII string as the name. This name by itself can not be sorted with a binary string compare. For sorting, the signbox text must be prefixed with a sequential list to become a sortable term.

## 2.2. Terms for Sorting

A term is a specialized sign that uses a sequential prefix before the 2-dimensional signbox text. The sequence is a list of writing symbols and/or detailed location symbols that identify temporal order and additional analysis. A valid sequence must contain at least one symbol and can not contain punctuation.

This optional prefix is written by the author or extracted from a dictionary. The sorting of terms in universally supported through the binary string comparison.

There are several theories on the best way to structure a sequence. The most productive is based on the SignSpelling Sequence theory of Valerie Sutton. A sequence is structured as a series of starting handshapes followed by optional movements, transitional handshapes, movement, and end handshapes. Only symbols from category 1 (hands)

and category 2 (movement) should be used in this first section. The last section of the sequence should contain symbols of dynamics & timing, head & face, or body: categories 3, 4, and 5.

Detailed location symbols from category 6 can be used in a sequence, but are rarely (if ever) needed for a sequence in general writing.

### 2.3. Mathematical Name

The mathematical name of a logographic sign is a plain text string of characters. This encoding model makes explicit those features which can be effectively and efficiently processed. Formal languages and regular expressions are used to solve fundamental problems.

Regular Expression Basics

| Characters | Description | Example |
|---|---|---|
| * | Match a literal 0 or more times | ABC* matches AB, ABC, ABCC, ... |
| + | Match a literal 1 or more times | ABC+ matches ABC, ABCC, ABCCC, ... |
| ? | Match a literal 0 or 1 times | ABC? matches AB or ABC |
| {#} | Match a literal "#" times | AB{2} matches ABB |
| [ ] | Match any single literal from a list | [ABC] matches A, B, or C |
| [ - ] | Match any single literal in a range | [A-C] matches A, B, or C |
| ( ) | Creates a group for matching | A(BC)+ matches ABC, ABCBC, ABCBCBC, ... |
| ( \| ) | Matches one of several alternatives | (AB\|BC\|CD) will match AB, BC, or CD |

Table 1

The mathematical name is structured with 11 different tokens. They can be grouped in 4 layers: the 5 structural makers (A, B, L, M, R),

the 3 base symbol ranges (w, s, P), the 2 modifier indexes (i, o), and the numbers (n).

The Tokens of SignWriting Text

| Token | Description |
|---|---|
| A | Sequence Marker |
| B | SignBox Marker |
| L | Left Lane Marker |
| M | Middle Lane Marker |
| R | Right Lane Marker |
| w | Writing BaseSymbols |
| s | Detailed Location BaseSymbols |
| P | Punctuation BaseSymbols |
| i | Fill Modifiers |
| o | Rotation Modifiers |
| n | Numbers: -250 thru 249 |

Table 2

These tokens are used in patterns to form written sign language. The following token patterns fully describe the SignWriting Text language. [3]

Token Patterns

| Regular Expression | Description |
|---|---|
| wio | a writing symbol as 3 tokens of writing base, fill modifier and rotation modifier |
| nn | coordinate with X and Y |

| | |
|---|---|
| | values as 2 numbers |
| wionn | a spatial symbol as 5 tokens, with 3 tokens for a writing symbol and 2 tokens for coordinates of top left placement |
| (wionn)* | zero or more spatial symbols |
| Bnn(wionn)* | a signbox with a preprocessed maximum coordinate and a list of spatial symbols used for horizontal writing |
| [LMR] | a lane marker: either left, middle or right. |
| [LMR]nn(wionn)* | a signbox in either the left, middle, or right lane with a preprocessed maximum coordinate and a list of spatial symbols used for vertical writing |
| [ws] | a writing base symbol or a detailed location base symbol |
| [ws]io | a writing symbol or a detailed location symbol |
| ([ws]io)+ | one or more writing symbols and/or detailed location symbols |
| (A([ws]io)+)? | an optional prefix as a prefix marker followed by one or more writing symbols and/or detailed location symbols |
| Pionn | a punctuation symbol as |

| | |
|---|---|
| | a punctuation base symbol with a preprocessed minimum coordinate |
| (((A([ws]io)+)?Bnn(wionn)*)\|Pionn)+ | a sign text for horizontal writing as a string of signboxes (with optional prefixes) and punctuation |
| (((A([ws]io)+)?[LMR]nn(wionn)*)\|Pionn)+ | a sign text for vertical writing as a string of signboxes in lanes (with optional prefixes) and punctuation |

Table 3

## 2.4. Visual Image

The visual image of a logographic sign is a 2-dimension arrangement of symbols inside of a signbox. Each signbox has a defined width, height, and 2-dimensional center that can be calculated from the plain text.

The TrueType Font is ready for experimental use. The entire ISWA 2010 is included with 2-dimensional arrangements of symbols for the logograms. The TrueType Font utilizes the Private Use Area Unicode characters. There are 4 open issues: the symbols are fuzzy, handshapes overlap incorrectly, arrow head/tail fill is missing, and Graphite occassionally crashes.

The SignWriting Icon Server (open source on GitHub) is able to create logographic sign images from the mathematical names. The SVG is print quality. The PNG images are pixelated. The SignWriting Icon Server includes multiple levels of caching to improve the speed and response of the user experience over time.

## 3. Formal SignWriting

According to Wikipedia, "In mathematics, computer science, and linguistics, a formal language is a set of strings of symbols that may be constrained by rules that are specific to it." [4]

Formal SignWriting defines a formal language for the signed languages of the world. Any sign of any sign language can be written as a string of ASCII characters.

Formal SignWriting is a heuristic model. The first prototypes were created in 2008. Through trial and error, the model was successively refactored to reduce the complexity and the computation cost of the implementations. The model has been optimized for common usage and processing. The final model has been stable since January 12th, 2012.

## 3.1. Lite Markup

ASCII characters are used to identify structure, symbols, and coordinates. It has proven to be beneficial to use a human readable lite markup of ASCII words separated by white space. Each word represents either a sign or a punctuation. The lite markup has the advantage of a small size without requiring special Unicode or XML functions. Simple regular expressions can quickly and efficiently process the lite markup.

Formal SignWriting uses the following structures with the associated regular expressions.

```
'Symbol key'  S[123][0-9a-f]{2}[0-5][0-9a-f]

'Coordinate'  [0-9]{3}x[0-9]{3}

'Explicit Coordinate'  (2[5-9][0-9]|[3-6][0-9]{2}|7[0-4][0-9])x(2[5-9
   ][0-9]|[3-6][0-9]{2}|7[0-4][0-9])

'Signbox'  [BLMR]([0-9]{3}x[0-9]{3})(S[123][0-9a-f]{2}[0-5][0-9a-f][0
   -9]{3}x[0-9]{3})*

'Term'  (A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)[BLMR]([0-9]{3}x[0-9]{3})
   (S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})*

'Punctuation'  S38[7-9ab][0-5][0-9a-f][0-9]{3}x[0-9]{3}

'Text'  ((A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)?[BLMR]([0-9]{3}x[0-9]{3
   })(S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})*|S38[7-9ab][0-
   5][0-9a-f][0-9]{3}x[0-9]{3})( (A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)
   ?[BLMR]([0-9]{3}x[0-9]{3})(S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x
   [0-9]{3})*| S38[7-9ab][0-5][0-9a-f][0-9]{3}x[0-9]{3})*
```

Any symbol key is 6 characters long. The first character of "S" identifies the start of a symbol key. The next 3 characters identify

the symbol base. The last two characters identify the fill and rotation modifiers.

There are 2 definitions for a coordinate. The more general definition simply defines 3 numbers followed by an "x" followed by 3 more numbers. The more explicit definition correctly restricts the number range from 250 to 749. The general coordinate definition is adequate for processing.

Each signbox includes 2 preprocessed maximum coordinate numbers (bottom-right). This pre-calculated value defines a tight bounding box around the symbols used in the sign.

Terms include a sequential symbol prefix (used for sorting) before the signbox definition.

Punctuation includes a preprocessed minimum coordinate. The maximum coordinate of a punctuation is derived by processing each coordinate number separately and combining the results. For both X and Y values, Maximum = 1000 - Minimum.

Text is defined as a list of intermixed signs and punctuation.

## 3.2. Query Language

The query language is a lite ASCII markup similar to Formal SignWriting. Any Formal SignWriting string can easily be converted into a query string. The query string is a concise representation for a much larger and detailed set of regular expressions. The regular expressions can be used to quickly and accurately search large files and databases containing Formal SignWriting.

A filter and repeat pattern of searching is used as a series of match criteria. A file, database, or text input is searched using a sequence of steps. Each step applies a single match criteria. Matching results are collated and the next search criteria is applied. The pattern of searching the previous results continues until all regular expressions have been used.

There are two main sections of a query string. The first searches the temporal prefix. The second searches the spatial signbox. Both sections use the same definition for a symbol or a range. The symbol search can match an exact symbol, or a set of related symbols. For the fill and rotation modifiers, the "u" character is a wildcard. The "u" stands for unknown and will match all values rather than a specific character. The range search can match a range of base symbols. The base symbol range consists of 2 values: the starting

base symbol and the ending base symbol. Every symbol between these 2 base symbols will be matched.

```
'Symbol Search'  S[123][0-9a-f]{2}[0-5u][0-9a-fu]
```

```
'Range Search'  R[123][0-9a-f]{2}t[123][0-9a-f]{2}
```

The full query string definition allows for the possibility of searching the temporal prefix and the spatial signbox at the same time.

```
'Query String'  Q((A(S[123][0-9a-f]{2}[0-5u][0-9a-fu]|R[123][0-9a-f]{
   2}t[123][0-9a-f]{2})+)?T)?(S[123][0-9a-f]{2}[0-5u][0-9a-fu]([0-9]{
   3}x[0-9]{3})?|R[123][0-9a-f]{2}t[123][0-9a-f]{2}([0-9]{3}x[0-9]{3}
   )?)*(V[0-9]+)?
```

### 3.2.1. Searching the Temporal Prefix

The temporal prefix is a sequential list of symbol keys. The query "QT" will find all signs that include a temporal prefix.

It is possible to specify the start of the temporal prefix by identifying a series of symbols and/or ranges. The query will start with an "QA" and end with a "T", such as "QA...T". Between the "QA" and "T", a series of symbol searches and/or range searches will specify the desired start of the temporal prefix. The order of the symbols and ranges is important.

```
'Temporal Prefix Search Query'  Q((A(S[123][0-9a-f]{2}[0-5u][0-9a-fu]
   |R[123][0-9a-f]{2}t[123][0-9a-f]{2})+)?T)?
```

Temporal Prefix Query Examples

| Query | Description |
|---|---|
| QT | All signs that include the temporal prefix |
| QAS100uuT | Signs with a temporal prefix that starts with the index handshape |
| QAS100uuR100t204S20500T | Signs with a temporal prefix that starts with the index handshape, followed by any handshape, followed by the single contact |

Table 4

### 3.2.2. Searching the Spatial Signbox

The spatial signbox is a list of symbols with 2-dimensional placement. The query "Q" will find all signs regardless of the symbols used or their placement.

It is possible to specify one or more symbols (or ranges of symbols) that must be included in the signbox to indicate a match. The order of the symbols is not important. Each symbol (or range) can include an optional coordinate. The coordinate is a restriction on the match, such that a symbol must be used within a certain variance of the coordinate to qualify as a match.

The variance is a number value, 0 or greater with a default value of 20. A variance of 0 will only find symbols used at an exact coordinate. A variance of 5 will match the symbols used at a coordinate, plus or minus 5 for both X and Y numbers.

'Symbol Search with Optional Coordinate' S[123][0-9a-f]{2}[0-5u][0-9a-fu]([0-9]{3}x[0-9]{3})?

'Range Search with Optional Coordinate' R[123][0-9a-f]{2}t[123][0-9a-f]{2}([0-9]{3}x[0-9]{3})?

'Variance' (V[0-9]+)?

'Spatial Signbox Search Query' Q(S[123][0-9a-f]{2}[0-5u][0-9a-fu]([0-9]{3}x[0-9]{3})?|R[123][0-9a-f]{2}t[123][0-9a-f]{2}([0-9]{3}x[0-9]{3})?)*(V[0-9]+)?

Spatial Signbox Query Examples

| Query | Description |
|---|---|
| Q | All signs |
| QS100uu | Signs with the index handshape in the spatial order |
| QS100uu480x480 | Signs with the index handshape in the spatial order used near coordinate (480,480) |
| QS100uu480x480V0 | Signs with the index handshape in the spatial order used at the exact coordinate (480,480) |
| QS100uuR2fft36c | Signs with the index handshape and a symbol from the head & face range |

Table 5

### 3.2.3. Transformation to Regular Expression

The conversion from Query String to Regular Expression has been fully implemented in the SignWriting Icon Server.

The Query Language to regular expression generator uses the following regular expression structures as building blocks.

'Term Prefix' (A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)

'SignBox Prefix' [BLMR]([0-9]{3}x[0-9]{3})

'Spatial Symbols' (S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})*

The Term Prefix is a structural marker followed by one or more symbols. For the query string "QT", the prefix is required. For the general "Q", the prefix is optional so "?" is appended to the Term Prefix regular expression.

The SignBox Prefix is a combination of structural marker and preprocessed maximum coordinate. Every constructed regular expression will include the SignBox Prefix.

The Spatial Symbols is zero or more symbol definitions and associated coordinates. The Spatial Symbols regular expression is used for every search. For both "Q" and "QT", it is the only symbol matching

used. When searching for specific symbols and ranges, the general Spatial Symbols definition will sandwich the specific search definitions.

Searching for number ranges with regular expressions requires a unique technique. This technique was described to the LinkedIn Regular Expression Experts at the end of 2011 [5]. Searching for number ranges in hexadecimal with regular expressions is slightly more complicated but uses the same solution.

## 4. Unicode Integration

SignWriting Text is integrated with Unicode in the Private Use Area.

### 4.1. Private Use Area Font Characters

The Unicode PUA is a simple shift of the x-Binary-SignWriting coded character set. Each code is increased by decimal value 1,038,080 which is FD700 in hex. An experimental TrueType Font converts the Unicode PUA to create the visual images.

### 4.2. Proposal

A shift of the 12 bit characters of x-Binary-SignWriting by 1D700 will use the range U+1D800 to U+1DFFF, using eight 8-bit rows of Unicode Plane 1 known as the the SMP: Supplementary Multilingual Plane. These rows occur inside an unassigned section of the Notational systems.

These are the characters being used by the community. The gap between the ISWA 2010 symbols and the number sections illustrates two truths. First, the entire Sutton MovementWriting family will be encoded. Second, it doesn't really matter where the numbers are placed, perhaps plane 14.

The number characters encode the ruler principle with characters. The ruler principle is built in automatically for scripts written sequentially in one dimension. The number characters are needed for 2-dimensional logograms, where the spatial relationship between symbols is explicitly stated with X,Y Cartesian coordinates. Number characters may be a useful concept for other scripts and notations to support 2-dimensional script processing.

The entire set of characters is used for a plain text model of a 2-dimension logographic script with freeform placement of symbols.

Future additions to the ISWA 2010 will include essential hand shapes and new mouth shapes. New characters will extend the SignWriting Text model with minimal complications.

Future proposals will include the rest of the Sutton MovementWriting System.

## 5. IANA Considerations

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to the code spaces of the Center for Sutton Movement Writing, in accordance with [RFC2978]. protocol, in accordance with BCP 26, [RFC2434].

See IANA: http://www.rfc-editor.org/rfc/rfc2978.txt

Conforms with RFC 2040.

There are three name spaces for the Center for Sutton Movement Writing that require definition and extension: x-ISWA-2010, x-Binary-SignWriting, and x-Character-SignWriting

SignWriting Text is an international standard with several coded character sets. These sets may require additional hand and mouth shapes.

The following terms are used here with the meanings defined in BCP 26: "name space", "assigned value", "registration".

The following policies are used here with the meanings defined in BCP 26: "Private Use", "First Come First Served", "Expert Review", "Specification Required", "IETF Consensus", "Standards Action".

## 6. Security Considerations

None.

## 7. References

### 7.1. URIs

[1] http://www.signwriting.org/lessons/cursive/shorthand

[2] http://www.signwriting.org/lessons/cursive/byhand5.html

[3] http://signpuddle.net/wiki/index.php/
MSW:Mathematical_Model#4.B._Proto_Encoding_of_SignWriting_Text

[4] https://en.wikipedia.org/wiki/Formal_language

[5] http://www.linkedin.com/groups/Searching-3-digit-number-simple-1066587.S.85595980?qid=9cb1768b-5413-4f7f-92b5-fbef2c243df8

[6] http://signpuddle.net/wiki/index.php/MSW

[7] http://signpuddle.net/wiki/index.php/The_Wall

[8] http://signpuddle.net/iswa

[9] http://signpuddle.org

[10] http://signbank.org/signpuddle2.0/data/spml

[11] http://swis.wmflabs.org

[12] http://signbank.org/swis

[13] http://homepage.uconn.edu/~hdv02001/Articles-pdfs/131%20-%20Notation%20Systems.pdf

[14] http://www.purdue.edu/tislr10/pdfs/van%20der%20Hulst%20Channon.pdf

[15] http://www.signwriting.org/archive/docs7/sw0623_TISLR_2010_SignWriting_SignTyp_Poster.pdf

[16] http://signwriterstudio.com

[17] http://www.delegs.com/DelegsPage

[18] http://www.researchgate.net/publication/230720646_SWift_a_SignWriting_improved_fast_transcriber

## Appendix A. Modern SignWriting

This Internet Draft is in complete agreement with the theory and example workbook released on January 12th, 2012 called Modern SignWriting [6]. Modern SignWriting has example text and concretely defines the processes available. It fully documented the text encoding with regular expressions.

The Formal Signwriting encoding is ready to deploy with a maturing infrastructure. The name of a sign with 4 symbols is 60 characters long. The plain text model fully supports the grammar of written ASL with an additional 350 characters of basic HTML and CSS. The stand

alone JavaScript engine for client side viewing is 1.3 K characters and qualifies as a micro script. This script can be applied to any modern browser through a site script or initiated within a browser using a bookmark.

To search for a sign with 4 spatial symbols requires 53 characters of query string and will create around 800 characters of regular expression for searching.

There is sometimes a limit on text size. Assuming a maximum size of 256 characters, here is the list for the number of symbols that can be used with an explicit number of words.

For one sign with proper sorting, 13 symbols can be recorded with 256 characters. For two signs with proper sorting, 12 symbols can be recorded with 247 characters. For three signs with proper sorting, 11 symbols can be recorded with 238 characters. For four signs with proper sorting, 11 symbols can be recorded with 248 characters

For one sign without sorting, 19 symbols can be recorded with 255 characters. For two signs without sorting, 18 symbols can be recorded with 251 characters. For three signs without sorting, 17 symbols can be recorded with 247 characters. For four signs without sorting, 17 symbols can be recorded with 256 characters

## Appendix B. ISWA 2010

The ISWA 2010 is the abstract symbolset for the x-ISWA-2010 coded character set. The symbols are visually iconic, uniquely identified, and organized in a layered hierarchy (Appendix B.3).

The x-ISWA-2010 is a 16-bit coded character used in the font software to access the symbol glyphs.

The x-Binary-SignWriting is a 12-bit coded character set that does not directly encode the symbols of the ISWA 2010, but divides each symbol into a combination of 3 characters. The first character represents the base of the symbol. The next represents the fill of the symbol. The last character represents the rotation of the symbol.

### B.1. Grapheme

The grapheme is the fundamental unit of writing for the SignWriting script. Many graphemes of SignWriting are visually iconic. The main writing graphemes of SignWriting represent a visual conception: either hands, movement, dynamics, timing, head, face, trunk, or limb. The body concept is a combination of trunk and limb. The specific

size and shape of each grapheme is designed to balance and complement other graphemes.

The writing graphemes are extensive and specifically organized for written sign language and sign gestures.  The writing graphemes do not include the specific graphemes of DanceWriting or the general graphemes of MovementWriting.

The writing graphemes are used in clusters.  A cluster is a spatial grouping of graphemes written as a single unit.  The graphemes can overlap and obscure graphemes underneath.  A cluster can represents a sign of a sign language or a visual performance of a sign gesture.

Detailed location graphemes are separate from the main writing graphemes.  Detailed location graphemes are used individually or sequentially.  They represent isolated analysis that is written outside the cluster.

Punctuation graphemes are used when writing sentences.  They are used individually, between clusters.

When written by hand, lines are drawn to form each grapheme. Different styles draw different types of lines: either for personal taste, speed, or quality.  The main types of handwriting are formal, cursive, and shorthand.  Formal handwriting, equivalent to block printing, includes defined lines for all grapheme features, specific palm facings for hand shapes, and detailed arrow heads and tails. Cursive handwriting is more fluid and less detailed.  Handwriting for personal use can omit palm facings, generalize arrows, and other liberties of personal consumption.  Shorthand is a further reduction of detail, written for speed.  Shorthand is a memory aid to a written record and should be rewritten soon after the notes were taken.

Understanding the ratios of size and shape for the graphemes improves hand writing.  SignWriting was an exclusively handwritten script for 7 years before publishing formalized the Block Printing model.

## B.2.  Symbol

There are 37,811 symbols, each with a unique ID.  A symbol ID is a sequence of six formatted numbers of increasing detail.  The first dashed number defines the category (11).  The first two dashed numbers define the group (11-22).  The first four dashed numbers define a base (11-22-333-44).  The fifth number represents the fill (55).  The sixth number represents the rotation (66).  A symbol ID is a combination of base ID with a valid fill and a valid rotation.  A symbol ID has the format "nn-nn-nnn-nn-nn-nn", where each "n" is a digit from 0 to 9.

The fill modifier can best be understood through the palm facing of the hand graphemes.  The palm facing is based on planes.  The SignWriting script uses two planes: the Front Wall (Frontal Plane) and the Floor (Transverse Plane).  There are 6 palm facings.  The first three palm facings are parallel with the Front Wall.  The second three palm facings are parallel with the Floor.  The reader can view the signer from different viewpoints (expressive or receptive) and can view the hands from different perspectives (front or top), but no matter what the viewpoint or perspective, the first three Fills represent the palm facing parallel to the Front Wall and the second three Fills represent the palm facing parallel to the Floor.

| Fill | Indicator | Meaning |
|---|---|---|
| 01 | grapheme with white palm | reader sees palm of hand parallel Front Wall |
| 02 | grapheme with half black palm | reader sees side of hand parallel Front Wall |
| 03 | grapheme with black palm | reader sees back of hand parallel Front Wall |
| 04 | grapheme with white palm and broken line | reader sees palm of hand parallel Floor |
| 05 | grapheme with half black palm and broken line | reader sees side of hand parallel Floor |
| 06 | grapheme with black palm and broken line | reader sees palm of hand parallel Floor |

Table 6

The fill modifier is redefined for the movement arrows of category 2.

| Fill | Indicator | Meaning |
|---|---|---|
| 01 | a grapheme with a black arrow head | movement of the right hand |
| 02 | a grapheme with a white arrow head | movement of the left hand |
| 03 | a grapheme with a thin, unconnected arrow head | spatial overlapping  of movement arrows for the left and right hands when they move as a unit |
| 04 | Irregular arrow stems | building blocks for complex movement |

Table 7

The rest of the other bases use a fill modifier for grouping and visual organization that is meaningful only for a particular base symbol or small set.

The rotation modifier can best be understood through the hand symbols.  The first 8 rotations progress 45 degrees counter clockwise.  The last 8 rotations are a mirror of the first 8 and progress 45 degrees clockwise.  Zero (0) degrees is understood to point to the top of the grapheme.

| Rotation | Direction | Degrees from top |
|---|---|---|
| 01 | Counter Clockwise | 0 |
| 02 | Counter Clockwise | 45 |
| 03 | Counter Clockwise | 90 |
| 04 | Counter Clockwise | 135 |
| 05 | Counter Clockwise | 180 |
| 06 | Counter Clockwise | 225 |
| 07 | Counter Clockwise | 270 |
| 08 | Counter Clockwise | 315 |
| 09 | Clockwise | 0 |
| 10 | Clockwise | 45 |
| 11 | Clockwise | 90 |
| 12 | Clockwise | 135 |
| 13 | Clockwise | 180 |
| 14 | Clockwise | 225 |
| 15 | Clockwise | 270 |
| 16 | Clockwise | 315 |

Table 8

### B.3. Hierarchy

The symbols of the ISWA 2010 are placed in a layered hierarchy for organization and access. There are 4 levels to the ISWA 2010 hierarchy: category, group, base, and symbol.

There are 7 categories. The first number of the symbol ID identifies the category. The first 5 categories contain writing symbols for use in clusters: 1) Hands, 2) Movement, 3) Dynamics & Timing, 4) Head &

Face, and 5) Body. The Body category can be broken into 2 subcategories: 5.1) Trunk and 5.2) Limb.

The 6th category is Detailed Location that contains symbols used alone or in sequence, always outside the cluster. The 7th category is Punctuation that contains symbols used between clusters for text.

The 7 Categories of the ISWA 2010

| Cat | Purpose | Name | Description |
|---|---|---|---|
| 1 | Writing | Hands | Handshapes from over 40 Sign Languages are placed in 10 groups based on the numbers 1-10 in American Sign Language. |
| 2 | Writing | Movement | Contact symbols, small finger movements, straight arrows, curved arrows and circles are placed into 10 groups based on planes: The Front Wall Plane includes movement that is "parallel to the front wall" and the Floor Plane includes movement that is "parallel to the floor". |
| 3 | Writing | Dynamics & Timing | Dynamics Symbols are used to give the "feeling" or "tempo" to movement. They provide emphasis on a movement or expression, and combined with Punctuation Symbols become the equivalent to Exclamation Points. The Tension Symbol, combined with Contact Symbols, provides the feeling of "pressure", and combined with facial expressions can place emphasis or added feeling to an expression. Timing symbols are used to show alternating or simultaneous movement. |
| 4 | Writing | Head & Face | Starting with the head and then from the top of the face and moving down. |

| | | | |
|---|---|---|---|
| 5 | Writing | Body | Torso movement, shoulders, hips, and the limbs are used in Sign Languages as a part of grammar, especially when describing conversations between people, called Role Shifting, or making spatial comparisons between items on the left and items on the right. |
| 6 | Detailed Location | Detailed Location | Detailed Location symbols used are used alone or in sequence outside of the cluster.  They may be useful for sorting large dictionaries, refining animation, simplifying translation between scripts and notation systems, and for detailed analysis of location sometimes needed in linguistic research. |
| 7 | Punctuation | Punctuation | Punctuation symbols are used when writing complete sentences or documents in SignWriting. |

Table 9

There are 30 groups.  The first 2 dashed numbers in the symbol ID identify the group.  The 30 groups can be divided into 3 sets of 10. The first ten are hands, category 1.  The second ten are movements, category 2.  The third ten are categories 3 thru 7.  In order, 1 group for the Dynamics & Timing category, 1 for Head, 4 for Face, 1 for Trunk, 1 for Limb, 1 for Detailed Location, and 1 for Punctuation.

The 30 groups with symbol ID segment.

| First Set | Second Set | Third Set |
|---|---|---|
| 01-01 Index | 02-01 Contact | 03-01 Dynamics & Timing |
| 01-02 Index Middle | 02-02 Finger Movement | 04-01 Head |
| 01-03 Index Middle Thumb | 02-03 Straight Wall Plane | 04-02 Brow Eyes Eyegaze |
| 01-04 Four Fingers | 02-04 Straight Diagonal Plane | 04-03 Cheeks Ears Nose Breath |
| 01-05 Five Fingers | 02-05 Straight Floor Plane | 04-04 Mouth Lips |
| 01-06 Baby Finger | 02-06 Curves Parallel Wall Plane | 04-05 Tongue Teeth Chin Neck |
| 01-07 Ring Finger | 02-07 Curves Hit Wall Plane | 05-01 Trunk |
| 01-08 Middle Finger | 02-08 Curves Hit Floor Plane | 05-02 Limbs |
| 01-09 Index Thumb | 02-09 Curves Parallel Floor Plane | 06-01 Detailed Location |
| 01-10 Thumb | 02-10 Circles | 07-01 Punctuation |

Table 10

There are 652 bases. The first 4 dashed numbers of a symbol ID identify the base. The 652 bases are divided between the 30 groups. For each group, there are less than 60 bases. The bases are often displayed in columns of 10.

Each base can have up to 96 symbols. All 6 dashed numbers of the symbol ID are required to identify a symbol. Each symbol is a combination of a base, fill, and rotation. The fill is identified by the 5th number of the symbol ID with possible values from 01 to 06. The rotation is identified by the 6th number of the symbol ID with possible values from 01 to 16.

## B.4. Combined Character Sequence

Each symbol of the ISWA 2010 can be expressed with a combination of 3 characters. The first character represents the base of the symbol. The next character represents the fill of the symbol. The last character represents the rotation of the symbol.

There are three forms the fill and rotation can use to represent their value: a hexadecimal key, an x-Binary-SignWriting character, or an x-Character-SignWriting character.

The x-Binary-SignWriting coded character set uses a 12-bit encoding. Code points in this set use a "B+" prefix along with the 3 hexadecimal digits that represent the value.

The x-Character-SignWriting coded character set uses the Private Use Area of Unicode. These code points occur on plane 15. Code points in this set use a "U+" prefix along with the 5 hexadecimal digits that represent the value.

The fill value ranges from 1 to 6. The fill key is 1 less than the value and ranges from 0 to 5.

| Fill Value | Key | x-Binary-SignWriting | x-Character-SignWriting |
|---:|---|---|---|
| 1 | 0 | B+110 | U+FD810 |
| 2 | 1 | B+111 | U+FD812 |
| 3 | 2 | B+112 | U+FD812 |
| 4 | 3 | B+113 | U+FD813 |
| 5 | 4 | B+114 | U+FD814 |
| 6 | 5 | B+115 | U+FD815 |

Table 11

The rotation value ranges from 1 to 16. The rotation key is written in hexadecimal and is equal to 1 less than the value and ranges from "0" to "f".

| Rotation Value | Key | x-Binary-SignWriting | x-Character-SignWriting |
|---|---|---|---|
| 1 | 0 | B+120 | U+FD820 |
| 2 | 1 | B+121 | U+FD821 |
| 3 | 2 | B+122 | U+FD822 |
| 4 | 3 | B+123 | U+FD823 |
| 5 | 4 | B+124 | U+FD824 |
| 6 | 5 | B+125 | U+FD825 |
| 7 | 6 | B+126 | U+FD826 |
| 8 | 7 | B+127 | U+FD827 |
| 9 | 8 | B+128 | U+FD828 |
| 10 | 9 | B+129 | U+FD829 |
| 11 | a | B+12A | U+FD82A |
| 12 | b | B+12B | U+FD82B |
| 13 | c | B+12C | U+FD82C |
| 14 | d | B+12D | U+FD82D |
| 15 | e | B+12E | U+FD82E |
| 16 | f | B+12F | U+FD82F |

Table 12

Further, a 16 bit symbol code from the x-ISWA-2010 exists for each of the valid combined character sequences. This relationship can be stated as (symbol code = ((base code - 256) * 96) + ((fill value - 1) * 16) + rotation value). The first symbol code is 1 and the last valid symbol code is 62,504.

The first symbol has an ID of "01-01-001-01-01-01" and a symbol code of 1.

Symbol code 1 = symbol key S10000 = B+130, B+110, B+120 = U+FD830, U+FD810, U+FD820.

Symbol code 1 = ( ( hexdec('100') - 256 ) * 96 ) + ( ( fill_value(1) - 1 ) * 16 ) + rotation_value(1).

Symbol code 1 = ( ( 256 - 256 ) * 96 ) + ( ( 1 - 1 ) * 16 ) + 1.

Symbol code 1 = ( 0 * 96 ) + ( 0 * 16 ) + 1.

Symbol code 1 = 1.

### B.5. Validity

Although there are 6 possible fills and 16 possible rotations, not every combination of base, fill, and rotation is valid. Each base has a set of valid fills and a set of valid rotation. These validity sets contain one or more values from the defined range.

For each value, the inclusion in the validity set can be expressed with a value of "0" or "1". For fill values, lining up the digit from left to right, will result in a string 6 digits long. The value of the 6 digit number is 2 ^ (value -1).

| Fill Value | 1 | 2 | 3 | 4 | 5 | 6 | Binary | Power of 2 |
|---|---|---|---|---|---|---|---|---|
| 1 | X |  |  |  |  |  | 100000 | 1 |
| 2 |  | X |  |  |  |  | 010000 | 2 |
| 3 |  |  | X |  |  |  | 001000 | 4 |
| 4 |  |  |  | X |  |  | 000100 | 8 |
| 5 |  |  |  |  | X |  | 000010 | 16 |
| 6 |  |  |  |  |  | X | 000001 | 32 |

Table 13

The value of any fill validity set is equal to the sum of the power of 2 for each fill value in the set. The empty set is invalid and has a sum of zero (0). The full set of all possible fills has a sum of 63.

| Fill Set | 1 | 2 | 3 | 4 | 5 | 6 | Binary | Power of 2 |
|---|---|---|---|---|---|---|---|---|
| {} | | | | | | | 000000 | 0 |
| {1,2,3,4,5,6} | X | X | X | X | X | X | 111111 | 63 |

Table 14

Each base has a defined validity set for fills. The "Fills" column in the "Bases" section.

The rotation validity sets have a larger range than the fills. The possible rotation values range from 1 to 16. The power of 2 numbers are 16-bit.

| Value | Binary | Power of 2 |
|---|---|---|
| 1 | 2^0 | 1 |
| 2 | 2^1 | 2 |
| 3 | 2^2 | 4 |
| 4 | 2^3 | 8 |
| 5 | 2^4 | 16 |
| 6 | 2^5 | 32 |
| 7 | 2^6 | 64 |
| 8 | 2^7 | 128 |
| 9 | 2^8 | 256 |
| 10 | 2^9 | 512 |
| 11 | 2^10 | 1024 |
| 12 | 2^11 | 2048 |
| 13 | 2^12 | 4096 |
| 14 | 2^13 | 8192 |
| 15 | 2^14 | 16384 |
| 16 | 2^15 | 32768 |

Table 15

The value of a rotation validity set is the summation of the power of 2 numbers. The minimum summation is 1. The largest possible summation is 65,535 where all 16 rotations are valid.

Each base has a defined validity set for rotations. The "Rotations" column in the "Bases" section.

Interestingly enough, there are only 12 possible validity sets in the ISWA 2010.

| Sum | Binary | Set |
|---:|---|---|
| 1 | 100000 | {1} |
| 2 | 010000 | {2} |
| 3 | 110000 | {1, 2} |
| 7 | 111000 | {1, 2, 3} |
| 15 | 111100 | {1, 2, 3, 4} |
| 31 | 111110 | {1, 2, 3, 4, 5} |
| 63 | 111111 | {1, 2, 3, 4, 5, 6} |
| 187 | 11011101 | {1, 2, 4, 5, 6, 8} |
| 255 | 11111111 | {1, 2, 3, 4, 5, 6, 7, 8} |
| 511 | 1111111110000000 | {1, 2, 3, 4, 5, 6, 7, 8, 9} |
| 48059 | 1101110111011101 | {1, 2, 4, 5, 6, 8, 9, 10, 12, 13, 14, 16} |
| 65535 | 1111111111111111 | {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16} |

Table 16

## Appendix C. SignPuddle Standard

The SignPuddle Standard for SignWriting text has been stable since January 12th, 2012.

### C.1. Licenses

The font software is available under SIL's Open Font License.

The reference material is licensed under Creative Commons attribution, share alike (by-sa).

The current open source projects are licensed under the GPL 2 for MediaWiki and GPL 3 for the general software on Github. Any contributions to the open source code must agree to a possible relicense in the future under a BSD like license.

After the financial issues [7] of the Center for Sutton Movement Writing have been addressed, the open source projects will relicensed under a more open and free BSD-like license, such as the MIT License.

## C.2. Infrastructure

### C.2.1. International SignWriting Alphabet Fonts

The International SignWriting Alphabet 2010 (ISWA 2010) Font Reference [8] is a product of the collaboration between SignWriting inventor, Valerie Sutton, and SignWriting encoder Stephen E Slevinski Jr. Special thanks to Adam Frost's excellent work on the SVG refinement and more.

The ISWA 2010 fonts have been stable since their initial release on October 20th, 2010.

Valerie Sutton

- hand crafted and organized 30K plus individual glyphs
- created a 2 dimension PNG of 3 colors for each
- named each individual glyph with 6 degrees of significance
- font name: ISWA 2010 Sutton

Steve Slevinski

- counted and numbered the glyphs
- created mathematical names
- analyzed PNGs for line and fill
- refactored glyphs - font name: ISWA 2010 PNG Standard
- extended glyphs - font names: ISWA 2010 PNG Inverse, Shadow, Colorized
- traced glyphs - font names: ISWA 2010 SVG Line Trace, Shaddow Trace, Smooth, and Angular
- refactored and extended Adam's SVG work - font name: ISWA 2010 SVG Refinement

Adam Frost

- manually traced each and every glyph that could not be automatically rotated

- font name: ISWA 2010 SVG Refinement

- physically performed and photographed every hand shape

- font name: ISWA 2010 Hand Photo

- consulted with Valerie in places of ambiguity

- found the Facial Irregularity, documented in the ISWA 2010 Errata

### C.2.2. SignPuddle Online

SignPuddle Online [9] is the current home of the international community of online writers of the SignWriting Script. Online tools make it possible to create SignWriting dictionaries and documents directly on the web. Each collection is freely available as a small XML file [10]. Dozens of sign languages from around the world are represented. Each language can have several collections of SignWriting.

### C.2.3. SignWriting Icon Server

The SignWriting Icon Server create SVG and PNG images and queries data collections using an open API. The image creation is stable and fully implemented. The API is currently under construction with only an initial level of support.

The main server is available on Wikimedia Labs [11] for all SignWriting projects.

A backup server is available on SignBank [12].

Each SignWriting Icon Server provides the SignWriting Thin Viewer as a site script and as a bookmark.

Additional SignWriting Icon Servers can be created directly from the GitHub source.

### C.2.4. Wikimedia Incubator

The SignWriting Script has been enabled on Wikimedia Incubator using the SignWriting Gadget.

### C.2.5. SignWriting Thin Viewer

The SignWriting Thin Viewer uses JavaScript to wrap the sign names with basic HTML and CSS to fully supports the grammar of written ASL. This script can be applied to any modern browser through a site script or initiated within a browser using a bookmark.

## C.3. Compatibility

SignTyp, SignWriter Studio, the DELEGS Editor, SWift, and more.

### C.3.1. SignTyp

This standard is being integrated with the SignTyp linguistic coding system developed by Rachel Channon through an NSF grant.

> Notation Systems [13] by Harry van der Hulst and Rachel Channon.
>
> Why dynamic features? [14] by Harry van der Hulst and Rachel Channon.
>
> Transcription systems as input to coding systems: SignWriting & SignTyp [15] by Charles Butler and Rachel Channon.

### C.3.2. SignWriter Studio

SignWriter Studio [16] is a Windows-only compatible application by Jonathan Duncan. It has an alternate symbol selection technique. According to Valerie Sutton, it illustrates a unique insight into the hand shapes of the ISWA.

Jonathan Duncan writes:

> SignWriter Studio has 4 ways to get the basic symbol base, and 3 ways to modify the selected base.
>
> 1) Select the base symbol from a complete list of base symbols organized in a tree view 2) Search for a hand symbol in hand search section by hand feature. 3) Select a symbol already present in the signbox. 4) Select a symbol from a Favorites section.
>
> Then one of three chooser to define the fill and rotation will become available. 1)The hand chooser. 2)The arrow chooser. 3)The general chooser.
>
> The Hand chooser is to quickly find the symbol for a certain, hand, plain(wall or floor), palm facing and rotation. The Hand

Chooser also extends add a fourth palm facing to logically show all possible symbols in their most common uses. This chooser resembles the instruction manual explaining the use of hand shapes.

The Arrow Chooser is to quickly find arrows for a certain hand, plain(wall or floor) and rotation.This chooser resembles the instruction manual explaining the use of arrows.

The General Chooser is for symbols for which the two previous chooser do not work well and gives a grouped list of symbols for the base group.

### C.3.3. DELEGS Online

The DELEGS Editor [17] from the University of Hamburg and C1 WPS GmbH in Germany is designed for Deaf Education. It is a tool for writing translation texts between spoken and signed languages.

Spoken language text is used to display horizontal SignWriting Text from left to right. The spoken language can appear beneath the sign or it can be hidden.

### C.3.4. SWift

SWift is a SignWriting improved fast transcriber [18] from Claudia Savina Bianchini, Fabrizio Borgia, and Maria De Marsico. SWift is under active development. The design "guides and simplifies the editing process".

SWift uses an alternate symbol hierarchy than the ISWA 2010. A conversion library is planned in the future to support Formal SignWriting strings.

Author's Address

Stephen E Slevinski Jr
SignPuddle

Email: slevin@signpuddle.net