An AAC Usability Toolkit

Jeff Higginbotham, Ph.D.
Carrie Kirkland, M.S.ed
Jenn Seale, M.S.
SUNY Buffalo

Why evaluate device use?

Many AAC devices utilize complex user interfaces, designed to perform a wide variety of communication, programming and computer control functions. While there is no single development approach, AAC device user interface development is usually guided by hardware and software designers, educators and clinicians, then tried out with expert AAC users who provide feedback for developers. Much less frequently is the design process iterative - where the device use is studied as individuals attempt to program and/or communicate with these devices. One of the problems is the lack of appropriate low cost usability tools for AAC design.

Over recent years a variety of cognitive processing models have been developed, each taking a common processing cycle and recasting it into a new task domain: technology use, language production, language comprehension, social interaction (e.g., Clark, 1996; Brennan, 1998; Norman, 1988). These models have been used to describe performance as well as locate sources of performance error. Norman (1988) provides a basic example for the cognitive cycle functioning during our use of everyday objects in our environment: **Form a goal-> Form an intention -> Specify an action -> Execute the action – (then) - Perceive -> Interpret -> Evaluate outcome**.

The AAC-UT consists of two tools for describing and evaluating the sufficiency of AAC interfaces as they are being used for communication and device programming. The AAC-UT provides a set of evaluation criteria to help the researcher/designer describe device use problems, and to specify the point in the cognitive action cycle that the problem occurred, probable causes, ramifications of the problem and the types of support present or absent during the specific event.

Version 1 of AAC-UT utilizes a Microsoft Access database to catalog information about the potential usability of a particular software interface component or specific goal-directed use of the application (e.g., construct an utterance using the keyboard).

Implemented on an open source video annotation program (ELAN), AAC-UT provides the research/designer the ability to specify events in a audio or video file of interest, code and comment upon that event, search for similar events, and output a record of the transcribed video for further analysis or report writing.

Functional Characteristics of the AAC Usability Toolkit

- Identify and segment specific user-device events from one or multiple video sources.
- Annotate and code events according to a variety of descriptive and evaluative categories dealing with device use and user error. Codes may be added to, or substituted with other codes on the Elan system.
- Results may be exported as a tab-delimited text file consisting of the code categories, notes and time onset, offset and duration information. These data may be opened in a spreadsheet. These results may be shared with device developers providing specific information about the usability of their products.

References

1. Higginbotham, DJ, & Caves, K (2002). AAC performance and usability issues: the effect of AAC technology on the communicative process. *Assistive Technology,* 14 (1), 45-57.
2. Brennen, S. (1998). The Grounding Problem in Conversations With and Through Computers. In *Social and cognitive psychological approaches to interpersonal communication* (pp. 201-225). Hillsdale: Lawrence Erlbaum.
3. Norman, D. (1990). The Design of Everyday Things, Basic Books.
4. Clark, H. (1996). *Using Language*, Cambridge University Press.

Elan may be downloaded free of charge at: http://www.lat-mpi.eu/tools/elan/.

The AAC-UT codes and code definitions for Elan may be downloaded from:
http://aac-rerc.psu.edu/index.php/projects/show/id

Acknowledgments

AAC-RERC
SPREAD THE WORD

University at Buffalo
The State University of New York

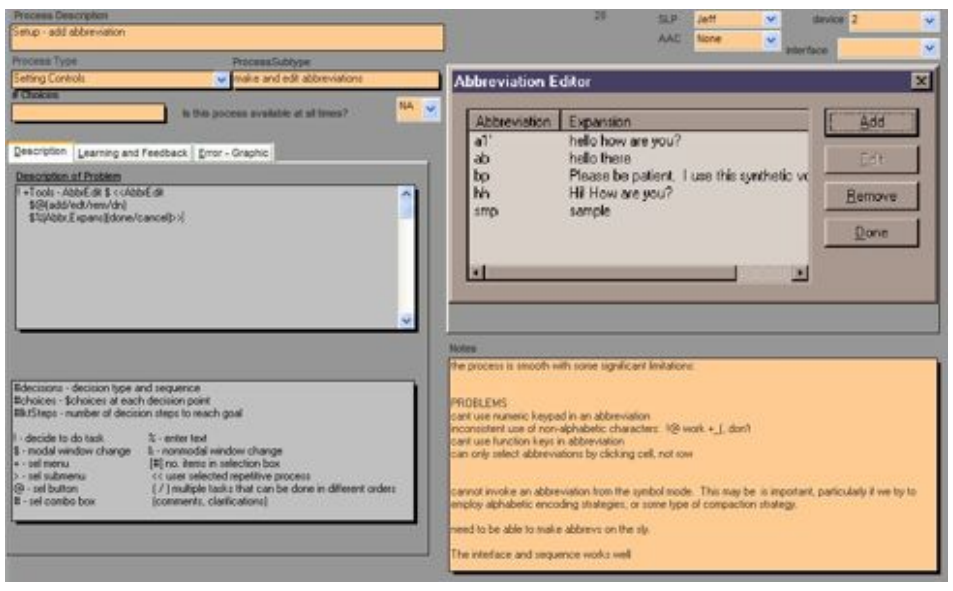
UB's Signature Center for Excellence in Augmented Communication

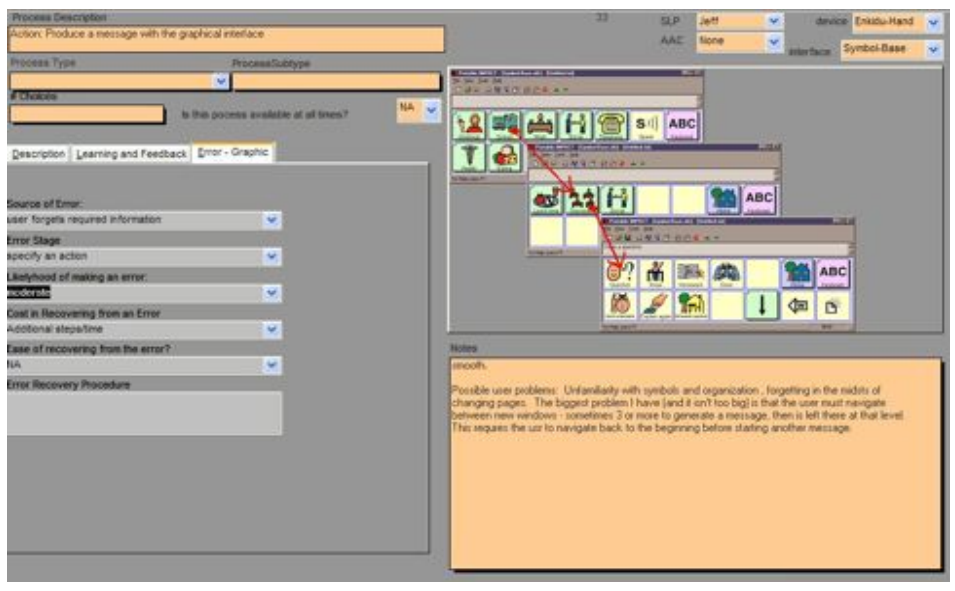
NATIONAL INSTITUTE ON DISABILITY AND REHABILITATION RESEARCH
NIDRR

The Rehabilitation Engineering Research Consortium on Communication Enhancement (AAC-RERC) is funded under grant #H133E080011 from the National Institute on Disability and Rehabilitation Research (NIDRR) in the U.S. Department of Education's Office of Special Education and Rehabilitative Services (OSERS). The contents do not necessarily represent the policy of NIDRR.

Corresponding Author: Jeff Higginbotham (cdsjeff@gmail.com

AAC-UT Version 1

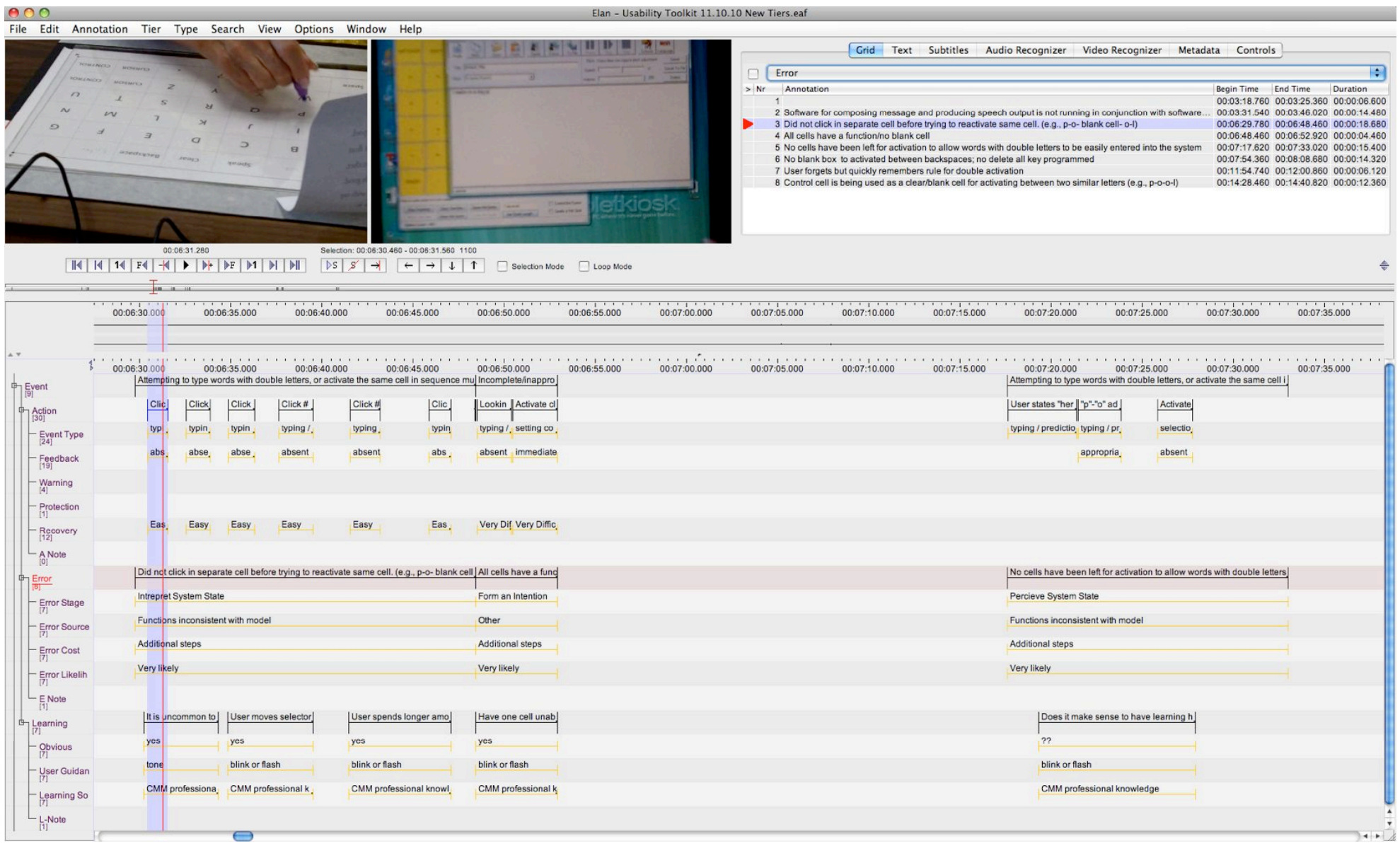




Version 1 of the usability toolkit was focused on describing and evaluating AAC software features and operations. The evaluator typically inspects the software, and analyzes its potential for producing errors based on his/her own experience. A graphic of a particular user interface feature can be stored in each record. A report can be generated for each software evaluation and sent to the developer.

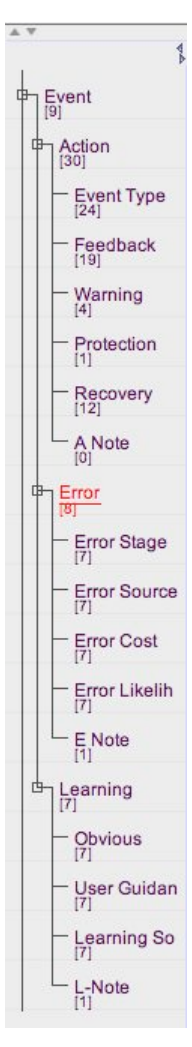
AAC-UT Version 2

Version 2 of the usability toolkit focuses utilizes ELAN – a free software tool for creating complex video and audio annotations (http://www.lat-mpi.eu/tools/elan/).



ELAN provides a means of annotating (transcribing, commenting, coding) user activity recorded from multiple video and audio sources. The length of the event, action, error and learning annotations reflect the amount of time each was observed to occur. The figure above shows the analysis of the errors made by an individual attempting to program a keyboard display on a prototype technology, using the AAC-UT protocol. Any event may contain multiple actions, errors or instances of learning. The figures below display the coding hierarchy (left), which can be collapsed or expanded. Codes appear in a drop down menu which can be selected using a mouse (middle). Once coded, the transcript can be output in a variety of ways, included tab-delimited text, amenable to analysis using a variety of software. The transcript (right) displays the event label, time codes and the content of each annotation.

These data can be used to provide quantitative and qualitative summaries of user performance, or selectively compiled to illustrate a particular user-device problem. Currently we are exploring ways to use the report capabilities of ELAN as subtitle input to illustrate user problems directly on video.



	A	B	C	D
1	Type	Onset (ss.ms)	Duration	Behavior
2	default	17.76	2.53	sssss
3	Event	198.76	6.6	event
4	Error	198.76	6.6	
5	Event	211.54	14.48	Prompt box pops up to indicate character limit has been exceeded in primary software being used.
6	Error	211.54	14.48	Software for composing message and producing speech output is not running in conjunction with software being used to input selections
7	Error Stage	211.54	14.48	Intrepret System State
8	Error Source	211.54	14.48	Lack of visible guidelines
9	Error Cost	211.54	14.48	info loss
10	Error Likelihood	211.54	14.48	likely
11	Action	213.94	1.7	Cont. Typing
12	Action	217.7	2.52	User asks to go on
13	Event	389.78	18.68	Attempting to type words with double letters, or activate the same cell in sequence multiple times (e.g., p-o-o-l)
14	Error	389.78	18.68	Did not click in separate cell before trying to reactivate same cell. (e.g., p-o- blank cell- o-l)
15	Error Stage	389.78	18.68	Intrepret System State
16	Error Source	389.78	18.68	Functions inconsistent with model
17	Error Cost	389.78	18.68	Additional steps
18	Error Likelihood	389.78	18.68	Very likely
19	Action	390.46	1.1	Click #1
20	Action	392.58	1.32	Click #2
21	Action	394.88	1.46	Click #3
22	Action	397.66	1.86	Click #4
23	Action	401.56	1.64	Click #5
24	Action	405.9	1.18	Click #6
25	Action	408.42	0.04	Instruction from observer
26	Event	408.46	4.46	Incomplete/inappropriate instruction
27	Error	408.46	4.46	All cells have a function/no blank cell
28	Error Stage	408.46	4.46	Form an Intention
29	Error Source	408.46	4.46	Other
30	Error Cost	408.46	4.46	Additional steps
31	Error Likelihood	408.46	4.46	Very likely
32	Action	408.52	1.9	Looking for unprogrammed cell
33	Action	410.5	2.42	Activate clear box
34	Event	437.62	15.4	Attempting to type words with double letters, or activate the same cell in sequence multiple times (e.g., p-o-o-l)
35	Error	437.62	15.4	No cells have been left for activation to allow words with double letters to be easily entered into the system
36	Error Stage	437.62	15.4	Percieve System State
37	Error Source	437.62	15.4	Functions inconsistent with model
38	Error Cost	437.62	15.4	Additional steps
39	Error Likelihood	437.62	15.4	Very likely
40	Action	437.66	3.76	User states "here we go again"
41	Action	441.48	2.38	p-"o" added to sentence; observer instructs to activate blank cell
42	Action	445.86	1.92	Activate "blank box"

Describing and Analyzing Device Use / Error

ACTION DESCRIPTION

These following are primarily low-inference, descriptive categories, providing contextual information about the type of activity, state of the system in which the error occurred.

Event – A description of the general activity that the device user is currently engaged.

Description of Problem (Version 1 only) – The description of user and device actions are described using a performance grammar developed for the toolkit (See Version 1 figures).

Event Type – The general activity that the user is performing (selection sequence, typing / prediction, programming, setting controls, initiating output, pointing.

Feedback – what types of feedback does the user receive in response to his/her actions (immediate, delayed, absent, appropriate, obtrusive, inconsistent).

Warning – What type of warning did the user receive indicating a significant change to the system (various graphic, auditory signals, changes in system state)

Protection Against Devastating Process – If the user action could result in a loss of data, shutdown, damage to the device, etc., what was the specific warning produced by the device alerting the user to this situation (Single Warning Cancel/No Cancel, Multiple Warnings, lockstep sequence, auditory siren, spoken message, text message, graphic warning, key click, none, other).

Ease of Crash Recovery – The degree to which a system can be restored (easy, difficult, very difficult)

LEARNING

These categories have to do with ways that the user could understand the event. What resources are available to the user, how is it presented. The following categories provide the background upon which the user is expected to use to complete a task. Filling these categories out requires a moderate level of inference / interpretation of the event by the evaluator.

Obviousness - Should the process be obvious to the user (Yes, No, ??, N/A) ?

User Guidance - what types of cues are being used to inform or guide the user when engaged in the task at hand (click, tone or beep, text deletion, color change, change in button outline or thickness, insertion of graphic element, synthesized speech output, thermometer, window appears/disappears, menu appears, disappears)?

Sources for Learning - ways in which the user can or could learn about this system:

- mental model based on common knowledge
- mental model based on professional knowledge
- mental model based on software knowledge
- spoken (explicit) prompts
- help menu
- inferred actions
- memorization required
- no information provided

ERROR

In this section, the specific error made by the user, or problem that occurred is explained in detail enough that to provide support for the developer to minimize its chances for occurring on a repeated basis.

Error Stage - Based on Don Norman's (1990) task-action cycle. Use these categories to determine what point in someone's cognitive activity can the error / problem be located:

- Forms goal - determines the desired outcome of his/her actions
- Forms intention – the user has a plan to accomplish task
- Specifies action - the user to accomplish intention
- Executes action - the user to accomplish intention
- Perceives state - the user perceives response to his/her action.
- Interprets state - the user understands meaning of response
- Evaluates outcome - the user appraises response in terms of user's goals

Error Source – The specific user-device context contributing to the specific user error.

- Lack of visible guidelines - the application provides little visible guidance or constraints that would facilitate decision making
- Too many choices - there are too many choices to make an a given time.
- Functions inconsistent with user model - the functions of the technology do not conform with the advertised model.
- User has no model upon which to base decisions - the user has not model of the system to use to inform decision making
- Operations are unfamiliar - the user is unfamiliar with the operations of the technology
- User forgets information

Error Cost - Deals with the consequences of the error made by the user (additional steps, information loss, device shutdown, program crash, damage to device, can't access device, other)

Error Likelihood - the probability in which this error would happen during the operation of this technology.