SIG Proceedings Paper in LaTeX Format*

Extended Abstract†

Ben Trovato‡
Institute for Clarity in Documentation
P.O. Box 1212
Dublin, Ohio 43017-6221
trovato@corporation.com

G.K.M. Tobin§
Institute for Clarity in Documentation
P.O. Box 1212
Dublin, Ohio 43017-6221
webmaster@marysville-ohio.com

Lars Thørväld¶
The Thørväld Group
1 Thørväld Circle
Hekla, Iceland
larst@affiliation.org

Lawrence P. Leipuner
Brookhaven Laboratories
P.O. Box 5000
lleipuner@researchlabs.org

Sean Fogarty
NASA Ames Research Center
Moffett Field, California 94035
fogartys@amesres.org

Charles Palmer
Palmer Research Laboratories
8600 Datapoint Drive
San Antonio, Texas 78229
cpalmer@prl.com

John Smith
The Thørväld Group
jsmith@affiliation.org

Julius P. Kumquat
The Kumquat Consortium
jpkumquat@consortium.net

ABSTRACT

This paper provides a sample of a LaTeX document which conforms, somewhat loosely, to the formatting guidelines for ACM SIG Proceedings.[1]

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference format:
Ben Trovato, G.K.M. Tobin, Lars Thørväld, Lawrence P. Leipuner, Sean Fogarty, Charles Palmer, John Smith, and Julius P. Kumquat. 1997. SIG Proceedings Paper in LaTeX Format. In *Proceedings of ACM Woodstock conference, El Paso, Texas USA, July 1997 (WOODSTOCK'97),* 2 pages.
DOI: 10.475/123_4

1 STORY CONTROL LOGICS

Noah Wardrip-Fruin introduced the analytical concept of operational logics [?] to describe the synergistic relationship of abstract process and representational goals. He did so in the context of understanding the relationships between surface, process and data in the work leading up to the book *Expressive Processing*, and described some of the key insights there [?]. The concept was later formalized and further defined in collaboration with Michael Mateas in [?]:

> An operational logic defines an authoring (representational) strategy, supported by abstract processes or lower-level logics, for specifying the behaviors a system must exhibit in order to be understood as representing a specified domain to a specified audience. [?]

Operational logics can be used to precisely model and analyze how an author communicates through a set of abstract processes and representations an underlying "domain" with an audience. This proposal's surface content model depends on the slipperiness of the domain of human affairs that is narrative, and how at the same time operational logics "determine the state evolution of a system," how they specify an abstract model of the underlying system, "with how they are understood at a human level," through the proposed encoding [?]. Some examples of operational logics that are proposed by Wardrip-Fruin include collision logics and dialogue tree logics, which reflect certain strategies to manage the content and the actual experience of engaging with it [?]. As long as a particular logic is maintained throughout a work, and that logic reflects a systematic application of an underlying process, it can be fruitfully analyzed as an operational logic. In this section, the operational logics targeted are principally those that control and represent the branching of the story and not necessarily logics that pertain to any underlying simulation of the story domain. In this way, they are not that much different than dialogue tree logics or the use of quest flags, and the proposed SIG extension takes advantage of that simplicity. The main difference is that the strategy of reflecting the choices through interfaces are tied to an underlying time-based performance that is critical to the pacing of the dramatic work.

*Produces the permission block, and copyright information
†The full version of the author's guide is available as `acmart.pdf` document
‡Dr. Trovato insisted his name be first.
§The secretary disavows any knowledge of this author's actions.
¶This author is the one who did all the really hard work.
[1]This is an abstract footnote

DOI: 10.475/123_4

The encoding of the surface representation depends on a model of the underlying operations that each of the proposed story control logics. These logics occupy the intersection between (graphical) interface logics and performance logics, as they provide the player with options tied to an underlying content selection architecture as well as afford the player a set of performance options that either determine or enact the choices.

The state evolution and representational strategy of cinematic choice-based adventure games is represented by modeling the following **story control logics**:

Operational Logic Name	Description
1 *Response Selection*	Language-based menus that reflect player-character options.
2 *Object-Verb Selections*	Verb-selection for object or character.
3 *Inventory*	Objects that either reflect ongoing or past plots
4 *Quick-time*	Affordances that require enactment to proceed
5 /"Payoff" Signaling/	Text that indicates (truthfully or not) underlying state

There are other operational logics at work, such as collision logics and camera control logics, but these don't affect the story structure as defined by the SIG. The four above operational logics cover every relevant non-linear control mechanism and corresponding representation. These logics must be reflected in a surface model in order to represent the possible variations. Because of the importance of these control logics in determining the surface structure, we will review each in the following sections.

1.1 Response Selection

Response selection is so named because it how a player selects the next actions or intent of the character. Like most adventure games, the character itself actually carries out the action or dialogue, and even sometimes the way a particular choice plays out may differ from what was anticipated. The underlying operation is simply one of selecting the content to be played.

1.2 Object-Verb Selection

This logic is a variant on the response selection, except that it plays the character interacting with an object instead of interacting with a character and often has physical actions instead of dialogue options. An example is either knocking or kicking down a door, or picking up an object.

1.3 Inventory Logics

In this logic, the main function is as a flag to whether certain content choices are available. Secondarily, it is a reminder of important plot threads.

1.4 Quick-time

This represents the enactment of the player of a particular situation the character is in, often through repeated button presses or through a quick decision in a heated moment. Failing these may result in replaying from a checkpoint, so most do not branch the story.

1.5 /"Payoff"/ signaling

This is one of the more complex logics that may not determine underlying state at all, but rather simply shaping player expectations of how their decision affected agents. It is because of this expectation that it needs to be represented as a distinct logic in the surface story content, and because it is consistently applied in a way that would suggest an underlying model is being affected.

The complexity of navigating the surface content defined by these logics and the recorded gameplay motivates the creation of a "story browser" tool to aid analyzing the distribution and patterns that the content takes. This tool is discussed further in section ../../.. /localhost/static/advancement/proposal/murrayAdvancement.org.