

# Structured Authoring Meets Technical Comics in TechCommix

Carlos Evia  
Virginia Tech  
Blacksburg, VA  
carlos.evia@vt.edu

Siroberto Scerbo  
Virginia Tech  
Blacksburg, VA  
scerbo@vt.edu

Michael Stewart  
Virginia Tech  
Blacksburg, VA  
michael\_stewart@vt.edu

Manuel Perez-Quiñones  
Virginia Tech  
Blacksburg, VA  
mperezqu@vt.edu

Tim Lockridge  
Saint Joseph's University  
Philadelphia, PA  
tlockrid@sju.edu

## ABSTRACT

*TechCommix* is an XML grammar and GUI that allows technical communicators to build comics based on the principles of structured authoring. *TechCommix* XML uses elements of two markup languages—ComicsML and DITA—the combination of which offers a means of tagging elements connected to a comics narrative (such as speech, action, narration) and to structured technical documentation (such as context, step, example). The resulting language allows a technical writer to differentiate between instructional and entertainment content, facilitating content analysis and reuse. Additionally, the *TechCommix* GUI provides assisted means of building web comics from DITA input. In this online environment, a technical writer can transform an XML file into an HTML deliverable with multiple presentation options—extending usability and accessibility beyond the current standard of image-based web comics. Future work will examine the efficacy of these comics in communicating procedural information.

## Categories and Subject Descriptors

H.5.2 [User Interfaces]: Training, help, and documentation; I.7.2 [Document Preparation]: Markup languages

## Keywords

comics, structured authoring, XML, DITA.

## 1. INTRODUCTION

Inspired mainly by the impact of the online comic created by Scott McCloud to introduce Google Chrome, technical communicators are looking at comic books and comic strips as a genre for conveying technical information to audiences who would not read traditional documentation [1, 4, 9, 5, 3].

Using comics for technical purposes, however, represents unique challenges that begin with achieving the proper balance of entertainment and information content: it is often too easy to create a “funny” comic book that keeps important content hidden among jokes and character development. Additionally, a graphic genre like comics requires writing practices that might not be compatible with technical communication trends of structured authoring (“a publishing workflow that defines and enforces consistent organization of information”[6]) and content reuse,

potentially missing on benefits that “include clear, consistent information for customers and business efficiencies”[7].

This poster presentation reports on work in progress to develop *TechCommix*: an Extensible Markup Language (XML) grammar and online comic computing[8] environment that allows technical communicators to incorporate principles of structured authoring to balance out elements of content and humor. *TechCommix* is based on ComicsML<sup>1</sup> (a markup language for archiving comic books) and the Darwin Information Typing Architecture (DITA, an international standard for structuring technical documentation). Benefits from this treatment include enhanced accessibility of technical comics, as users who cannot follow images are able to listen to the content; opportunities for reusing, archiving, and searching content inside comics; and improved Return on Investment (ROI) metrics, allowing authors to justify the use of a comic by comparing its procedural content to that of traditional text-based documentation.

## 2. CREATING TECHCOMMIX

### 2.1 Marking up existing comics

The first stage of developing *TechCommix* involved extending ComicsML's Document Type Definition (DTD) to incorporate DITA tags and allow structured transcription of existing technical comics. Following DITA's three types of content (procedures, background or conceptual information, and quick reference information[2]), all speech and narration elements had to be tagged with a type attribute, starting with the generic plot and humor, and then specializing with *introduction*, *example*, *question*, *answer*, *explanation*, and *demonstration* (for conceptual and reference comics), and *context*, *prereq*, *step*, *choice*, *info*, *example*, *postreq*, *hazardstatement*, and *result* (for procedural comics).

Through a series of automated transformations based on Extensible Stylesheet Language (XSL), the structured transcriptions generated a Hypertext Markup Language (HTML) deliverable with options for viewing the content as a) comic only, b) comic and transcription, or c) transcription only. All transcriptions were automatically annotated to identify and label speech and narration types (step, example, prereq, postreq, etc.). Thus, a comic became a document with measurable and reusable content elements.

---

<sup>1</sup> [http://jmac.org/projects/comics\\_ml](http://jmac.org/projects/comics_ml)

Preliminary evaluation of phase 1 asked students in an *Introduction to Professional Writing* course to adapt existing text-based instructions on topics of computer safety into comics created with the online comic generator Bitstrips<sup>2</sup>. Then, students had to use a text editor to code *TechCommix* transcripts of their comics. The following code displays a sample panel transcription of a student-produced comic:

```
<panel id="p03">
<img>img/3.jpg</img>
<panel-desc>
<action>Eddie sits. Ralph stands
up.</action>
<speech characterid="2" type="step">The
first thing you should do is run
MalWareBytes</speech>
<speech characterid="1" type="plot">But I
don't have MalWareBytes.</speech>
<speech characterid="2" type="humor">Stop
interrupting; I'm getting to that.</speech>
</panel-desc>
</panel>
```

The students, who had basic knowledge of HTML but had not been previously exposed to XML or DITA, did not report problems working directly with code. However, they unanimously supported the idea of a graphical user interface (GUI) to simplify the process of building and transcribing technical comics. The main problem reported in this evaluation stage was the lack of a system for generating task-oriented panels. Conceptual and reference panels were relatively easy to build, with characters paraphrasing problems and solutions, but procedural panels demanded a faithful representation of all steps included in the original text-based task.

## 2.2 Facilitating task-to-comic translation

The next stage in developing *TechCommix* addresses the need for a comic computing GUI and facilitates proper adaptation of steps from a text-based task to a technical comic.

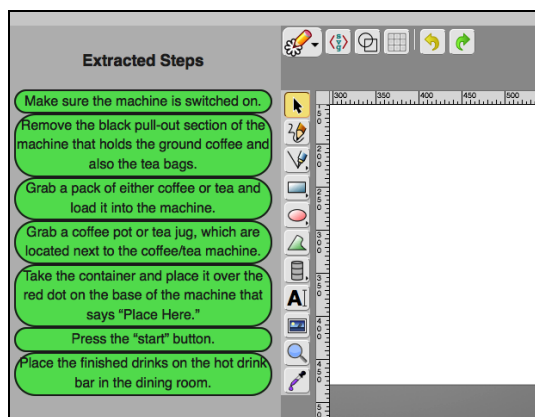


Figure 1: Steps extracted from a task

The authoring environment evolved to an online interface with four columns: 1) an area to drag and drop or open an existing DITA task for input, 2) an accordion-like list of automatically

extracted elements from the DITA input, 3) an image editor based on SVG-edit<sup>3</sup>, and 4) a *TechCommix* XML output. The user drops or opens a DITA task in column 1, then column 2 is automatically populated with the source file's structured content. The user can then open a stock comic panel (the project is using the open source DesignComics<sup>4</sup> panels) in column 3's editor and drag and drop step, prereq, example, etc. tags from column 2, which can be moved around the canvas and modified to become speech bubbles. Lastly, column 4 automatically generates the panel's XML transcription. **Figure 1** shows an example of columns 2 and 3, with extracted steps from a DITA task ready to be dropped into the editor. In its current version, the interface only works with DITA task topics, as phase 1 showed that concept and reference topics were easier to adapt into comics.

## 3. CONCLUSIONS

*TechCommix* facilitates the process of translating DITA tasks into comics and producing semantically representative XML for the resulting comic. The next phase will evaluate the efficacy of these comics in communicating the steps and other elements in the task. Additionally, it will include evaluating the usability of the comic computing environment and its resulting comics in contrast to comics produced using other tools, and in contrast to the effectiveness of source text-based tasks.

## 4. REFERENCES

- [1] R. Bailie. Comics for consumer communication: Reaching users with word and image. <http://www.bboxesandarrows.com/view/comics-for-consumer>, 2008.
- [2] L. Bellamy, M. Carey, and J. Schlotfeldt. *DITA Best Practices: A Roadmap for Writing, Editing, and Architecting in DITA*. IBM Press, 2012.
- [3] E. Noll Webb, G. Balasubramanian, U. OBroin, and J. Webb. Wham! pow! comics as user assistance. *Journal of Usability Studies*, 7(3):105–117, May 2012.
- [4] M. Opsteegh. What technical communicators can learn from comics. *TechniScribe: Orange County STC Newsletter*, 47(9), 2008.
- [5] A. Porter. Writing technical comics. *Intercom*, February 2010.
- [6] A. S. Pringle and S. S. O'Keefe. *Technical Writing 101: A Real-World Guide to Planning and Writing Technical Content*. Scriptorium Publishing Services, Incorporated, 3rd edition, 2009.
- [7] C. Robidoux. Rhetorically structured content: Developing a collaborative single-sourcing curriculum. *Technical Communication Quarterly*, 17(1):110–135, 2007.
- [8] Tobita, R. Comic computing: creation and communication with comic. In *Proceedings of the 29th ACM international conference on Design of communication (SIGDOC '11)*. ACM, New York, NY, USA, 91–98, 2011.
- [9] E. Yuh. Trendspotting: The comic as a technical communication tool. *Coast Lines: Newsletter blog of the Society of Technical Communicators Canada West Coast chapter*, December 2008.

<sup>3</sup> <https://code.google.com/p/svg-edit/>

<sup>4</sup> <http://designcomics.org>

<sup>2</sup> <http://www.bitstrips.com>