# Are Wikis Usable?

Alain Désilets
National Research Council of Canada
Institute for Information Technology
1-613-990-2813
alain.desilets@nrc-cnrc.gc.ca

Sébastien Paquet
National Research Council of Canada
Institute for Information Technology
1-506-861-0956
sebastien.paquet@nrc-cnrc.gc.ca

Norman G. Vinson
National Research Council of Canada
Institute for Information Technology
1-613-993-2565
norm.vinson@nrc-cnrc.gc.ca

## ABSTRACT

Wikis are simple to use, asynchronous, Web-based collaborative hypertext authoring systems which are quickly gaining in popularity. In spite of much anecdotal evidence to the effect that wikis are usable by non technical experts, this has never been studied formally. In this paper, we studied the usability of a wiki through observation and problem-solving interaction with several children who used the tool to collaboratively author hypertext stories over several sessions. The children received a minimal amount of instruction, but were able to ask for help during their work sessions. Despite minimal instruction, 5 out of 6 teams were able to complete their story. Our data indicate that the major usability problems were related to hyperlink management. We report on this and other usability issues, and provide suggestions for improving the usability of wikis. Our analysis and conclusions also apply to hypertext authoring with non wiki-based tools.

## Categories and Subject Descriptors

H5.2 [**User Interfaces**]: Ergonomics, User-centered design, Evaluation/methodology.

## General Terms

Human Factors

## Keywords

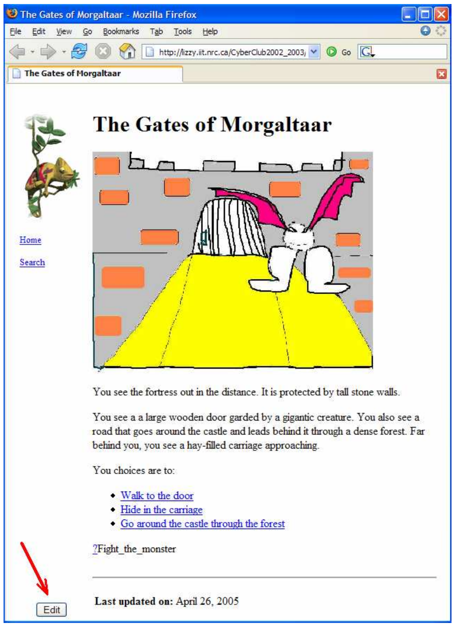
Wiki, Collaborative Web-Authoring, Groupware, Hypertext, Usability

## 1. INTRODUCTION

Wikis are simple to use, asynchronous, Web-based collaborative hypertext authoring systems. The original concept is due to programmer Ward Cunningham [12], whose prototype implementation has inspired many variants [15]. While a precise definition of wiki does not exist [11], the general consensus is that a wiki is *a collective website where a large number of participants are allowed to modify any page or create a new page using their Web browser*. Wiki introduced groundbreaking innovations at the level of technology for supporting collaborative web-authoring, but also at the level of the process, philosophy and even sociology [16, 24] of such collaborative authoring.

From the point of view of technology innovation, wiki introduced a new and simple way to edit web pages, and this at a time (1995) when the web was a read-only medium for all but the most

**Figure 1: Opening a wiki page for Editing**


The Gates of Morgaltaar - Mozilla Firefox
File
Edit
View
Go
Bookmarks
Tab
Tools
Help
http://lizzy.iit.nrc.ca/CyberClub2002_2003/
Go
The Gates of Morgaltaar
Home
Search
Edit

The Gates of Morgaltaar

You see the fortress out in the distance. It is protected by tall stone walls.

You see a a large wooden door garded by a gigantic creature. You also see a road that goes around the castle and leads behind it through a dense forest. Far behind you, you see a hay-filled carriage approaching.

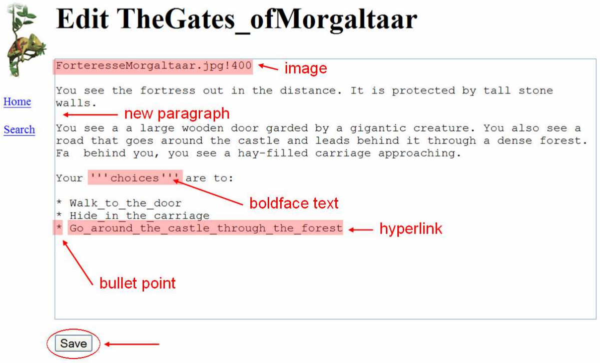
You choices are to:

- Walk to the door
- Hide in the carriage
- Go around the castle through the forest

?Fight_the_monster

**Last updated on:** April 26, 2005

**Figure 2: Editing the wiki page**



Edit TheGates_ofMorgaltaar

```
ForteresseMorgaltaar.jpg!400

You see the fortress out in the distance. It is protected by tall stone
walls.

You see a a large wooden door garded by a gigantic creature. You also see a
road that goes around the castle and leads behind it through a dense forest.
Fa  behind you, you see a hay-filled carriage approaching.

Your '''choices''' are to:

* Walk_to_the_door
* Hide_in_the_carriage
* Go_around_the_castle_through_the_forest
```

*WikiSym'05*, October 16–18, 2004, San Diego, CA, U.S.A.

ACM 1-59593-111-2/05/0010...$5.00.

technically adept users (i.e. webmasters). To edit a page on a wiki site, all a user needs to do is to click on the Edit button that appears on that page, modify the text that is then displayed in an editable field, and click on a Save button (see Figure 1). In most wiki systems, authors do not use a WYSIWYG ("What You See Is What You Get") editor to modify the content of a page. Instead, they use a text area to modify text that is marked up using a very simple syntax called wiki markup (see Figure 2).

Wiki also makes it very easy to create links between pages. All the user needs to do is open the page where she wants to insert a link, and then type the name of the page she wants to point to, using a special syntax called WikiWords (ex: `SomePage, some_page` or `[[some page]]`). When the user clicks on the **Save** button, the page appears featuring a new hyperlink. If the link points to a page that exists on the wiki site, it will look like a normal hyperlink to that page. If the link points to a new page that does not exist yet, it will typically appear as a question mark link (?) next to the WikiWord. Clicking on this question mark brings the user to a page like the one in

Figure 1, which allows the user to create the new page and type its content in an initially empty text field.

From the point of view of innovation in the **process, philosophy and sociology** of collaborative web authoring, wiki introduced a new way of thinking that favours:

- Content over Form
- Open Editing over Security and Control
- Free Form Content over Structured Content
- Incremental Growth over Upfront Design

While this may at first look like a recipe for disaster except for small and obscure web sites, it turns out to be a reasonable strategy with attributes that enable a wide variety of applications. Wikis have been used to collaboratively create and maintain software documentation (e.g. http://codex.wordpress.org/), Frequently Asked Questions (FAQ) repositories (e.g. http://allmyfaqs.com/), textbooks (e.g. http://wikibooks.org/), travel guides (e.g. http://wikitravel.org/) and specialized knowledge bases (e.g. http://iawiki.net, http://jugglingdb.com/jugglewiki/). One of the most high-profile wiki projects is **wikipedia** [32], which aims at creating a free, hyperlinked encyclopedia, and reached the respectable size of one million articles just before its fourth anniversary in September 2004.

The present paper is only concerned with wikis as a technological innovation. In particular, it will investigate the extent to which wikis are truly usable by non-technical people. There is anecdotal evidence for the usability of wikis: for instance, the fact that Wikipedia's 16,000+ contributors, many of them not technical people, are successfully building a hyperlinked encyclopedia using a wiki system. However the usability of wikis has not been formally studied to date and specific usability issues have not been identified.

The Usability Professionals Association defines usability as *"..the degree to which something - software, hardware or anything else - is easy to use and a good fit for the people who use it. It is a quality or characteristic of a product. It is whether a product is efficient, effective and satisfying for those who use it."* [30]. A high level of usability supports the widespread adoption of a product by non-technical users. Of course, usability is not the only factor affecting product adoption (think of cost, for example), but lack of usability can be a hindrance [21]. Therefore, a wiki that is usable is more likely to enjoy widespread adoption by non-technical users.

To identify the major usability problems with wiki, we observed and interacted with several children while they used a particular wiki implementation (Lizzy) to collaboratively author hypertext stories. Most of these interactions involved helping them solve the problems they faced, many of which were related to usability. Our analysis of these interactions (and other data) revealed the major usability problems of this wiki implementation. Due to the large overlap between features of different wiki implementations, these usability problems are common across many, if not all, wikis. In this paper, we describe our experiment and results, and provide suggestions for improving the usability of wikis.

The remainder of the paper is organized as follows. Section 2 briefly reviews related work. In Section 3, we describe the setup and methodology used for our experiment. In Section 4, we discuss results from our analysis of the data gathered during the experiment. We describe a number of high-level types of problems which we observed and compare their relative frequencies. We also make recommendations regarding how the Lizzy Wiki and wikis in general could be modified to make them more usable.

## 2. RELATED WORK

The usability of hypertext by readers or visitors has been examined rather extensively [e.g. 2, 20, 31], but the usability of hypertext authoring systems is largely understudied. Early work by Boyle, Teh and Williams [1] suggested that the systems with the simplest models are easiest to use (as measured by authoring speed), as seems to be the case with wiki. Elliott, Jones and Barker [8] proposed to use grounded theory to study the learnability of hypertext systems, but otherwise provided only very general findings.

As the analysis by Thimbleby [28] indicates, the general problem of authoring Web sites is quite complex. While design work on browser-based hypertext authoring systems does have a long history [3, 4, 22, 28], to the best of our knowledge, those systems have not been subjected to focused attention on identifying and solving usability problems.

Consequently, our investigation into the usability of wikis constitutes an important contribution to the literature on hypertext authoring.

## 3. EXPERIMENTAL DESIGN

### 3.1 Context and Subjects

The context in which this study took place was an extra-curricular activity involving collaboratively authoring a hypertext story with a custom implementation of wiki called Lizzy (see section 3.1.1). 15 grade 4 students (ages 8 and 9) at a French language public school participated in this activity. All subjects used a version of Lizzy with dialogs displayed entirely in French. 60% of the subjects were male. The activity took place at the school itself, in a lab with 25 computers. The students had been instructed in using the computers starting in Grade 1, but none had participated in the story writing activity or used a wiki before.

Although the group was smaller than a typical class, it was large enough to simulate conditions or a real class. This provides ecological validity for the study, supporting the generalization of the results to actual educational settings. Note however that these were extracurricular activities, and as such, they had no impact on school grades. Moreover, this study conformed to the exacting standards of the National Research Council of Canada's Research Ethics Board.

### *3.1.1 The Lizzy Wiki*

The Lizzy Wiki [13] used by subjects in our experiment was developed at the National Research Council of Canada, and is an extension of the original C2Wiki [5] implemented in Perl by Ward Cunningham.

Lizzy includes many unique features that distinguish it from other wiki implementations (including the original C2Wiki implementation). Below is a list of those particularities which are relevant to our study. For a full list of the special features of Lizzy, the reader is referred to [13].

#### *3.1.1.1 Link syntax*

With Lizzy, users can write links using a variety of WikiWord formats like: `SomeLink`, `some_link` `Some_link`, etc. Alternative formats all point to the same page, but they provide control over how the anchor text is rendered (Some Link, some link Some link). Users can also override the anchor text using the simple syntax: `page_name!anchor_text`. In a storytelling context, this was useful for hiding what lies behind a link from the reader. For example, by writing:

```
the_dungeon_of_death!open_the_door
```

the user could create a link open the door that lead to page `the_dungeon_of_death`, hence preserving the surprise effect.

The Lizzy link syntax also makes it easy to insert an image that resides on the wiki site. Instead of typing a fully qualified URL like:

http://lizzy.iit.nrc.ca/MyWikiSite/uploads/red_dragon.gif

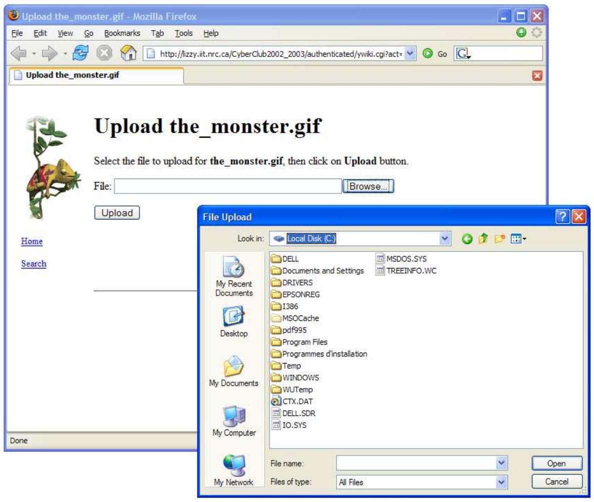
the user simply types the WikiWord for the image, followed by its image format extension (ex: `red_dragon.gif`). If the extension is a known image format, the WikiWord will be rendered as an inline image. In all other cases, it will be rendered as an ordinary link to the corresponding document.

In the case of a link to an inline image, the user can additionally control its display width (in number of pixels) by writing something like this: `red_dragon.gif!300`.

#### *3.1.1.2 Image and document uploading*

Lizzy allows users to easily upload images and documents (ex: MS-Word) to the wiki site, through a procedure similar to how you create new wiki pages. Like for new pages, the user starts by creating a link to the image or document she wants to upload (ex: `red_dragon.gif` for an image or `business_plan.doc` for a document). When she saves the page, this WikiWord appears with a ? link in front of it. The user can click this link to upload a new image or document onto the wiki site. When she clicks on it, she is presented with the dialog in Figure 3. Clicking on the **Browse** button, she can then navigate her local machine's file system to select the image or document she wants to upload to the wiki site.

**Figure 3: Lizzy upload dialog**



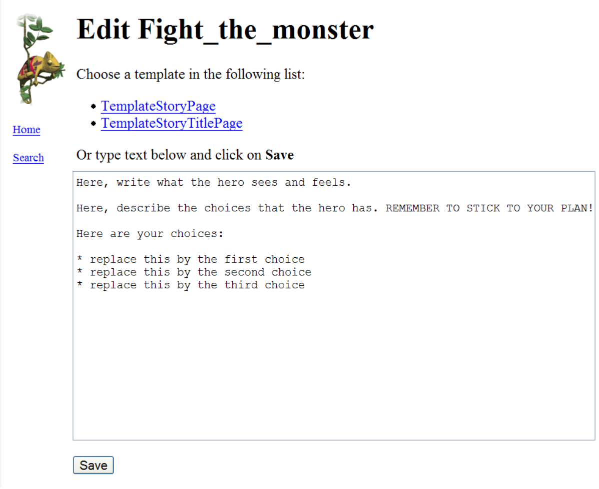
Upload the_monster.gif

Select the file to upload for **the_monster.gif**, then click on **Upload** button.

File: Browse...

Upload

**Figure 4: Lizzy template pages**



Edit Fight_the_monster

Choose a template in the following list:

- TemplateStoryPage
- TemplateStoryTitlePage

Or type text below and click on **Save**

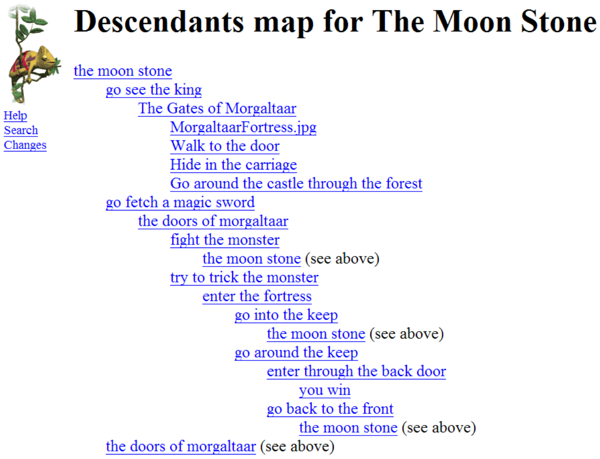
```
Here, write what the hero sees and feels.

Here, describe the choices that the hero has. REMEMBER TO STICK TO YOUR PLAN!

Here are your choices:

* replace this by the first choice
* replace this by the second choice
* replace this by the third choice
```

**Figure 5: Lizzy Link Map**



Descendants map for The Moon Stone

- the moon stone
  - go see the king
    - The Gates of Morgaltaar
      - MorgaltaarFortress.jpg
      - Walk to the door
      - Hide in the carriage
      - Go around the castle through the forest
  - go fetch a magic sword
    - the doors of morgaltaar
      - fight the monster
        - the moon stone (see above)
      - try to trick the monster
        - enter the fortress
          - go into the keep
            - the moon stone (see above)
          - go around the keep
            - enter through the back door
              - you win
            - go back to the front
              - the moon stone (see above)
  - the doors of morgaltaar (see above)

#### *3.1.1.3 Template pages*

Lizzy allows users to create new pages using templates. When a user creates a new page, she is presented with a list of possible templates. If she chooses one, she is presented with an edit page like in Figure 4.

#### *3.1.1.4 Rename button*

With Lizzy, users can easily rename a page while preserving the integrity of links that point to it. This is done by clicking on the **Rename** link which appears at the bottom of the page. When the user clicks on that link, she is prompted for a new name for the page. After clicking **OK**, the system displays the page under its new name, and it automatically changes all references to the old name in other pages, so that they now point to the new name.

#### *3.1.1.5 Link map*

To help users understand the hypertext's structure, Lizzy provides a **Link Map** link at the bottom of each page. When the user clicks on that link, the system shows her a map of all pages that can be reached from the current page within 7 mouse clicks. An example of this map is provided in Figure 5. This map shows not only pages that currently exist, but also dangling links that point to pages yet to be created. This is to help users get a better view of what work remains to be done in the vicinity of the current page.

Note that the **Link Map** only shows **downstream** pages, i.e. pages that can be reached from the current one. It does not show **upstream** pages, i.e. pages from which you can get to the current one.

### *3.1.2 Procedure and Tasks*

The collaborative storytelling activity that we observed took place over the course of 6 sessions in the spring 2004 semester, each session lasting 90 minutes. The first session introduced the subjects to the activity through an example wiki-based story, which had a protagonist, a goal, and a world. The story is structured as a set of states, (corresponding to locations in the story and pages of the wiki site) with state-to-state transitions (corresponding to actions in the story and links in the wiki site).

Following this introduction, the subjects formed teams of 2 to 5, and developed a synopsis of their own story, with the same structure as that described above. Subjects then drew a state transition diagram representing their story on a shared paper poster. The instructor verified the diagram to ensure it would serve as useful guide for implementing the story in a wiki site. In order to keep the workload manageable, the map had to conform to the constraints shown in Table 1. Figure 5 shows an example of such a story map.

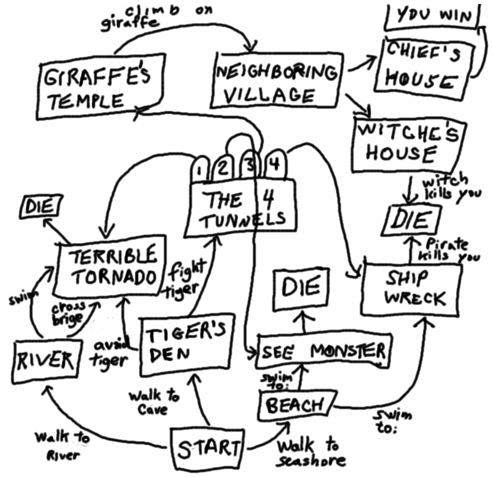
Between the first and second sessions, the instructor went on the wiki and created the start page of each story, without however including any content. In the second and subsequent sessions the subjects worked directly on the wiki site using their computers. Teammates sat side by side, each with their own computer, with the story map posted close by so that it could easily be consulted and annotated while implementing the story on the wiki. All stories were authored in a single wiki space shared by all teams.

In session 2, the instructor gave a 15-minute demo on creating new pages, typing in content, and linking the pages. The students then began their computer work by reproducing the story's link topography, which involved creating new pages, and adding some text content (no images at this point) to the pages. When creating new pages, students could use a template page like the one in Figure 4.

**Table 1: Story map constraints**

| **States (places/pages, drawn as nodes)** |
|---|
| Maximum of 5 per person on the team |
| Meaningful name |
| 1-3 lines of descriptive |
| **Transitions (events/links, drawn as arrows)** |
| A maximum of 3 links from each state |
| Each link has a title and a one line description of what the protagonist must do to move from one state to the other |

**Figure 6: Sample story map**



After each session beyond session 2, the instructor inspected the content of all pages for problems in the content or structure of the story. When those problems hinted at an usability issue (ex: a link that does not follow the WikiWord syntax), a description of the problem was included in our data. This is what we call **post-session observations** (see section 3.2 for details). For each problem, the instructor would additionally post a comment on the start page of the corresponding story, so that subjects would know about it and fix it in the next session.

In sessions 3 to 6, the subjects learned how to insert and resize images through a 15-minute demo, and they then worked to refine their text and images, and add new pages, if desired.

Throughout all sessions, the instructor and a colleague were available to answer questions.

## 3.2 Data Collection

There were two sources of data for this study which we call **in-session** and **post-session** data.

As we mentioned in section 3.1.2, the **post-session** data was collected by inspecting the subjects' work after each of the sessions to detect potential problems. Additional post-session data was also generated during the data analysis phase, when new types of problems were discovered and we went looking for instances of those in the content of the stories.

The **in-session** data was collected during each of the session with the subjects. This primarily involved subjects asking questions of the instructor when the need arose. These interactions with the instructor were recorded on audiotape. The time at which the interaction occurred was recorded, but the subject's identity was not. Consequently our data set is not divided by subject. Subjects tended to ask questions primarily when they were no longer able to move forward with their task.

Such ethnographic methods, involving the observation of subjects in naturalistic settings, are commonplace in interface design projects [19, 25]. Ethnographic methods provide a rich data set of observed events, which are typically classified into event categories that relate to work actions or interface features. The categorization we employed is discussed below.

## 3.3 Data Analysis

One researcher listened to the audio taped interactions between instructor and subjects (the **in-session** data). Whenever an interaction was relevant to the usability of the wiki, a concise textual description of the event was logged, along with a time stamp. These **in-session** events were combined with the **post-session** events that were logged based on inspection of the material produced by the subjects, to form a complete event list.

We categorized each event along two dimensions: **severity** and **type**. The severity category was used to quantify the degree to which the event was disruptive to the subject [21]. The type category was used to capture the particular area of the wiki system or the particular subtask to which the problem pertained. For both dimensions, categories were first assigned by one researcher and then reviewed by a second researcher, and differences of opinion were resolved through discussion. The categories for severity (**Catastrophe**, **Impasse** and **Annoyance**) were determined a-priori and assigned during the data analysis phase, but the categories for type emerged from the data analysis phase through a bottom up process. Each event was first assigned one or more low level types (ex: "problem pointing to an existing page"), and these low level types were later on grouped into higher level ones (ex: "link creation and management").

The a-priori severity categories were:

**Catastrophes:** problems that result in lost work or time, or in a site that requires a lot of work to repair.

> E.g. subject creates a brand new page when he should have pointed to an existing page instead, forcing merging of the two pages later on.

**Impasses:** problems that prevent the subject from continuing.

> E.g. subject cannot remember how to create a link to a page.

**Annoyances:** problems that neither prevent the subject from continuing nor have catastrophic consequences.

> E.g. subject does not remember how to resize an image on a wiki page, but decides she can live with the current size of the image.

Our bottom up classification of events into types yielded the following top level types:

**Hypertext:** This is an umbrella type that encompasses any problem which is due to the hypertext nature of the wiki medium. Note that the Hypertext category is a strict subset of the other categories (in other words, all Hypertext problems also appear in at least one other top-level category in this list, but there were some problems which were not of hypertext nature).

**Link creation and management:** The subject has difficulty creating and managing links to pages or images.

E.g. subject is trying to create a link to an existing page, but does not remember its exact name.

**Image uploading:** The subject has difficulty uploading images to the wiki site.

E.g. subject forgets to put an image extension after the WikiWord for an image.

**Creating/Editing pages:** The subject has difficulty creating and/or editing pages with the wiki.

E.g. subject forgets that he needs to press the Edit button to modify the content of a page.

**Hypertext authoring:** The subject has difficulty writing and organizing content in a way that makes sense for a hypertext medium.

E.g. subject writes content for a page in such a way that it only makes sense to the reader if he got to it through a particular path.

**Basic computer skills:** The wiki requires a certain computer skill which the subject does not possess.

E.g. subject does not know how to type the underscore character.

**Collaboration:** Teammates have difficulty collaborating using the wiki.

E.g. subject tries to modify a page at the same time as one of his teammates.

**Global Site View:** The subject has difficulty getting a global sense of the wiki site's structure and state of development.

E.g. subject believes his story is completed when in fact it still contains dangling links to pages yet to be created.

**Navigation:** The subject has trouble navigating the wiki site.

E.g. subject is unable to find a particular page.

**Interpersonal problems:** The subject experiences interpersonal conflict while collaborating using the wiki.

E.g. subject does not get along with his teammate and sits in opposite corner of the room, hence causing collaboration problems.

It is important to note that the data collection was somewhat biased toward the collection of impasses since these events were more likely to induce the subjects to interact with the instructor.

As with every case study, generalizability is an issue. In our case, some of the observed events may have been artifacts caused by features specific to our wiki implementation, by characteristics of the storytelling task, or by the age of the subjects.

However, most of the problems we observed are clearly common to all wikis. Exceptions to this will be noted explicitly in the paper. Also, we have had the opportunity to observe and gather feedback from adults using our Lizzy implementation for tasks other than storytelling. These included: collaborative maintenance of the web site for a small not-for-profit organization, sites for managing software projects and sites that acted as a knowledge repository for a particular topic. Our overall sense from informally observing those subjects is that the problems encountered by adults in those contexts were no different qualitatively from those encountered by our young subjects, although frequencies may differ. Again, whenever we have reason to believe that a particular observation is dependant on age or nature of the activity, we will note this explicitly in the paper.

# 4. RESULTS

## 4.1 Problem Frequencies

Table 2 shows the frequency by severity for each of the top-level problem types. The last row of this table gives the total number of unique problems for each of the severity types. Note that this total is not equal to the sum of the numbers in the corresponding column, because most problem instances were assigned to more than one type. Note also that the percentages in this table are expressed over those number of unique problem observations.

Overall, a total of 224 problems were observed. The most common severity level was impasse (95 or 42% of all problems), followed closely by catastrophes (84 or 37% of all problems).

Looking at Figure 7, we see that **Hypertext** and **Link creation and management** clearly stand out as the most important types of problem, respectively covering 62% and 49% of all observations. In comparison, the next most frequent type is **Hypertext authoring** which only covers 14% of observations. This finding is consistent with Calvi and Bra [2] who note that link creation and management is the most difficult aspect of hypertext authoring. Even though wikis were designed to make it easy to create hyperlinked pages, we see clearly that creating and managing links is still a problem.

Moreover, the **Hypertext** and **Link creation and management** types respectively account for 86% and 79% of the more severe problems of type **Catastrophe**.

In the remainder of this section, we describe each of the top problem categories in detail. For each type, we describe the problems we observed, and provide solutions to alleviate them in future wiki implementations. Furthermore, we make note of other features already implemented in existing wikis that might also alleviate those problems. Note that when discussing solutions, we focus on solutions that require as little client-side scripting as possible, in order to keep the system simple and browser-independent.

## 4.2 Hypertext

### *4.2.1 Problem description*

The vast majority of usability problems had to do with the **Hypertext** nature of the medium (62% of all problems). Because hypertext is more complex than linear content, it tended to cause difficulties in a range of different contexts like: editing, navigation and site design. It is worth noting that the hypertext nature of the medium caused much more problems than its collaborative nature (see section 4.8). Because all problems of type **Hypertext** also appeared in at least one other type category, we leave detailed discussion of these problems to other sections below.

### *4.2.2 Possible solutions*

Solutions to specific sub-types of **Hypertext** problems will be discussed in more details in the sections related to other problem types below.

## 4.3 Link creation and management

### *4.3.1 Problem description*

The second most common type of problem was **Link creation and management** (110 or 49% of all problems).

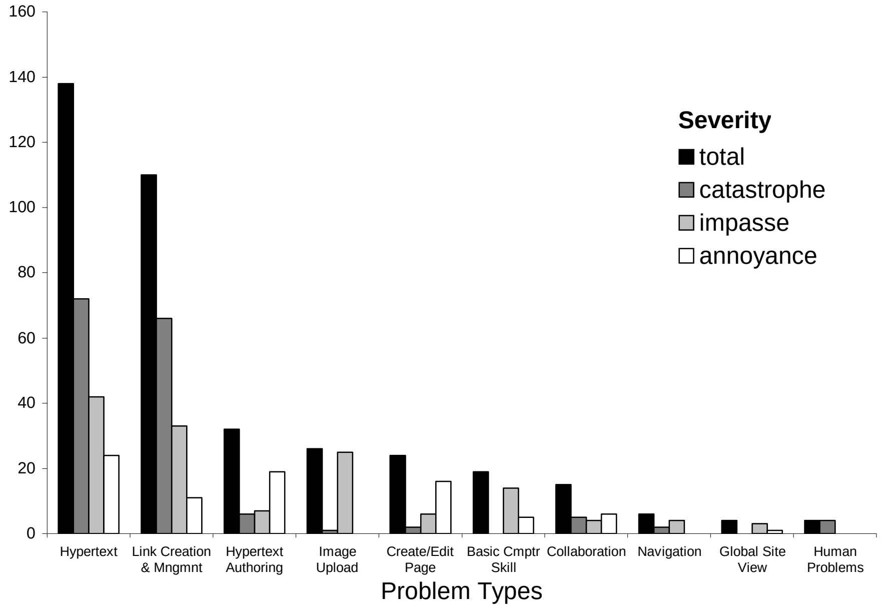
Out of those 110 problems, 46 (41%) were of subtype **Rename page**. It is worth noting that all **Rename page** problems were of **Catastrophe** severity, and that this subtype alone accounted for 55% of all catastrophic problems. The issue here was that subjects often misspelled names of pages when creating them, and had trouble renaming them afterwards, without breaking the links to that page. Subjects did not tend to use the Lizzy **Rename** link which sits at the bottom of each page. Instead, they seemed to want to rename the page by manipulating a link that points to it. For example, if they wanted to rename a misspelled page `red_devell`, they would do so by editing a page that pointed to `red_devell`, and then correcting the WikiWord to `red_devil`. But instead of renaming the page from `red_devell` to `red_devil`, this ended up creating a dangling ? link for creating a new page called `red_devil`. This type of problem usually had catastrophic consequences, because subjects would click on that ? link and retype the content of the original `red_devell` page from memory. Often, subjects did not realize until much later that they had two versions of the same page, and when they did, they had to somehow merge the two. Note that the large proportion of page renaming problems is probably due to the young age of our subjects. In our experience, adult users do not routinely misspell page names, and therefore do not have to rename them as frequently. But when they do need to rename pages, they tend to experience the same issues as our young subjects.

Other very common **Link creation and management** problems (40 out of 110) were fairly shallow issues having to do with the syntax of links. Repeatedly, we had to remind the subjects to put underscores between words (instead of white spaces) when creating links. In our experience, most adults seem to learn this syntax after being told once. Another difficulty subjects had with the link syntax was how to override the anchor text of a link to a page, or changing the size of an inlined image. For both tasks, many subjects simply did not remember that they needed to use the special exclamation mark syntax (see section 3.1.1.1). In our experience, adults also have difficulty with those last two points. While anchor overriding and image resizing are Lizzy-specific features, we would expect other implementations to encounter similar issues if they try to also support those features using a special link syntax.

**Table 2: Observation frequencies for top level problem categories**

| | Frequency | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Total | | Catastrophe | | Impasse | | Annoyance | |
| *Hypertext* | 138 | 62% | 72 | 86% | 42 | 44% | 24 | 53% |
| *Link creation and management* | 110 | 49% | 66 | 79% | 33 | 35% | 11 | 24% |
| *Hypertext authoring* | 32 | 14% | 6 | 7% | 7 | 7% | 19 | 42% |
| *Image uploading* | 26 | 12% | 1 | 1% | 25 | 26% | 0 | 0% |
| *Creating/Editing pages* | 24 | 11% | 2 | 2% | 6 | 6% | 16 | 36% |
| *Basic computer skills* | 19 | 8% | 0 | 0% | 14 | 15% | 5 | 11% |
| *Collaboration* | 15 | 7% | 5 | 6% | 4 | 4% | 6 | 13% |
| *Navigation* | 6 | 3% | 2 | 2% | 4 | 4% | 0 | 0% |
| *Global Site View* | 4 | 2% | 0 | 0% | 3 | 3% | 1 | 2% |
| *Human problems* | 4 | 2% | 4 | 5% | 0 | 0% | 0 | 0% |
| | | | | | | | | |
| *Total* | 224 | | 84 | | 95 | | 45 | |

**Figure 7: Histogram of frequencies for the various problem types and severities**



Another type of **Link creation and management** problems (13 out of 110) was that subjects had trouble remembering the exact name of the page or image they wanted to point to, or they would misspell it. For example, they might type `the_devil` or `the_red_devell` in an attempt to point to a page actually called `the_red_devil`, which would result in a ? link for creating a new page. Like for problems involving page renaming, this usually had catastrophic consequences because subjects clicked on that ? link and duplicated page `the_red_devil` under a different name.

Another type of **Link creation and management** problem (5 cases) had to do with ensuring and preserving the correctness of the story's topology (i.e. the structure of links between the various pages). Sometimes we noticed in the post-session analysis that stories contained orphan pages (1 case) and self-referencing pages (1 case) and multiple copies of a page (1 case). In 2 cases, the user had difficulty transposing the topology he saw on the paper map, onto the wiki site. For example, he might create a direct link between two pages, which, on the paper map, were connected only indirectly through a third page.

It is worth noting that many of the **Link creation and management** problems seemed to be caused by confusion in the subject's mind between different concepts related to links, namely: the link itself, its anchor text (i.e. text displayed for the link), the name of the page it currently points to, and the actual page that lies at the other end of the link. This is very similar to the confusion that novice C programmers experience when they first start working with pointers [23], and is best illustrated by an example. Suppose a subject edits a page and changes a WikiWord from `old_wikiword` to `modified_wikiword`. We observed that in some cases, the subject would correctly expect the wiki to to reorient the link to point at page `modified_wikiword`, but keep page `old_wikiword` intact. But in other cases, then expected the wiki to simply change the anchor text to modified wikiword, but leave the link pointing at page `old_wikiword`. In other cases still, the subject would expect the wiki to actually rename page `old_wikiword` to `modified_wikiword` and to make the link point to that renamed page. Note that we observed these different expectations not only between different subjects, but sometimes for a same subject at different points in time.

The confusion between anchor text and name of the target page also often manifested itself in a different guise. Often, subjects would type extraneous action words in the WikiWord for a link. For example, in trying to create a link to a page `the_dungeon`, they might type `return_to_the_dungeon` instead of `return to the_dungeon`.

It is worth noting also that with the exception of anchor overriding and image resizing, problems described in this section are common to most, if not all wikis. They are not specific to the Lizzy Wiki nor to the storytelling nature of the activity we observed. We have also observed essentially the same problems in our informal interactions with adult users.

### 4.3.2 Possible solutions

Most of the problems described above indicate that manipulating links through the raw wiki syntax is not appropriate. This is a very significant result for wiki developers. The raw wiki syntax for links simply does not convey the complexities of links clearly enough, and users have difficulty carrying out link creation and management tasks by manipulating this raw syntax directly.

One way to address this would be for the wiki to provide link wizards to carry out specific tasks like: pointing to a page without knowing its exact name, changing the anchor text of a link or the size of an image, renaming a page or image and keep the integrity of references to it. Such wizards could easily be provided without client-side scripting and without making the system browser dependant.

The system could also make it easier to preserve the integrity of the topology, for example by warning the user when a particular edit operation results in the creation of an orphan or self-referencing page, or by providing admin tools that allow the user to search for and repair this type of problems. The link wizard described above would also help in that respect, by making page renaming more intuitive. Most users seem to want to rename a page through a link that points to it instead of the **Rename** link on the page.

Several wiki implementations have syntax wizards that help users create links: Wikipedia [32], SocialText [27] to name a few. But those only help with the surface feature of the syntax (e.g., joining words with underscore to create a link), and they are limited to the basic syntax (e.g., no anchor overriding). They don't help with deeper issues like not knowing the exact name of the page you want to point to. When you click on a dangling link to create a new page, Wikipedia does offer you to search for a page that is similar to the anchor text of the dangling link, but then you have to copy and paste the name of the page once you find it. Also, the search option is not prominent, so users are not likely to notice it. It would be better if the system prominently showed likely candidates for link destinations. Other wikis like JotSpot [10] and Flash [9] try to go even one step further, and provide a WYSIWYG editor that runs inside the browser. However, at the moment of this writing, neither of those implementations worked well with all versions of Internet Explorer, Firefox, Safari (the Mac browser) on all OS platforms.

A feature that would help prevent link problems for children (but not necessarily for adults) would be a spell checker that warns them when they are about to create a link that contains a spelling mistake. MoinMoin [18] and the original C2Wiki [5] both implement spell checking of pages, however they are entirely form-based and very awkward to use. Moreover, those spell checkers do not automatically bring the attention of the user to spelling mistakes in link names (the user must explicitly press the Spell Check button). More interactive spell-checkers that highlight errors as soon as they are typed could be developed, but they would require a significant amount of client-side scripting, and they are not likely to work well across browsers.

## 4.4 Hypertext authoring

### 4.4.1 Problem description

The third most frequent problem type (32 or 14% of all problems) had to do with **Hypertext authoring** issues where the subject had difficulty writing and organizing content in a way that makes sense for a hypertext medium.

For example, some subjects had trouble planning the topology of their story (8 out of 32). Others had difficulty with writing text so that it makes sense in a hypertext medium (14 out of 32). For example, they would write a page or chose its name in a way that made sense if the reader came to it from a particular page, but would not make sense if he came to it from another page (9 out of 32). Or they would not know what to write on a terminal page where the reader wins or dies (7 out of 32). Or they would not know how to write a first page introduction for a story that is hypertext in nature (2 out of 32).

### 4.4.2 Possible solutions

**Hypertext authoring** problems do not seem to have a technological solution. Such problems would probably be best solved by educating the user about the differences between writing in a linear versus hypertext style.

## 4.5 Image uploading

### 4.5.1 Problem description

The next most frequent problem type was **Image uploading** (12% of all problems). This may be an artifact of the storytelling activity (which required that each page contain an illustration) or the specific procedure used in Lizzy for uploading. However, we have noticed that adult users involved in non-storytelling activities also experience many difficulties with the uploading procedure for images and documents. Moreover, almost all problems with Image uploading (22 out of 26) have to do with the lack of integration between the tool used to create the image and the wiki, and this is an issue that is common to all wiki implementations, not just Lizzy.

Because of the lack of integration between the image authoring tool and the browser, our subjects had to create the image with the drawing tool, save it to their hard disk, then locate that image using the OS' files system browsing and upload it to the site (see Figure 3). This required too many operations and included too many opportunities for errors. For example, subjects would save the image to one location on their hard disk and then not be able to find it again through the wiki. Or they would end up choosing the wrong image. Sometimes, they would use a WikiWord with an image format extension that did not correspond to the format of the image on their hard disk. At other times, the name they had used for the image on the wiki was different from the name they had used to save the image on their hard disk and this confused them. Things were even worse in cases where a subject wanted to modify an image that was already on the wiki site. In those cases, she had to download the image to her hard drive, modify it with the drawing tool, and then upload it back to the wiki site.

One problem that did not have to do with tool integration (4 cases out of 26) was that the upload would often take too long and either the connection would time out or the subject would give up and press the Back button. In the later case, when they tried uploading again, the system would tell them that the file already existed, which would confuse them. The reason the upload took so long was that the image drawing tool used by the subjects could only save images in uncompressed BMP format.

### *4.5.2 Possible solutions*

Problems which were due to the lack of integration between tools are difficult to resolve. Ideally, the wiki should support some sort of drag-and-drop or copy-and-paste capability that would allow users to transfer the content of the drawing tool directly into the wiki for uploading. Unfortunately, Web browsers do not support this kind of capability. Consequently, there does not seem to be a good solution for this particular type of problem.

Problems where the user specifies an image format in the wiki that differed from the format of the image saved on disk could however be dealt with through a link wizard approach (as per section 4.3.2). The issue of image upload taking too long would be best dealt with outside of the scope of the wiki by providing the subjects with an image drawing tool capable of saving images in compressed form (GIF or JPEG). However, one can imagine cases where large documents of other types (ex: audio, video, PowerPoint) could take too long to upload. One way to avoid the user giving up would be to warn them that this could take a long time. A better solution still would be to provide a progress bar and a cancel button, but it is not clear that such functionality could be provided in a browser-independent fashion.

## 4.6 Creating/Editing pages

### *4.6.1 Problem description*

Close to half of the problems in this type had to do with wiki markup (10 out of 24). A typical example is a subject forgetting to separate paragraphs with a blank line, which resulted in the two paragraphs being rendered as a single one. All but one of those markup problems were of annoyance level. They did not prevent the subject from continuing with the task and did not have irremediable or hard to recover consequences. At worst, the text ended up looking not quite the way the subject wanted it to look.

The second half (13 out of 24) of the problems had to do with the sequence of steps required to create or open a page, edit its content and save it. For example, some subjects might forget that they must click on the Edit button (1 case) before they can modify the content, or once they opened the page and modified its content, they would not know to press the Save button (2 cases). In other cases, subjects would not know which page template to select for creating a page (2 Cases), or they would be confused by the instructions provided on the template page (4 cases). Often, subjects did not understand that in order to create a new page, they should first create a link to it on some page, then save that page and use the ? link in front of the WikiWord (4 cases).

### *4.6.2 Possible solutions*

The ideal way to deal with the markup problems would be for wiki to provide a WYSIWYG editor. But this would require heavy client-side scripting, and as pointed out in section 4.3.2, it is not clear that it could be implemented in a browser-independent fashion. A simpler approach would be to implement wiki markup wizards that would assist users in tasks like: marking up text as bold, italic, bullet points, etc. This too requires client-side scripting but it is more lightweight and has been successfully implemented in a browser-independent fashion by a number of wikis (e.g., Xwiki [33], SocialText [27], and MediaWiki [17]). However, it is not clear whether even that lightweight solution is warranted. In our experience, adult users at least, quickly master simple wiki markup.

For the problems related to the page creation and editing procedure, there does not seem to be any unified solutions that would deal with all of them. They are mostly independent problems, each of them occurring infrequently. The fact that none of those individual problems is reoccurring seems to indicate that once subjects have learned the solution to the problem, they do not encounter it afterwards. The (admittedly infrequent) problem with users not knowing to press the **Edit** button to edit the page might be solved by having the wiki move to the edit page whenever the user clicks somewhere in the page. This would be simple to implement in a browser-independent way with a small amount of client-side scripting. But given how infrequent the problem was in the fist place, this highly non-standard behavior might cause more confusion than it is worth. For example, the user might accidentally click on the page when she does not mean to edit it, and be confused by the system's behavior at that point. Some wikis like SocialText [27] let users create new pages by typing a name into a box, or clicking a **New Page** link, instead of creating a link from an existing page. While this makes it more straightforward to create a new page, it initially leaves the page in a state where it is unconnected to the rest of the site, and requires additional steps to connect it afterwards. Creating such a link after the fact may be difficult for most users, given that most users have difficulty with **Link creation and management**.

## 4.7 Basic computer skills

### *4.7.1 Problem description*

Most problems of that type were not related to the wiki per se, but had to do with some other tool or some generic computer skill. For example, many subjects had trouble remembering their userid and password or the URL for the wiki site (13 cases). Other subjects had difficulty using the web browser (1 case). Others did not know how to use the image drawing tool (2 cases), or how to search for images on the web (1 cases). Some subjects had trouble using the keyboard, especially when it came to typing special characters like underscore (1 case).

### *4.7.2 Possible solutions*

The solution to most of these problems is outside the scope of the wiki. However, the issue with typing special characters could be dealt with through link and syntax wizards (as per section 4.3.2 and 4.6.2).

## 4.8 Collaboration

### *4.8.1 Problem description*

In general, the subjects we observed were very good at using the wiki as a collaborative tool, and this is evidenced by the relatively small number (15 cases) of **Collaboration** problems. For a more thorough discussion of the collaboration aspects of our experiment, see Désilets and Paquet [6].

Some of this ease of collaboration may be an artifact of experimental setup, because teammates were co-located the whole time and had a paper map of the story which they could use as the basis for coordination. But part of it can also be attributed to the fact that hypertext authoring is highly parallelizable. Because the content is split into a number of small pages, this makes it easy to divide work amongst teammates, and it decreases the likelihood of an **edit collision** (i.e. two teammates trying to change the same page at the same time). Indeed, our experience in informally observing adult users who are not co-located, is that collaboration through a wiki is very easy there too.

The bulk of **Collaboration** problems we observed (8 cases) had to do with coordinating the division of labor between teammates. This happened mostly at the beginning of the activity, when too few pages had been written to allow teammates to work concurrently without interfering with each other. For example, when there were initially fewer pages than the number of members in the team, some teams did not know how to keep all team members busy. Even after the story had grown to have more pages than teammates, teams could still experience problems when the story was still relatively small. This was because the small number of pages increased the likelihood of **edit collisions**. In such cases, modifications made by one of the team members would be lost because the wiki did not have file locking mechanism.

Another type of coordination issue (3 cases) was that sometimes the writing style was not consistent from one page to the next. For example, different members of a same team might use a different voice for the narrative (ex: “You die.” versus “The knight dies.”).

We observed a small number (2 cases) of **page name collisions**, meaning that two people want to use the same page name for two different purposes. In one case, the collision was within a same team, but in another case, members of two separate teams working on different stories wanted to use a same fairly generic page name like: `the_dungeon`.

Because our activity setup was not typical of most wiki uses, it is interesting to ask whether our observations are representative of the **Collaboration** problems that would occur if teammates were not co-located and did not have a shared paper map to help in coordinating the work. Based on our experience setting up and supporting various wiki sites for collaboration, we believe that they are. Like in our experiment, **edit collisions** seem relatively rare out in the “real world”, even when teammates are not co-located and have no explicit way of coordinating. Like in our experiment, **name page collision** happen somewhat more frequently than **edit collisions**, but the second user typically solves the problem easily by choosing a slightly different (often more specific) name for his page. And like in our experiment, inconsistent writing styles across a same site are very common in the real world. The one type of **Collaboration** problems that we believe is an artifact of our experimental setup is problems encountered because of the initial small number of pages. These problems occurred because all teammates started working on the story at the exact same time, which is not typical of how wikis start in the “real world”. Typically, when a new wiki is created, someone starts by seeding it with a number of pages to which other contributors add to later on.

### *4.8.2 Possible solutions*

Although **edit collisions** are fairly rare, they have dire consequences (one set of edits being lost), so it is worth thinking about possible solutions. The simplest solution would be to have some form of locking mechanism that would prevent anyone from editing a page while it is being modified by another user. Unfortunately such locking mechanism often results in pages being accidentally locked “forever”, for example if a user opens the page for editing and never saves his edits. One way to address this issue is to use **soft locks** which can be overridden by the second user who opens a page, or that expire automatically after a reasonable period of time. Some wikis (e.g., MoinMoin [18]) implement this kind of time-limited page soft locking. Others (e.g., TikiWiki [29], and SnipSnap [26]) enable select users to do hard locking. But such soft locks do not solve the problem entirely because there is still a possibility that one person will overwrite changes made by someone else. Whenever this happens, the wiki should warn the user and provide her with an intuitive interface that allow her to merge her changes with those of the previous user (for example, see Figure 8). However, such a page merging interface would require a significant amount of client-side scripting, and it is not clear to what extent non-technical users would be able to use it.

Désilets and Paquet [6] described how subjects used the paper map of their story for coordination in a number of informal ways that had not been prescribed by the instructor (ex: subject writing his name beside a node on the map to indicate that he is working on that page, or a subject crossing a node to indicate that this page is completed). We suggested that implementing an electronic equivalent to this paper map might help team collaboration in a non co-located situation, by providing a shared annotatable global view of the site. One way to implement this would be to augment the Lizzy links map with textual annotations.

Some wiki implementations (e.g., MASE [14]) address the **page name collisions** problem by allowing users to put pages into separate namespaces (for example, one space could be devoted to each story). One disadvantage of this approach is that it makes hyperlinking more difficult because one has to be aware that pages may reside in different spaces. Considering that page name collisions are fairly rare and easily resolvable while problems with the creation and management of links are highly frequent and hard to resolve, such a “solution” may be causing more problems than it solves. Also, this approach would not have solved page name conflicts that occurred within a same team, because a same team would have shared a common space on the server.

There does not seem to be an easy technological solution to the problems of inconsistent writing styles. This is probably better dealt with by educating authors about the importance of reading each others texts (especially at the start of the story writing) and harmonizing their styles. Technological solutions could however be devised to supoort this, for example, prominently displaying a list of recently changed pages.

## 4.9 Global Site View

### *4.9.1 Problem description*

The wiki interface essentially presents the site one page at a time, making it difficult to get a sense of the structure and state of development of a whole region of the site. This caused problems for the subjects. The frequency of such problems was probably smaller in our study than it would be in a normal situation, because in our setup, subjects had a paper map of the whole site in plain view all the time, and they used it often for assessing and navigating their story.

All 4 of those problems were instances where subjects had difficulty figuring out what work was left to do in the story. Subjects who had finished a page would go looking for work to do but would fail in the attempt.

### *4.9.2 Possible solutions*

Whenever students did not know how to find work to do, we suggested that they invoke the Lizzy link map from the title page of their story. This allowed them to see their whole story at once. Pages that had a link to them but did not exist appeared with a dangling ? link in front of their names, signaling to the user that those pages had yet to be created. In spite of that, it seemed subjects often forgot about the link map functionality, or invoked it from a page situated in the middle of their story. Since the link map only shows downstream pages, this only gave them a view of part of the story. Maybe the link map could be made more prominent by having it displayed on the left side of the page all the time (at least when the user is editing the site). Also, the link map should probably show pages that are upstream as well as pages that are downstream of the current one, so that users can get a global view of the whole story even when they activate the map from a page in the middle of the story.

The textual annotations to the link map proposed in section 4.8.2 could also help by allowing users to tag a particular page with a list of things that remain to be done on it.

## 4.10 Navigation

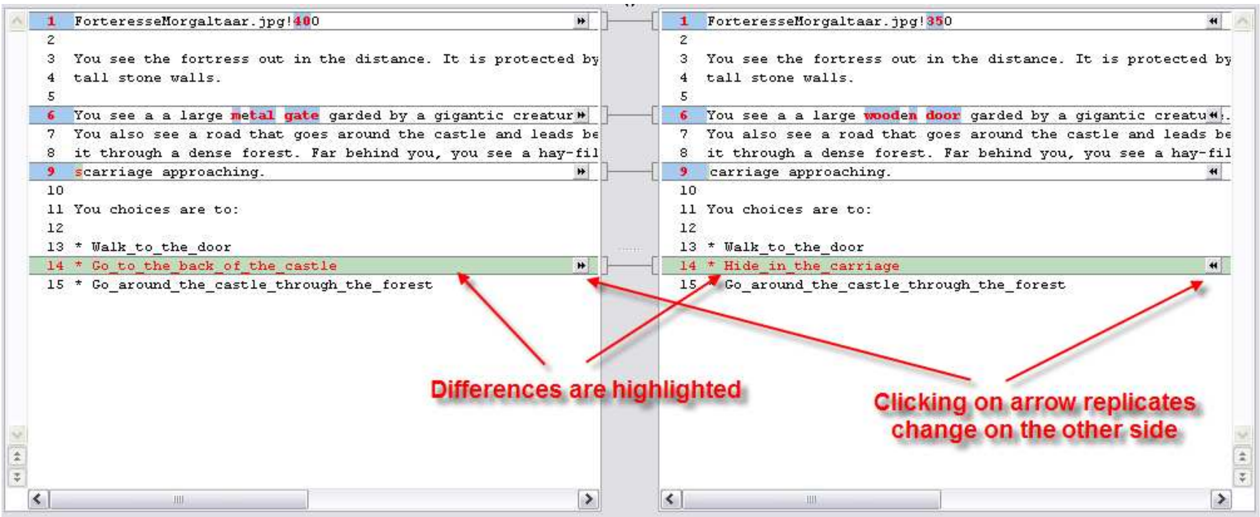
### *4.10.1 Problem description*

Our subjects experienced very few navigation problems (6 cases). This is surprising since navigation is such a common problem in web browsing and web authoring [7]. However, this abnormally small number of navigation problems may be more an artifact of our experimental setup than something that is due to the ease of navigation with wiki. Indeed, all teams had a poster size graphical map of their story in plain view all the time and they used it extensively to orient themselves around the site.

In 3 of the cases, the problem was that the subject looked for a particular page and could not find it. In 2 of those cases, the subject had even thought of using the **Link Map**, but had invoked it from a page that was downstream from the one he was looking for. Since the **Link Map** only shows pages which are downstream from the current one, the page the subject was looking for did not appear on the map.

In 3 other cases, the problem was caused by navigation using the browser's **Back** button. For example, a subject might modify a page, then move through a sequence of other pages. At some point, the subject would repeatedly press Back to return to the first page, but he would overshoot by one and end up seeing an old version of the page, as it stood before his change. On the rare

**Figure 8: A change merging interface**



```
ForteresseMorgaltaar.jpg!400

You see the fortress out in the distance. It is protected by
tall stone walls.

You see a a large metal gate garded by a gigantic creatur
You also see a road that goes around the castle and leads be
it through a dense forest. Far behind you, you see a hay-fil
scarriage approaching.

You choices are to:

* Walk_to_the_door
* Go_to_the_back_of_the_castle
* Go_around_the_castle_through_the_forest
```

```
ForteresseMorgaltaar.jpg!350

You see the fortress out in the distance. It is protected by
tall stone walls.

You see a a large wooden door garded by a gigantic creatu
You also see a road that goes around the castle and leads be
it through a dense forest. Far behind you, you see a hay-fil
carriage approaching.

You choices are to:

* Walk_to_the_door
* Hide_in_the_carriage
* Go_around_the_castle_through_the_forest
```

Differences are highlighted
Clicking on arrow replicates change on the other side

occasion when this happened, it would be very disconcerting to the subject who thought he had lost his changes to that page.

### *4.10.2 Possible solutions*

Problems with the Link Map could be solved by having it display both upstream and downstream pages. Problems caused by the Back button are generic to all web sites that display dynamic content whose state can become stale. At present time, there does not seem to be a good solution to this issue.

## 4.11 Interpersonal problems

### *4.11.1 Problem description*

Some subjects experienced interpersonal conflicts in the course of collaborating using the wiki. These problems were not attributable to the wiki itself and this is why they are listed in a separate type from the **Collaboration** type. They have no bearing on its usability, but they are included here for completeness.

### *4.11.2 Possible solutions*

Solutions to interpersonal problems are outside the scope of the wiki tool.

# 5. CONCLUSION

Overall our study indicates that wiki (or at least our Lizzy implementation) is indeed usable by non-technical users. The fact that a class of 15 Grade 4 children can use it to collaboratively create complex web-based stories with only two 15 minute training sessions attests to that. Though the instructor was present to answer the children’s, Désilets has administered wiki sites for which there was no live help present but were nonetheless used successfully by non-technical adult users. This is consistent with the success and size of the WikiPedia on-line encyclopoedia.

However the study also identified types of usability issues that were encountered by our subjects. The most important type had to do with the creation and management of links to pages and images, and it accounted for 49% of all problems and for 79% of problems with catastrophic consequences.

We have described possible solutions for some of the problems encountered by our subjects and plan to test some of those ideas (particularly those related to link management) in our next observation session. Other questions of interest include learnability of wiki. For example, does the frequency of errors decrease over time as users learn to use the system, and if so, how rapidly? Does the nature of the errors change over time? Another interesting question concerns usability of wiki in more typical situations where collaborators are not co-located and are not working on a storytelling task.

# 6. REFERENCES

[1]. Boyle, C.D.B., Teh, S.H. and Williams, C. An empirical evaluation of hypertext interfaces. *Hypermedia, 2* (1990) 235-248.

[2]. Calvi, L. and Bra, P.D. Improving the usability of hypertext courseware through adaptive linking in *Proceedings of the eighth ACM conference on Hypertext* ACM Press, Southampton, United Kingdom 1997 224-225

[3]. Chang, B.W. In-place editing of Web pages: Sparrow community-shared documents. *Computer Networks and ISDN Systems (WWW7 Proceedings Special Issue), 30* (1998) 489-498.

[4]. Crespo, A. and Bier, E.A., WebWriter: a browser-based editor for constructing Web applications. In *Proceedings of Www5*, (Paris, France, May 6-10, 1996).

[5]. Cunningham, W. *WikiWiki, a.k.a c2*, http://c2.com/cgi/wiki?WikiWikiWeb, 2005.

[6]. Désilets, A. and Paquet, S., Wiki as a tool for Web-based collaborative story telling in primary school: A case study. In *Proceedings of Ed-Media 2005*, (Montreal, Canada, June 27-July 2, 2005).

[7]. Dias, P. and Sousa, A.P. Understanding navigation and disorientation in hypermedia learning environments. *Journal of Educational Multimedia and Hypermedia, 6* (1997) 173-186.

[8]. Elliott, G.J., Jones, E. and Barker, P. A grounded theory approach to modelling learnability of hypermedia authoring tools. *Interacting with Computers*, *14* (2002) 547-574.

[9]. Flash. http://www.whatisflash.com/

[10]. JotSpot. http://www.jot.com/

[11]. Lamb, B. Wide Open Spaces: Wikis, Ready or Not. *EDUCAUSE Review*, *39* (2004) 5, 36-48.

[12]. Leuf, B. and Cunningham, W. *The Wiki Way: Collaboration and Sharing on the Internet*. Addison-Wesley, 2001.

[13]. Lizzy. http://lizzy.iit.nrc.ca/

[14]. MASE. http://sern.ucalgary.ca/~milos/mase/M-ASE.htm

[15]. Mattison, D. QuickiWiki, SWiki, TWiki, ZWiki and the Plone Wars: Wiki as a PIM and Collaborative Content Tool. *Searcher: The Magazine for Database Professionals, 23* (2003).

[16]. Meatball Collective. *Meatball Wiki*, http://www.usemod.com/cgi-bin/mb.pl, 2005.

[17]. MediaWiki. http://wikipedia.sourceforge.net/

[18]. MoinMoin. http://moinmoin.wikiwikiweb.de, 2005.

[19]. Nardi, B. The use of ethnographic methods in design and evaluation. in Helander, M., Landauer, T. and Prabhu, P. eds. *The handbook of human-computer interaction*, Elsevier, 1997, 361-366.

[20]. Nielsen, J. *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing, 2000.

[21]. Nielsen, J. *Usability Engineering*. Morgan Kaufmann, 1993.

[22]. Pier, K., Bier, E.A., Fishkin, K. and Stone, M., WebEdit: shared editing in a Web browser. In *Proceedings of WWW4 (Poster)*, (Boston, USA, December, 1995).

[23]. Postner, L. and Turns, J., Using facet-based assessment to understand introductory programming. In *Proceedings of FIE 2002*, (Boston, MA, November, 2002).

[24]. Shah, S., The Internet is Jain: How Gunslingin' Technolibertarianism Leads to Lotus Petals. In *Proceedings of New Forms Festival: Technography*, (Vancouver, B.C.2004).

[25]. Shneiderman, B. *Designing the User Interface: Strategies for Effective Human-Computer-Interaction (3rd ed.)*. Addison Wesley Longman, 1998.

[26]. SnipSnap. http://snipsnap.org, 2005.

[27]. SocialText (2005).

[28]. Thimbleby, H.W. Gentler: a tool for systematic web authoring. *International Journal of Human-Computer Studies*, *47* (1997) 139-168.

[29]. TikiWiki. http://tikiwiki.org, 2005.

[30]. Usability Professionals Association. *Resources: About Usability*, http://www.upassoc.org/usability_resources/about_usability/, 2005.

[31]. van Nimwegen, C., Pouw, M. and van Oostendorp, H. The influence of structure and reading-manipulation on usability of hypertexts. *Interacting with Computers, 12* (1999) 7-21.

[32]. wikipedia. http://wikipedia.org/, 2005.

[33]. Xwiki. http://www.Xwiki.org/

The authors wish to thank the following people for their assistance. René Bastien, director of École Côte-du-Nord, for allowing us into his school and for his unconditional support to the project. The following people who helped develop Lizzy, the Wiki tool used for this experiment: Hakan Erdogmus, Marta Stojanovic and Patrick Paul. And last but not least, all the wonderful students at École Côte-du-Nord who participated in the storytelling activity.