

TC 518

Use Scenario – Task & Context: “TiddlyThesaurus”

Title: Instructor evaluates product as a learning tool

Situation: Brent decides to “try out” the TiddlyThesaurus to determine if it would be a useful tool for his class to use in LIS 537 Construction of Indexing Languages. Since adding terms to a thesaurus is a common activity for his students, he decides to try to add a new term to an existing TiddlyThesaurus example. He begins this task with his students in mind...

Method to address the situation:

- Read the documentation
- Review the example thesaurus
- Decide on a term to add
- Navigate to the page
- Log in
- Decide where the new term should go
 - Use the Classified Schedule
 - (OR) Use the search box
- Add the new term
 - Create new term from existing term
 - (OR) Create new term from scratch
- Verify that term relationships are correct
 - Yes, they are correct → move on to generate notation
 - No, they are broken → fix the problem and verify again
- Generate notation
- Save the updated thesaurus
- Produce output (ie. print)

Execution path:

Step 1: Read the documentation

Because he is methodical, Brent is one of those rare people who actually reads documentation before starting to use something new. When evaluating a resource for students to use, this is even more important. While the TiddlyThesaurus appears to have some minimal amount of documentation, he was hoping for a lot more. Since this is to be a learning tool for students, he would have liked separate documentation for instructors and for students. Having such targeted documentation would have helped him evaluate this tool even faster, because he could have easily dismissed it if it appeared not to support his teaching goals. Instead, what he found was the typical, and slightly cryptic “nuts and bolts” documentation often found with software.

Step 2: Review the example thesaurus¹

Brent realizes that in order for him to come up with a realistic term to add to the example thesaurus, he needs to become familiar with its content and structure. (In the case of the student project, his students would select a topic for their thesaurus and make decisions about which terms to include relative to that topic.) After a quick look he discovers the example thesaurus domain is “amateur astronomy.” He reads the integrated documentation about the domain and potential users of this thesaurus², and likes the fact that this is the first thing you see when you open the example. He also likes that example terms in the documentation are linked to the actual terms in the thesaurus. In addition to reading about the functionality available, the links make it easy to jump to a term and actually “play” with the functionality to better understand the features of a thesaurus.

Step 3: Decide on a term to add

Since the topic of the example is amateur astronomy, Brent decides to invent a new planet. Because he has a slight obsession for alpacas, he decides to name the new planet “Alpaca.” Armed with a term to add, he is ready to add this term to the example thesaurus.

Step 4: Navigate to the page

Brent realizes the TiddlyThesaurus is web-based, and locates the URL of the example in the TiddlyThesaurus documentation. According to the documentation, this should work in all commonly used web browsers like Firefox, Internet Explorer, Safari, and Opera. This is comforting to Brent, because he knows his students have a variety of different PCs and Macs. While the iSchool tends to encourage use of one or two of the main browsers, he knows that some student will try to use a different one anyway, and he’s glad that the tool will work with these.

Step 5: Log in

Since the TiddlyThesaurus supports concurrent, collaborative editing, Brent understands that he needs to log in to use the tool. With little effort, he finds the username and password in the documentation for the account to use while editing the example. When he attempts to use the account information to log in, he is slightly annoyed by the fact that the area to enter this information on the page is hidden under an “options” link. If he hadn’t read the documentation, he would have had trouble finding it. Since he knows some students aren’t

¹ This “step” could easily be broken out into a complete and separate scenario, and could point to such a scenario. This new scenario would describe the steps required to review and use a thesaurus, including navigating to the example, reviewing the documentation for that thesaurus (separate from the TiddlyThesaurus user documentation) to become familiar with the domain, searching for terms, looking at relationships between terms, etc. In this assignment, *the step is rather vague because the focus here is on the steps required to add a term.*

² This is NOT the same as the usage documentation for the TiddlyThesaurus. This documentation is specifically about the thesaurus content and users, and should be created as part of the thesaurus construction process when using the tool. It’s an introduction/preface to the example, not documentation about the tool itself!

going to read the documentation, he realizes this could be an issue. However, he does like the AutoLogin feature, which remembers the username/password entered and automatically logs you in when coming from the same browser on the same machine. Although this feature is off by default, Brent wonders if this could cause an issue for students if several of them share the same machine. While he saw a message that told him his login was successful, he also doesn't see a clear indication on the page of who is actually logged in at the moment. This could also be a source of confusion.

Step 6: Decide where the new term should go

Brent reminds himself that he wants to add "Alpaca" as a new planet in the thesaurus, so he needs to find an appropriate location to place it. He assumes that since the example thesaurus is on amateur astronomy, it will contain a list of planets visible with the naked eye or amateur telescopes. Because he has spent years teaching LIS 537, he is familiar with the terminology and he decides to view the Classified Schedule. Although it takes a few seconds to display, he likes the fact that it is seemingly generated "on the fly" from the terms that have been entered in the system. He'd much rather his students spend time thinking about how to construct a proper thesaurus by effective use of relationships between terms, than have to worry about spending much time on the "grunt work" of formatting and editing a Classified Schedule. In any case, he reviews the schedule and locates an entry for "planets" that lists "Jupiter", "Mars", "Mercury", and others as narrower terms. This seems to be the appropriate place to add the new planet!

Step 7: Add the new term

Although Brent remembers reading that there are several ways to add a term into the thesaurus, he decides to choose the easiest from where he is currently located. He also wants to try this idea of creating a term "on the fly" from an existing term. He clicks on the "planets" link to view that term. Once it is open, he goes into edit mode and adds "[Alpaca]" to the list of narrower terms on "planets". He's not thrilled that he has to type the extra brackets around each term, but he figures it is just a property of the system and shouldn't really hamper students' ability to learn. It sure would be nice to not have to enter them, as it could add up to a lot of extra keystrokes when hundreds of new terms are entered... Since he doesn't want to spend a lot of time thinking about related terms and other properties, he leaves those boxes blank. He also remembers reading from the documentation that all new terms in the system must be tagged "preferred" to have them appear in the classified schedule. While this might serve as a reminder to students that only preferred terms appear in the classified schedule, it seems like this is an extra step that isn't required. While there should be some indication of whether the term is preferred or not, it seems silly to have to type the word in as a tag, for every preferred term entered into the system. Even for a small project with about 100 preferred terms, this would be cumbersome and annoying.

Step 8: Verify that term relationships are correct

Brent decides to verify that the new term is hooked up properly. He runs the check for Broken Links and feels embarrassed and slightly annoyed with himself because he forgot to add "planets" as a broader term of "Alpaca." While forgetting to add links in both directions is a

common mistake made by students, and a good lesson to learn, Brent thinks the TiddlyThesaurus should make life a bit easier on everyone. Adding the companion links automatically would hide the problem, and might detract from learning. But perhaps the tool could ask the user when adding a new link if they would also like to add its partner link. This sort of thing would be nice for broader/narrower, related/related, use/use for, and basically all relationships where it made sense. In any case, he goes back and manually adds the missing link, reruns the verification process, and proceeds now that the problem is fixed.

Step 9: Generate notation

Because he's been dealing with thesauruses for a while, Brent knows that creating the notation scheme is one of the last steps you do when creating a thesaurus. One of the strengths of using this TiddlyThesaurus seems to be the fact that you can add a new term and then generate the notation automatically. It should make updates a lot less painful because assigning notation to terms (one by one) is not the slow, manual process it used to be. However, one of the interesting things he'd like his students to think about in the class is what sort of notation scheme to assign to terms. He can't find a way to actually change the format of the notation in the TiddlyThesaurus, and it seems to only support a simple numbering scheme that applies to the levels in the hierarchy. This is unfortunate, and he wishes there was a way to specify the syntax/format of the notation and still allow the system to generate the values automatically. However, he does note that it is possible to edit the notation on each term in the system manually. It's just that by doing so, you end up ignoring the great time saving feature of automatic notation generation.

Step 10: Save the updated thesaurus

After a quick review of the classified and alphabetical schedules, Brent feels confident that his term has been added successfully and the changes are ready to be saved. Again, he finds it somewhat annoying that the "save changes" button is hidden under an options menu. He'd encourage his students and anyone else using the TiddlyThesaurus to save frequently so as not to lose any changes, so this should be an easy feature to find and use repeatedly.

Step 11: Produce output (ie. print)

Since the LIST 537 thesaurus construction assignment he has been giving his students has required a printed copy of the thesaurus to be turned in, Brent decides to print this example thesaurus and inspect it. The output appears clean, and clear. He does note that there is no obvious way to automatically show student names, the course number, and date on the printout. He also thinks it would be useful to be able to produce a Word document so he could use the commenting feature in that application and return the graded assignment to students.