Computational Thinking
(for the "Non-Technical")

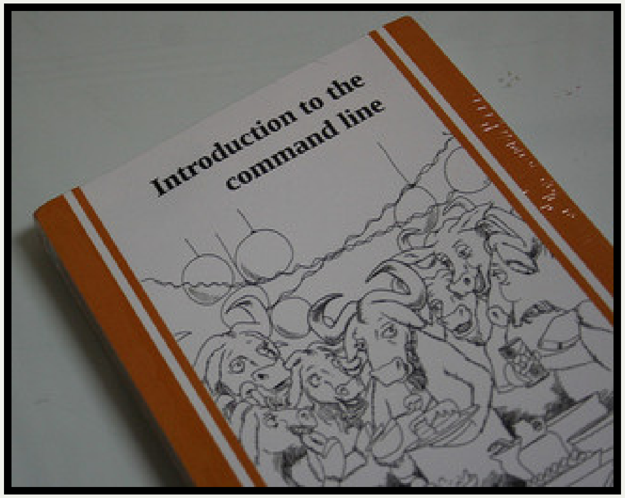
OLA Super Conference 2014

Alan Harnum, Toronto Public Library

goo.gl/TdU3A6 (slides)

twitter.com/waharnum (tweets)

Introductions



(http://www.flickr.com/photos/osama_khalid/3801230523/)

Introductions

Who I Am (1)

"Developer, librarian, various other hats as needed. I like libraries, open platforms, cloud services and doing good stuff with technology at minimal cost. I'm interested in agile (the development methodology and the general concept of agility), organizational technology cultures, and rethinking how libraries use and approach technology to serve their patrons and manage their internal processes."

(I took this from my LinkedIn profile)

Who I Am (2)

I've worked in:

- Public service
- Web development and online services
- Collection development

What I'm Hoping to Achieve

I want you to go away from this talk with enough interest in the concept of computational thinking that you'll think about it after leaving the room, and maybe even apply it to your work.

(I am also hoping not to bore you)

Who Are You?

I don't know, so I'm going to ask. A show of hands for:

- I do "techical" library work, primarily
- I do "non-technical" library work, primarily
- I'm a manager or administrator in a library
- I'm something else entirely

Who Is This For?

(http://www.flickr.com/photos/orkomedix/3675825944)

The "non-technical" myth (1)

An increasingly small amount of library work (and work in general, in 21st-century North America) is unmediated, untouched, unencumbered or unfrustrated by technology.

Who Is This For?

The "non-technical" myth (2)

I consider the "technical" / "non-technical" divide to be a pernicious one that prevents us from having effective conversations professionally or in our workplaces about the influence, role and values of technology in libraries.

(I use "values" here in both possible meanings)

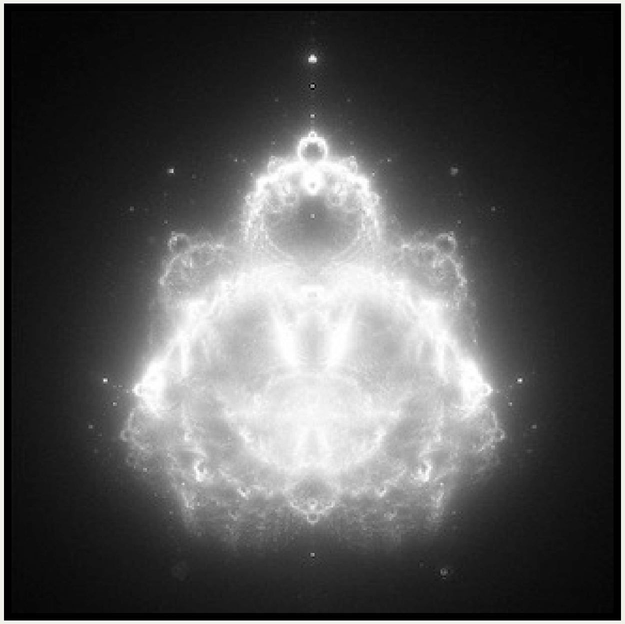
Who Is This For?

I'm pitching this talk at:

- The "non-technical" library worker
- "Non-technical" library administrators or managers

(Please stay even if you don't fit these two categories, though!)

What Is Computational Thinking?



(http://www.flickr.com/photos/generated/416849/)

Definitions

Jeanette M. Wing:

"Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent."

(http://www.cs.cmu.edu/link/research-notebook-computational-thinkingwhat-and-why)

Definitions

Jonathan Rochkind:

"I think the â€˜computational thinkingâ€™ approach is exactly how to think of what non-programmer librarians will want to usefully learn, not to be programmers, but to be able to plan out the role of software in libraries (and software is involved in nearly every aspect of a library these days), look forward and conceive of how technology might help us even more in the future, and generally control their own destinies when dealing with vendors and programmers (internal and external)."

(http://bibwild.wordpress.com/2012/11/27/computational-thinking-getting-started/)

Definitions

Alan Harnum:

> *"Something many of us in the library world already have an inchoate ability to do, but think we do not due to the conflation of computation with technology."*

How Do We Think Computationally?

(http://www.flickr.com/photos/pschadler/4932737690/)

But I Don't Want to Think Like a Computer!

Jeanette M. Wing:

> *"Computational thinking is a way humans solve problems; it is not trying to get humans to think like computers. Computers are dull and boring; humans are clever and imaginative. We humans make computers exciting. Equipped with computing devices, we use our cleverness to tackle problems we would not dare take on before the age of computing and build systems with functionality limited only by our imaginations."*

(http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf)

How Do We Think Computationally?

Some Computational Thinking Techniques

- Decomposition
- Pattern recognition
- Pattern generalization and abstraction
- Algorithm design

(http://www.google.com/edu/computational-thinking/what-is-ct.html)

How Do We Think Computationally?

Decomposition

"The ability to break down a task {or problem} into minute details so that we can clearly explain a process to another person or to a computer, or even to just write notes for ourselves."

How Do We Think Computationally?

Pattern Recognition

"The ability to notice similarities or common differences that will help us make predictions or lead us to shortcuts."

How Do We Think Computationally?

Pattern Generalization and Abstraction

"The ability to filter out information that is not necessary to solve a certain type of problem and generalize the information that is necessary."

How Do We Think Computationally?

Algorithm Design

"The ability to develop a step-by-step strategy for solving a problem."

Libraries and Computational Thinking

(still from "Desk Set", which I know everyone and their dog and their dog's dog uses in presentations about libraries and computers)

Cataloguing Standards (1)

Decomposition: we've got a lot of books and we want people to be able to find the relevant ones more quickly

Pattern Recognition: each book is a different physical object, but all have common characteristics.

Cataloguing Standards (2)

Pattern Generalization and Abstraction: what matters and does not matter among the common characteristics of books for the specific problem we are trying to solve?

Algorithm Design: here is how to record and interpret the common characteristics of a book in a systematic, replicable way to support organization and discovery.

Reference Interviews (1)

Decomposition: people seldom ask directly for the information they actually want

Pattern Recognition: there are discernable common aspects to the interaction when someone approaches the reference desk, asks a question, and receives a response

Reference Interviews (2)

Pattern Generalization and Abstraction: what common characteristics exist in both successful and unsuccessful reference transactions?

Algorithm Design: here is a process of iterative questioning and listening to improve the success of reference transactions.

You Already Know How

Computational thinking encompasses a set of natural problem-solving technique used by humans in many different contexts.

You Don't Need a Computer

Jeanette M. Wing:

"Computer science is not computer programming. Thinking like a computer scientist means more than being able to program a computer. It requires thinking at multiple levels of abstraction."

(http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf)

Let's All Get Excited!

(about computational thinking!)

(http://www.flickr.com/photos/abbyladybug/2424168642/)

Why? (1)

This won't solve all our challenges.

(http://www.flickr.com/photos/newt42/3713370145/)

Why? (2)

Just some of the challenges computational thinking can't solve:

- Shrinking budgets
- Challenging patrons
- Customer service issues
- Labour relations

Why? (3)

Where computational thinking can help us:

- Understanding which parts of a problem are amenable to computation (and thus to automation with technology, and the complexity of that automation)
- Evaluating the match between computational tools and techniques and a problem
- Understanding the limitations and power of computational tools and techniques
- Applying or adapting a computational tool or technique to a new use

(adapted from Wing, http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why)

Coda

The power and limits of computation

Windows
A fatal exception 0E has occurred at 0028:C0011E36 in VXD VMM(01) +
00010E36. The current application will be terminated.
* Press any key to terminate the current application.
* Press CTRL+ALT+DEL again to restart your computer. You will
lose any unsaved information in all applications.
Press any key to continue

(http://vimeo.com/2838688)

Coda

Ursula Franklin

″Today′s real world of technology is characterized by the dominance of prescriptive technologies... While we should not forget that these prescriptive technologies are often exceedingly effective and efficient, they come with an enormous social mortgage. The mortgage means that we live in a culture of compliance, that we are ever more conditioned to accept orthodoxy as normal, and to accept that there is only one way of doing ′it.′″

(http://books.google.ca/books?id=FaaR5hA-U-0C)

A personal view (1)

There are lots of problems with how libraries use technology

Part of the problem is the conversation

We alternately fetishize or mistrust technology

And many of us (even those of us who work daily with technology!) feel alienated from our work by it, often with justification

A personal view (2)

The excitement I have about computational thinking is my sense that it could help change the conversation

We need to talk more effectively across the "technical" / "non-technical" divide

And we need to be universally better at questioning the ability of automation and technology to solve our problems, while acknowledging their immense power

Coda

Thanks / Questions / Discussions

(http://www.flickr.com/photos/14829735@N00/202872717)