Design Tools

Jena Cummiskey and Jesse Qin

Ability-Based Design: Concept, Principles and Examples

J. O. Wobbrock, S. K. Kane, K. Z. Gajos, S. Harada and J. Froehlich

Ability-Based Design

Ability-based design focuses on ability

The current approach to accessible computing centralizes disability rather than ability

(Edwards 1995).

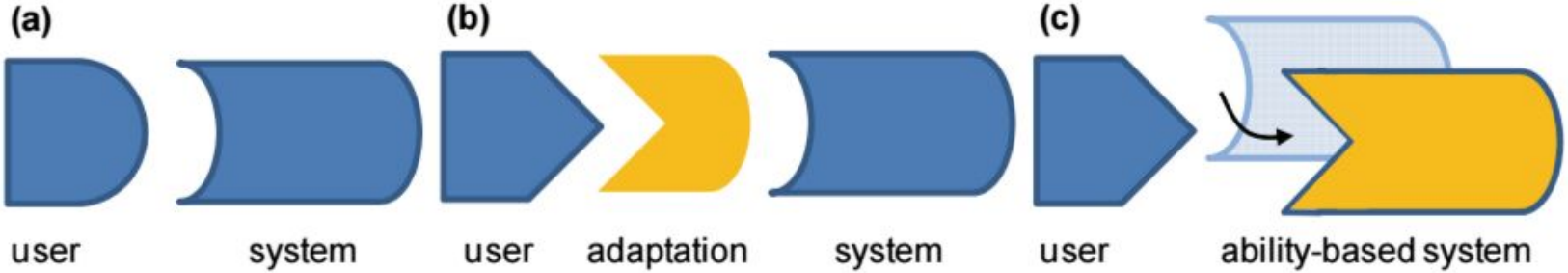
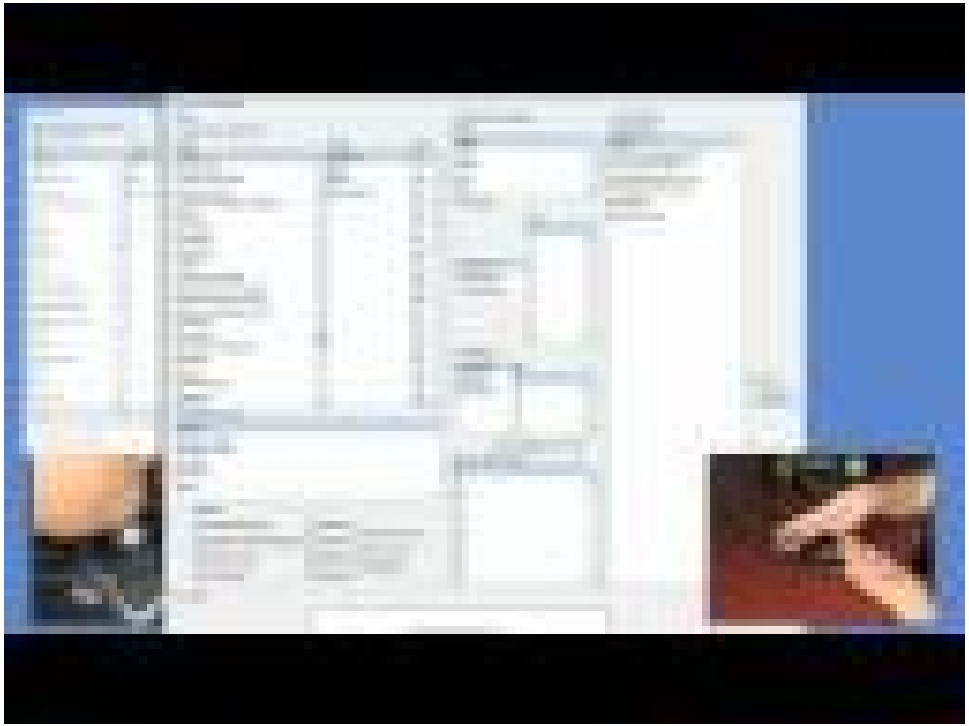


Figure 2. (a) A user whose abilities match those presumed by the system. **(b)** A user whose abilities do not match those presumed by the system. Because the system is inflexible, the user must be adapted to it. **(c)** An ability-based system is designed to accommodate the user's abilities. It may adapt or be adapted to them. Our symbols are based on those from prior work (Edwards 1995).

Adaptability and/or Adaptivity

- **Adaptivity:** the degree to which software can change itself in response to user behaviour
- **Adaptability:** the degree to which software can be customized by a user, therapist, or caregiver

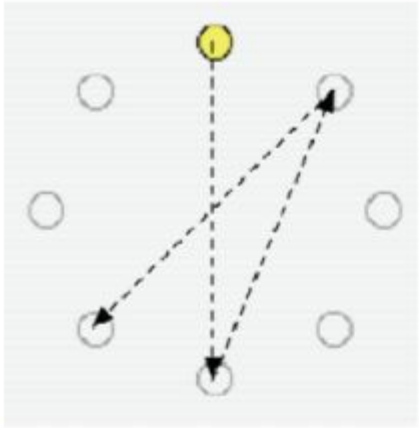


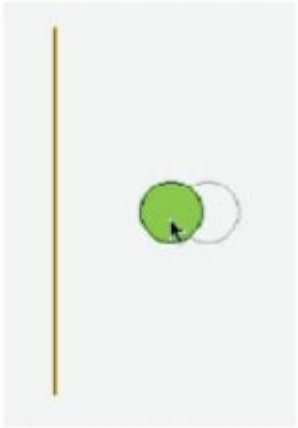
https://youtu.be/B63whNtp4qc?t=1m34s

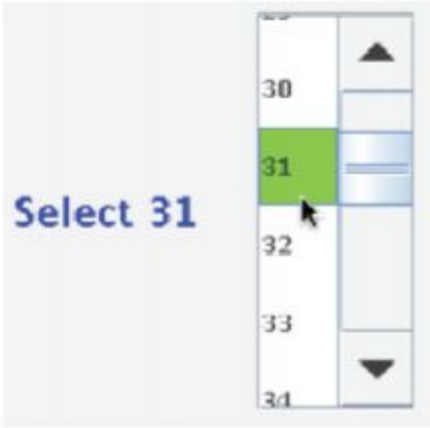
“In the ideal case, an ability-based system should be flexible enough to enable people to use a system without requiring them to alter their bodies, knowledge, or behavior.”

SUPPLE

How SUPPLE works

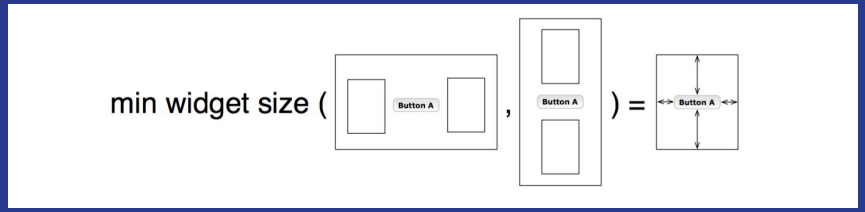






How Supple optimizes for expected speed of use

An optimization framework is used to generate user interfaces that are optimized for a user's *performance*, using a predictive model of how fast a person can perform basic user interface operations such as pointing, dragging, list selections and performing multiple clicks.



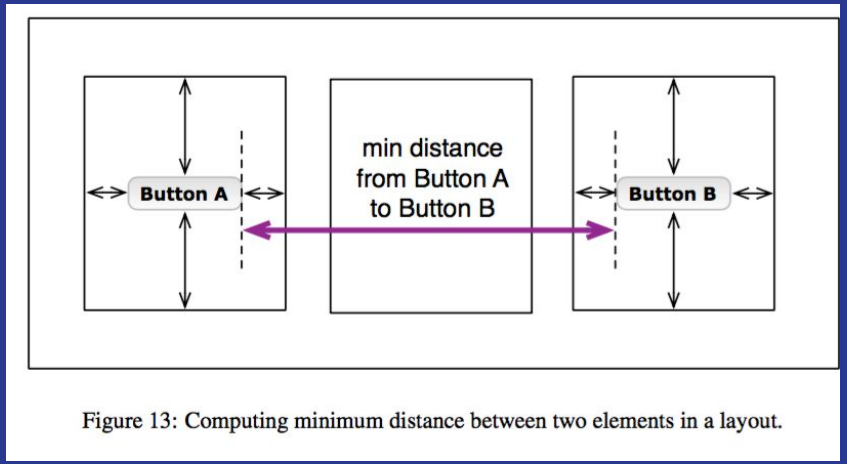


Figure 13: Computing minimum distance between two elements in a layout.

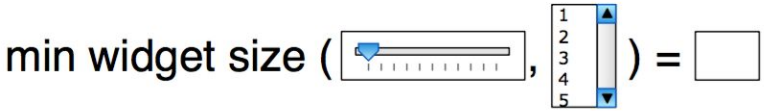
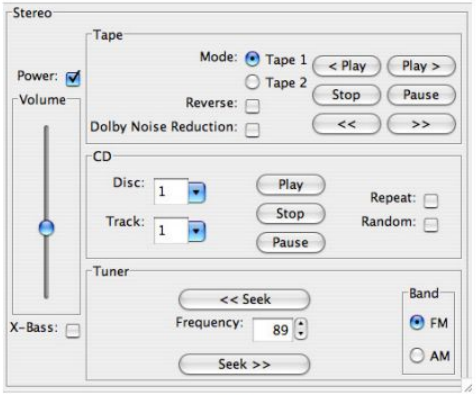
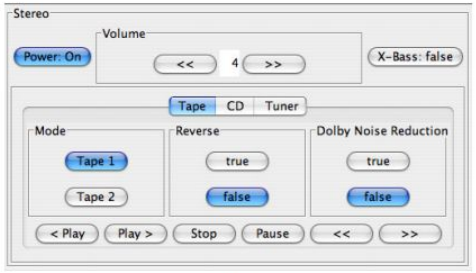


Figure 11: Computing minimum of widget sizes for primitive widgets.

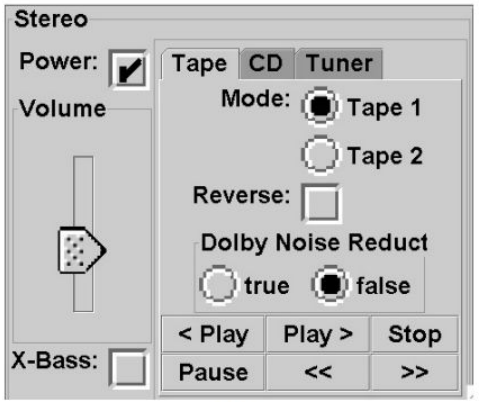
SUPPLE: GUI Variants



(a)



(b)



(c)

Adaptability and/or Adaptivity

- **Adaptivity:** the degree to which software can change itself in response to user behaviour
- **Adaptability:** the degree to which software can be customized by a user, therapist, or caregiver

Discussion:

We have outlined the adaptivity of SUPPLE, but how does SUPPLE provide Adaptability? Additionally, are there any trade-offs to the amount of adaptability an interface provides? (groups of 2-3, 2 min)

How does SUPPLE address the challenges of Ability-Based Design?

Identifies abilities **in a reliable fashion**

Takes advantage of users' abilities to interact **with available resources**

Seven Principles of Ability-Based Design

STANCE	**1. Ability.**	Designers will focus on ability not *dis*-ability, striving to leverage all that users *can* do.	*Required*
	2. Accountability.	Designers will respond to poor performance by changing systems, not users, leaving users as they are.	*Required*
INTERFACE	**3. Adaptation.**	Interfaces may be self-adaptive or user-adaptable to provide the best possible match to users' abilities.	*Recommended*
	4. Transparency.	Interfaces may give users awareness of adaptations and the means to inspect, override, discard, revert, store, retrieve, preview, and test those adaptations.	*Recommended*
SYSTEM	**5. Performance.**	Systems may regard users' performance, and may monitor, measure, model, or predict that performance.	*Recommended*
	6. Context.	Systems may proactively sense context and anticipate its effects on users' abilities.	*Recommended*
	7. Commodity.	Systems may comprise low-cost, inexpensive, readily available commodity hardware and software.	*Encouraged*

Table 1. Seven principles of ability-based design. Principles are divided into three categories: STANCE, INTERFACE, and SYSTEM. All ability-based designs will uphold principles 1-2. The next four principles (3-6) are recommended, but are not required in every ability-based design. Finally, designers are encouraged to consider upholding the seventh principle, but with the understanding that custom, specialized solutions are often warranted, especially for severely disabled users.

SUPPLE

Adapts **interfaces** to users based on their **performance** in a test battery.

Gives users the power to inspect and override its choices, resulting in good **transparency**.

Allows **commodity** pointing devices to perform faster and more accurately.

28% faster
73% more accurate

DESKTO	2008b)	with motor impairments	performance, commodity
	5.5 Trackball EdgeWrite (Wobbrock and Myers 2006a, Wobbrock and Myers 2006c, Wobbrock and Myers 2007)	gestural text entry from trackballs for people with spinal cord injuries	ability, accountability, adaptation, transparency, performance, commodity
MOUSE POINTING	**5.6 Steady Clicks** (Trewin *et al.* 2006)	mouse pointing for users with motor impairments	ability, accountability, performance, commodity
	5.7 Angle Mouse (Wobbrock *et al.* 2009)	mouse pointing for users with motor impairments	ability, accountability, adaptation, transparency, performance, commodity
	5.8 SUPPLE (Gajos *et al.* 2007, Gajos *et al.* 2008, Gajos *et al.* 2010)	mouse pointing for users with motor impairments	ability, accountability, adaptation, transparency, performance, commodity
	5.9 Automatic mouse pointing assessment (Hurst *et al.* 2007, Hurst *et al.* 2008a)	automatic assessment of mouse pointing performance	ability, accountability, adaptation, performance, commodity
	5.10 VoiceDraw (Harada *et al.* 2007)	paintbrush control for users with motor impairments	ability, accountability, adaptation, performance, commodity
ICES	**5.11 Barrier pointing** (Froehlich *et al.* 2007)	stylus pointing for users with motor impairments	ability, accountability, performance, commodity

Explain how SUPPLE works to improve the user experience for individuals with motor impairments (irregular movements and muscle spasms) associated with mild Cerebral palsy.

“SUPPLE will be able to sense the sudden accelerated mouse movement in case of an involuntary movement. For this it will slow down the mouse speed and may space out components on the screen far from each other to allow room for movement.” **Pallavi Agarwal**

https://youtu.be/B63whNtp4qc?t=2m10s

SUPPLE

Implications of User reactions to Supple

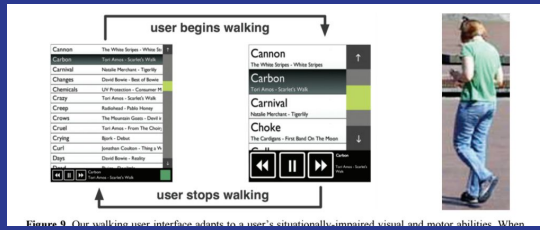
"This type of technology is monumental because it solves a technical problem as well as a psychological problem. Having technology that changes the mentality of those with disabilities from victimization to empowerment changes its role from merely a utility to being a driving positive force in society." **Jeremy Blackstone**

In testing SUPPLE researchers "saw genuine delight by many of our users when they could use "normal" devices with new software tailored to their abilities"

Commentaries

Sensing Context

“A person’s ability is not determined solely by his or her health, but also by the current environment or context (Newell 1995, Sears et al. 2003).”



In groups of 2-3:
Individually think of a situation where your abilities are impacted by environment or context. Now ask your group to consider how they would identify context and ability, in a reliable fashion, for that situation. Consider possible Ability-based design solutions, if time allows. (5 minutes)

What is required to make the universal application of ‘design-for-one’ a reality?

1. Research methods to automatically detecting, assessing, and understanding people’s abilities
2. Examine how to make devices more aware and/or responsive to environmental factors on the go
3. Considering how commodity input devices can be repurposed in novel ways for people with disabilities.

All users have abilities, and to the extent that context or impairment affect those abilities, ability-based design should be relevant.

Example-Centric Programming: Integrating Web Search into the Development Environment

Joel Brandt, Mira Dontcheva, Marcos Weskamp, Scott R. Klemmer

Key Learnings

- Low-Level

  1. Task-specific search interfaces greatly improve the search space between not knowing what you're doing at all and knowing exactly what you're doing

- High-Level

  2. We can utilize powerful resources for research if we wrap them creatively

  3. Combining small-scale and large-scale studies is an excellent technique for gaining holistic insight

Discussion - Groups of 3 (2 mins)

When you are doing a search, what kinds of content are you looking for if it's a topic/concept/implementation you:

1. Know nothing about
2. Know about but have forgotten the details

Which of the above two do you search for the most?

Research

Inspiration - Developers are on the web A LOT

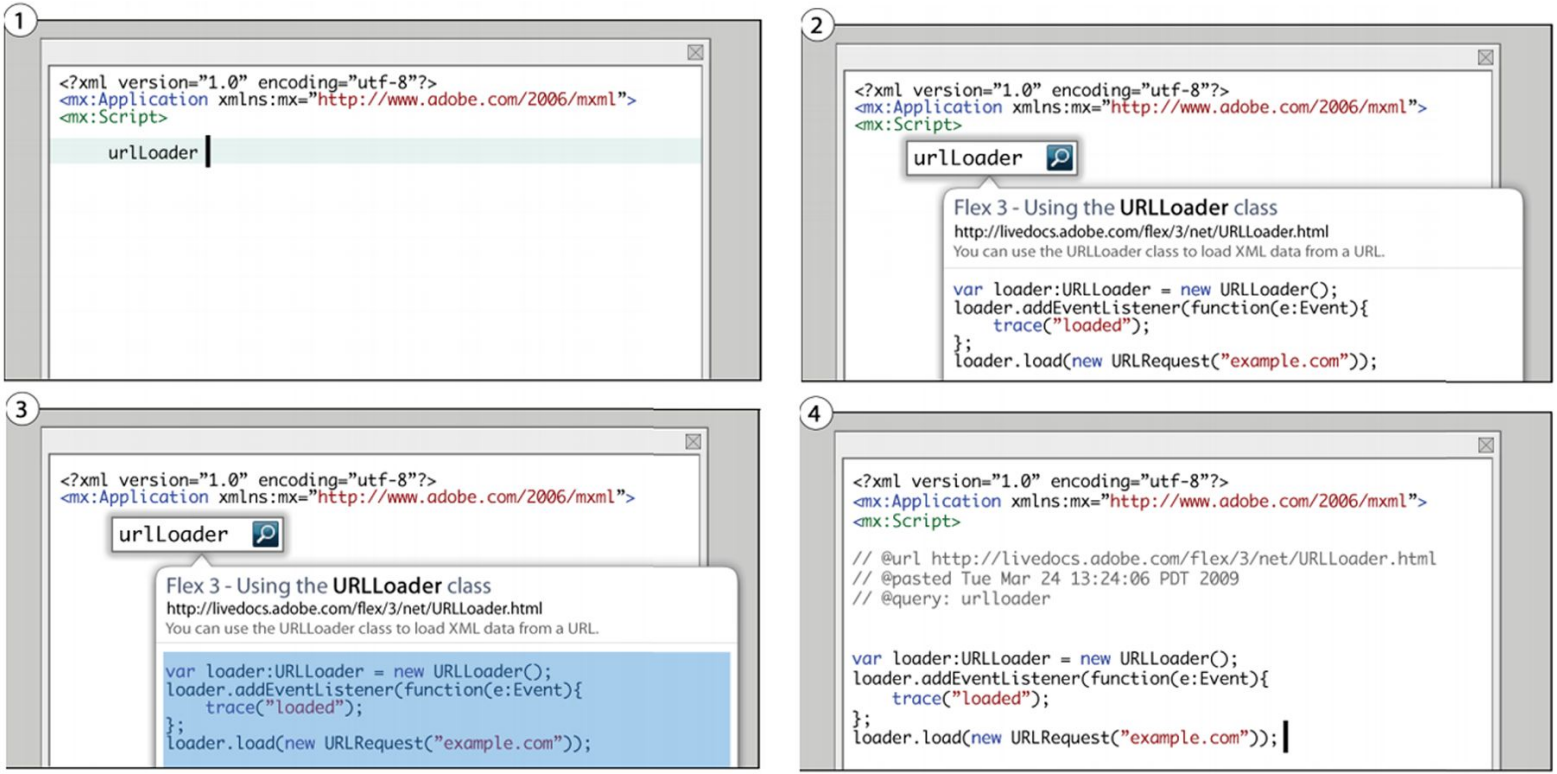
1. A web browser that is independent of other tools in the development chain
2. A search engine that has no notion of a user’s development context
3. A code editor that assumes all code is typed by hand

Blueprint

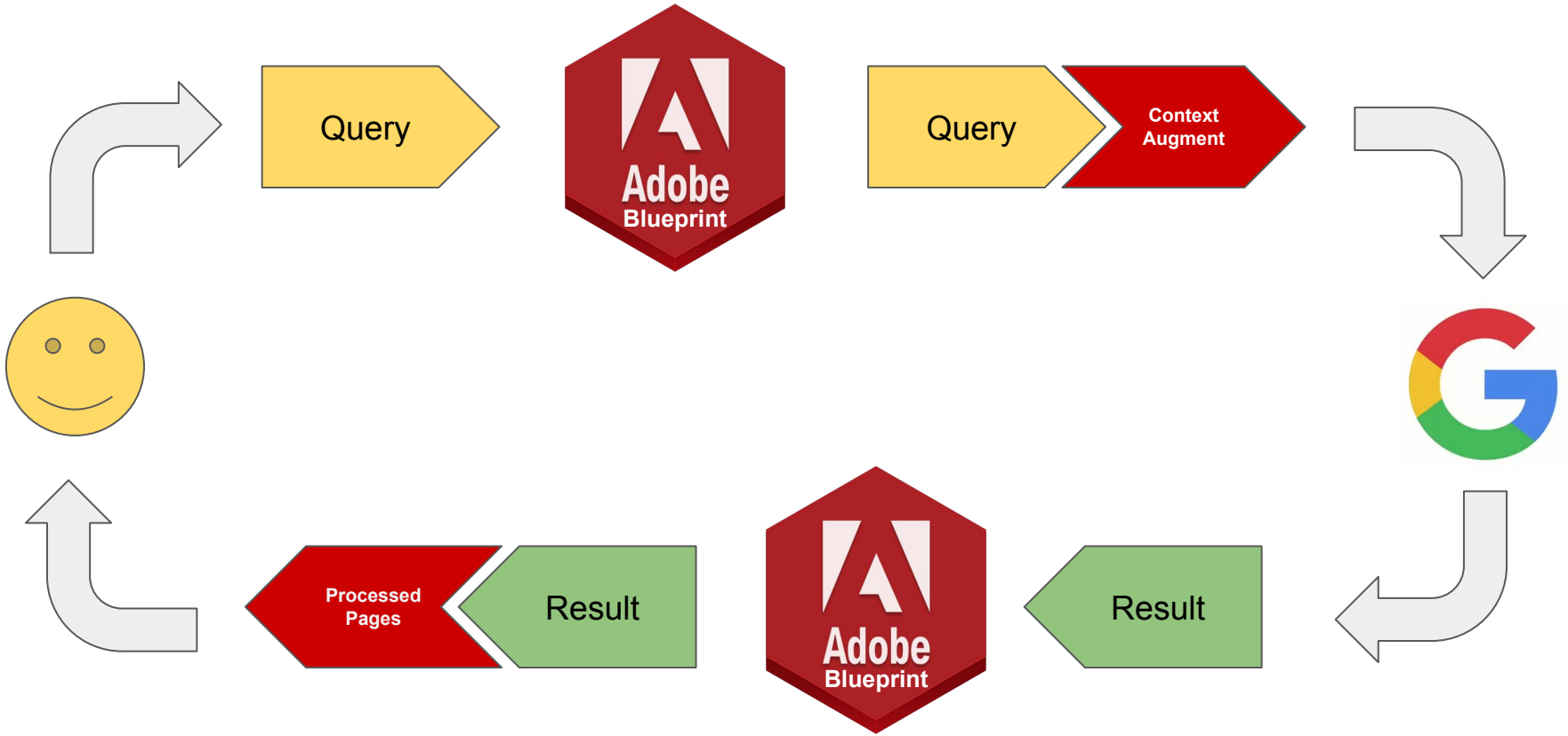
A web search interface integrated into Adobe Flex Builder development environment that helps users locate example code

1. Provides a UI for search queries and results
2. Sends contextual information
3. Focused on getting code examples

Blueprint


1
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
<mx:Script>
urlLoader
2
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
<mx:Script>
urlLoader
Flex 3 - Using the URLLoader class
http://livedocs.adobe.com/flex/3/net/URLLoader.html
You can use the URLLoader class to load XML data from a URL.
var loader:URLLoader = new URLLoader();
loader.addEventListener(function(e:Event){
trace("loaded");
};
loader.load(new URLRequest("example.com"));
3
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
urlLoader
Flex 3 - Using the URLLoader class
http://livedocs.adobe.com/flex/3/net/URLLoader.html
You can use the URLLoader class to load XML data from a URL.
var loader:URLLoader = new URLLoader();
loader.addEventListener(function(e:Event){
trace("loaded");
};
loader.load(new URLRequest("example.com"));
4
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
<mx:Script>
// @url http://livedocs.adobe.com/flex/3/net/URLLoader.html
// @pasted Tue Mar 24 13:24:06 PDT 2009
// @query: urlloader
var loader:URLLoader = new URLLoader();
loader.addEventListener(function(e:Event){
trace("loaded");
};
loader.load(new URLRequest("example.com"));

Blueprint - Implementation



Blueprint - Implementation

Advantages to leveraging commercial search engine

1. More resource-efficient to implement
2. High-quality results from natural-language queries
3. General-purpose search engines do surface code examples

Blueprint - Key Learning

2. We can utilize powerful resources for research if we wrap them creatively

A PHD student is capable of doing search research without having access to the guts of an engine like Google

Blueprint - Hypotheses

Set 1

1. Programmers will complete direct tasks more quickly because they will find example code faster
2. Code produced will have the same or higher quality as code written with traditional means
3. Programmers will produce better designs on an exploratory design task

Set 2

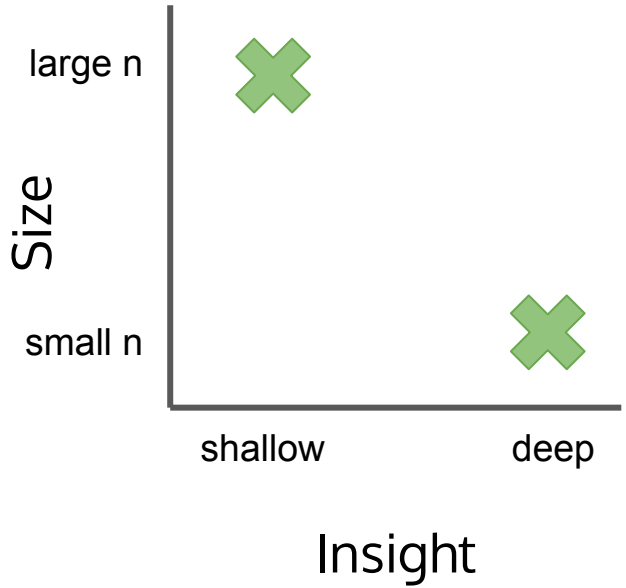
1. If additional context is not necessary, Blueprint queries should have a significantly lower click-through rate
2. If users are using Blueprint with other IDE features queries should contain more correctly formatted code
3. If Blueprint is used for reminders, users should repeat queries more frequently across sessions

Discussion - Groups of 3 (2 mins)

What are the tradeoffs between small-scale and large-scale studies?

1. Left half: advantages & disadvantages of small-scale
2. Right half: advantages & disadvantages of large-scale

Discussion - Cont.



Blueprint - Hypotheses

Set 1

1. Programmers will complete direct tasks more quickly because they will find example code faster
2. Code produced will have the same or higher quality as code written with traditional means
3. Programmers will produce better designs on an exploratory design task

Set 2

1. If additional context is not necessary, Blueprint queries should have a significantly lower click-through rate
2. If users are using Blueprint with other IDE features queries should contain more correctly formatted code
3. If Blueprint is used for reminders, users should repeat queries more frequently across sessions

Blueprint - Hypotheses

Set 1 (small n)

1. Programmers will complete direct tasks more quickly because they will find example code faster
2. Code produced with Blueprint will be the same or higher quality as code written with traditional methods
3. Programmers will produce better designs on an exploratory design task

Laboratory Study

Set 2 (large n)

1. If additional context is not necessary, Blueprint queries should have a significantly lower click-through rate
2. If users prefer Blueprint to other IDE features queries should contain more correctly formatted queries
3. If Blueprint is used for reminders, users should repeat queries more frequently across sessions

Longitudinal Study

Study 1 - Laboratory Study

Participants: 20

Control: Firefox + Adobe Community Help Search Engine

Treatment: Blueprint

Measures: Time, Rank

1. Tutorial - guided development of simple app
2. Directed Task - use a URLLoader class to retrieve text
3. Exploratory Task - make the best visualization of some data

Study 1 - Results

Set 1

1. Programmers will complete direct tasks more quickly because they will find example code faster
2. Code produced will have the same or higher quality as code written with traditional means
3. Programmers will produce better designs on an exploratory design task

1. Time to first copy/paste of example
2. Time to completion

(1) - treatment wins time (57s vs. 121s)

(2) - treatment wins time (346s vs. 479s)

Paste time strongly correlated with task completion time

Study 1 - Results

Set 1

1. Programmers will complete direct tasks more quickly because they will find example code faster
2. Code produced will have the same or higher quality as code written with traditional means
3. Programmers will produce better designs on an exploratory design task

1. Professional software engineer external to the project rank-ordered participants code

(1) - Treatment produced significantly higher-rated code

Better starting example

Study 1 - Results

Set 1

1. Programmers will complete direct tasks more quickly because they will find example code faster
2. Code produced will have the same or higher quality as code written with traditional means
3. Programmers will produce better designs on an exploratory design task

1. Professional designer external rank-ordered the participants charts

(1) - Treatment produced higher-rated designs but the result was not statistically significant

Might have been significant with a larger study

Study 2 - Longitudinal Study

Participants: 2,024 Blueprint users + 13283 Community Help users

Control: Users who used the Community Help search engine over same duration

Treatment: Blueprint users

Measures: Click-throughs, syntax, queries

Study 1 - Results

Set 2

1. If additional context is not necessary, Blueprint queries should have a significantly lower click-through rate
2. If users are using Blueprint with other IDE features queries should contain more correctly formatted code
3. If Blueprint is used for reminders, users should repeat queries more frequently across sessions

1. Number of click-throughs to source pages on search

(1) - treatment is much lower (0.38 vs. 1.32)

Scale of data: ~43,000 queries measured

Study 1 - Results

Set 2

1. If additional context is not necessary, Blueprint queries should have a significantly lower click-through rate
2. If users are using Blueprint with other IDE features queries should contain more correctly formatted code
3. If Blueprint is used for reminders, users should repeat queries more frequently across sessions

1. Check for camelCase in queries

(1) - treatment contains much more camelCase (49.6% vs. 16.2% of queries)

Blueprint searches generated directly from code. symbiotic with IDE

Study 1 - Results

Set 2

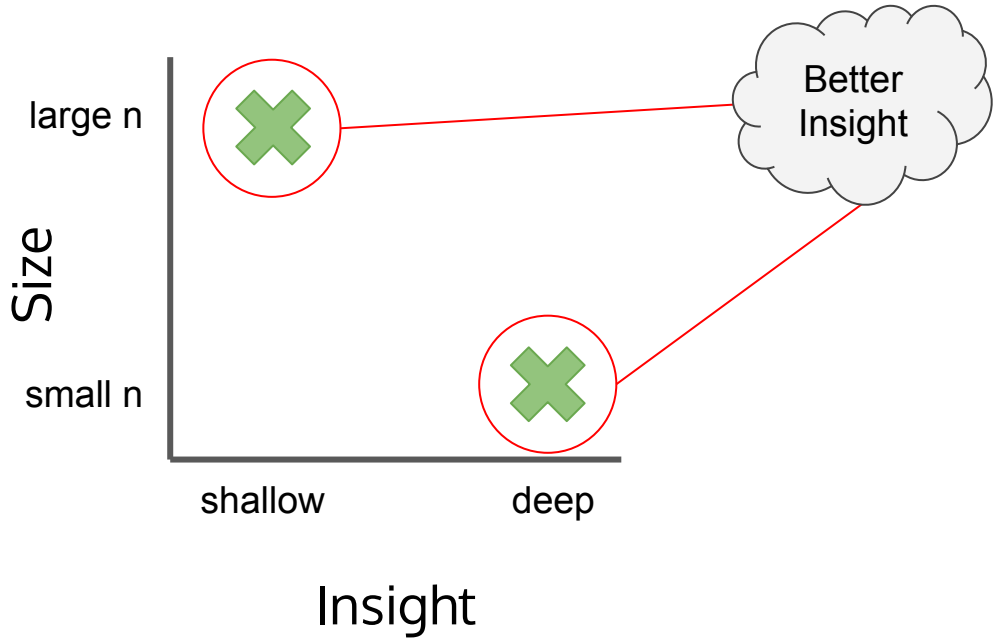
1. If additional context is not necessary, Blueprint queries should have a significantly lower click-through rate
2. If users are using Blueprint with other IDE features queries should contain more correctly formatted code
3. If Blueprint is used for reminders, users should repeat queries more frequently across sessions

1. Repeat queries during sessions

(1) - treatment has much more queries issued again by the same user (12.2% vs. 7.8% of queries)

Blueprint - Key Learning

3. Combining small-scale and large-scale studies is an excellent technique for gaining holistic insight



Blueprint - Key Learning

Active User Interviews

1. The benefits of consistent, example-centric results outweigh the drawbacks of missing context.
2. Blueprint is symbiotic with existing IDE features; they each make the other more useful.
3. Blueprint is used heavily for clarifying existing knowledge and reminding of forgotten details.

Set 2

1. If additional context is not necessary, Blueprint queries should have a significantly lower click-through rate
2. If users are using Blueprint with other IDE features queries should contain more correctly formatted code
3. If Blueprint is used for reminders, users should repeat queries more frequently across sessions

Design Space of Task-Specific Search

Task: Depth of the task is important

Expertise: Users have the expertise to evaluate result appropriateness

Time scale: More suited to make small tasks faster

Approach: More suited for directed tasks: a well-specified query

Review - Key Learnings

- Low-Level

  1. Task-specific search interfaces greatly improve the search space between not knowing what you're doing at all and knowing exactly what you're doing

- High-Level

  2. We can utilize powerful resources for research if we wrap them creatively

  3. Combining small-scale and large-scale studies is an excellent technique for gaining holistic insight

Overarching Discussion - Groups of 3 (2 mins)

How would you relate Blueprint to Ability-Based Design?

1. How are commercial search engines designed?
2. Think about the ability of a Blueprint user