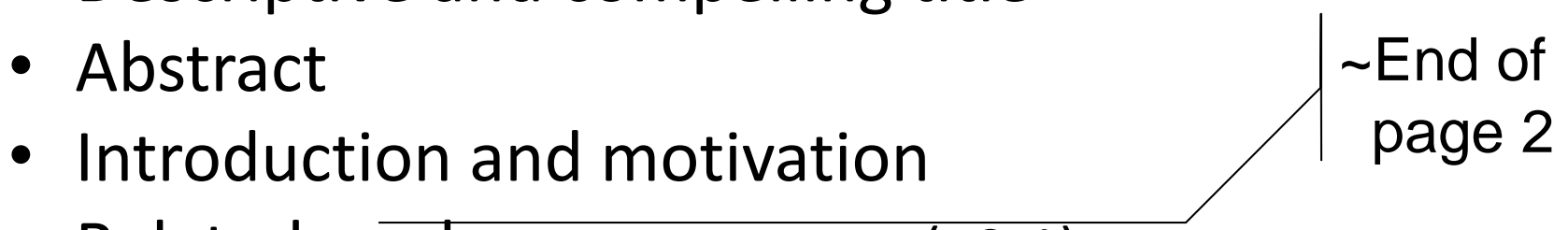


HCI paper writing patterns

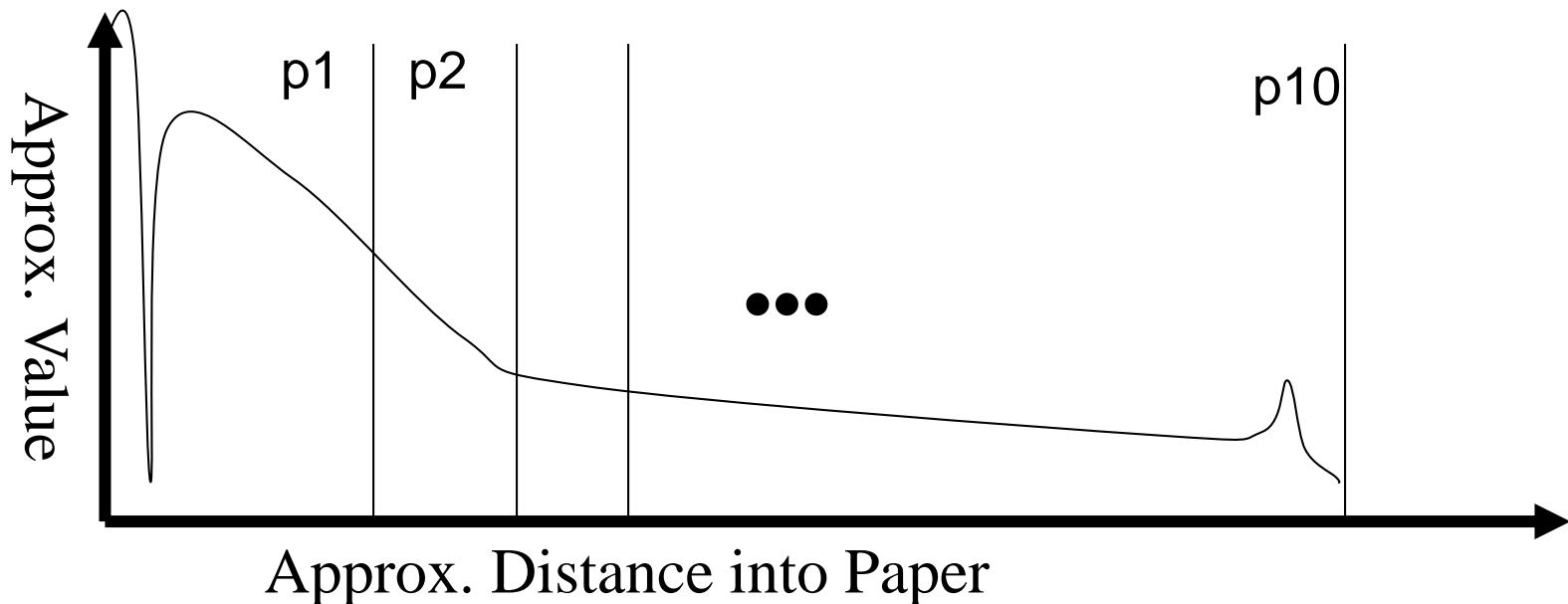
Xiaojun Bi
Department of Computer Science
Stony Brook University

Organizational template

- Descriptive and compelling title
 - Abstract
 - Introduction and motivation
 - Related work (≈0-1)
 - Technical “meat” (≈3-4 pg)
 - Validation (e.g., studies) (≈1-2)
 - Discussion and/or future work (≈1/2)
 - Conclusions (≈1/4)
 - Acknowledgements & References (no limit)
- 
- ~End of page 2

Implication: Differential value of paper real-estate

- The front of the paper is very valuable space
 - It is your only opportunity to convince most potential readers to actually read the paper



Title

- Needs to be descriptive and compelling
- Needs to represent what work is about
 - Will have to stand alone in many situations
- Tough because you don't have much space

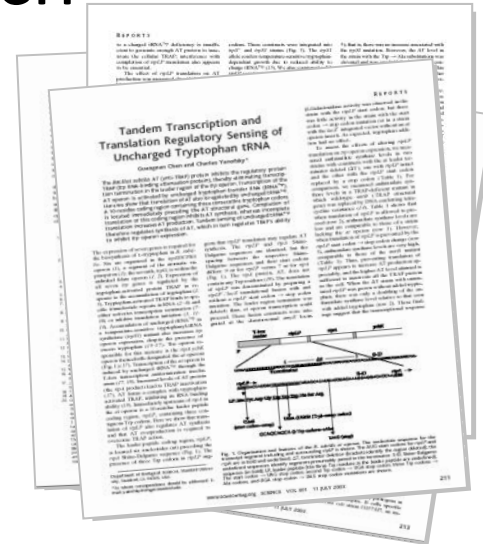
Abstract

- Has to say clearly what the paper is about and why its interesting
 - Live or die here (and intro)
 - Tell them all of what
(so that you hook their interest)
but not details of how
(not enough space)
- Schema (~2-3 sentences each?):
 - Background/Motivational setup
 - Your Approach
 - Results

How to Write an Abstract for Your Report

What are Research Abstracts?

- Stand-alone summary of the research
 - What is the problem?
 - What did you do?
 - What were your results?
 - What do the results mean?
- Abstracts are important because:
 - Some conferences ask for abstracts first
 - First thing people read in papers
 - Used in searchable databases
- **Is this research interesting to me?**



DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design

James Lin, Mark W. Newman, Jason I. Hong, James A. Landay

Group for User Interface Research, Computer Science Division

University of California, Berkeley

Berkeley, CA 94720-1776 USA

+1 510 643-3043

{jimlin, newman, jasonh, landay}@cs.berkeley.edu

ABSTRACT

Through a study of web site design practice, we observed that web site designers design sites at different levels of refinement—site map, storyboard, and individual page—and that designers sketch at all levels during the early stages of design. However, existing web design tools do not support these tasks very well. Informed by these observations, we created DENIM, a system that helps web site designers in the early stages of design. DENIM supports sketching input, allows design at different refinement levels, and unifies the levels through zooming. We performed an informal evaluation with seven professional designers and found that they reacted positively to the concept and were interested in using such a system in their work.

Keywords

Web design, Zooming User Interface (ZUI), Sketching, Informal, Pen-based Computers, Rapid Prototyping

INTRODUCTION

Web site design has much in common with other types of design, such as graphic design and “traditional” graphical

pages is depicted as a label. They then proceed to create *storyboards* of interaction sequences, which employ minimal page-level detail and focus instead on the navigational elements required to get from one page to another. Later still, designers create *schematics* and *mock-ups*, which are different representations of individual pages.

The design process often includes rapid exploration early on, with designers creating many low-fidelity sketches on paper. These sketches are considered crucial to the process. Designers can quickly sketch the overall look and feel of a web site without having to deal with unnecessary low-level details and without having to commit a large amount of time and effort to a single idea. Furthermore, sketches are important for communicating ideas with other team members and gaining valuable feedback early in the design process. These uses of sketches are similar to what has been previously reported for GUI design [12, 26].

Yet, there is a gulf between the needs of web designers during early design phases and the tools available to them. Most web design tools focus only on the creation of production web sites. The high-fidelity nature of these tools

What is the Problem?

Through a study of web site designers design sites at different levels of refinement—site map, storyboard, and individual page—and that designers sketch at all levels during the early stages of design. However, existing web design tools do not support these tasks very well.

Informed by these observations, we designed DENIM, a system that helps web site designers in their design process. DENIM supports sketching input, allows design at different refinement levels, and unifies the levels through zooming. We performed an informal evaluation with seven professional designers and found that they reacted positively to the concept and were interested in using such a system.

What did we do?

What were the results? What do they mean?

Tips on Writing Abstracts

- Write the abstract for your audience
 - Researchers in your field vs. General audience
- Abstract is stand-alone
 - Often read first to see if paper is interesting
- Length of abstracts vary
 - 1¶ for posters, tech reports, conference papers
 - 1-2¶ for journals
 - Some conferences 1-2 page abstracts first
 - Different kind of research abstract, explains what you want to do, why it's important, and any early results

Abstracts vs Introductions

- Both present an overview of the research
 - Introduction is longer, needs to have stronger, broader problem motivation
 - Introduction also provides background info, describing history of the research area
- Okay to duplicate some info in both
- Abstracts should not have:
 - References
 - Lots of background info

What are Research Abstracts?

- Abstracts should have:
 - Objective – the problem
 - Methods – how you approached the problem
 - Results – interesting results, the facts
 - Conclusions – what we think the results mean

Introduction

- Schema
 - Setup with **POTS** (next slide)
 - Background
 - Places work in research landscape
 - Try to work in very quick refs to related work here
 - Approach and/or overview of innovation
 - “The hook” (Demo of most compelling results)
 - Screen dump (or other visual) on page 2!
(page 1 if you can swing it, but that’s hard)

POTS : Promise, Obstacle, Technical Solution

- Promise
 - “Wouldn’t it be great if...”, or
“X provides a lot of potential advantages”, etc.
- Obstacle
 - “But we can’t because...”, or
“But its severely limited by...”, etc.
- Technological solution
 - “And in this paper we are going to show you how to overcome that with ...”

Introduction Schema

- Setup with POTS
- Background
- Approach and/or overview of innovation
- “The hook”
- Typical to do 1 paragraph setup and 1 to 1.5 pages for rest

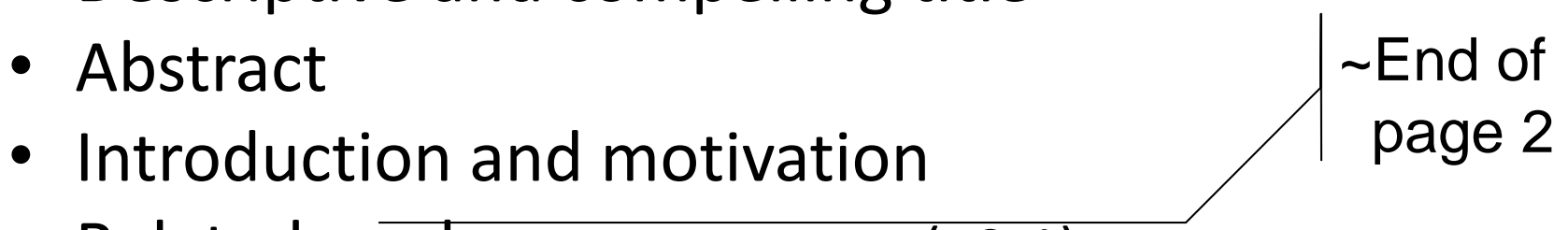
Differential value implications

- Purpose of the early parts of the paper is to entice the reader to actually read the rest
 - Have to have convinced them of this by about the end of page 2 or you are dead (2.5 at the outside)
- Need to give them early:
 - Motivation (why they should care)
 - All of “what” (what you did and what results you got)
 - At a high level (not in detail)
 - Only as much of “how” (and generally “details”) as needed make the “what” understandable

“Screen dump on page 1 or 2”

- Very important to have a visually compelling hook fairly early
 - To achieve goal, need to show at a glance that something cool is happening here
 - Lead with your best example
 - If system doesn't have a screen dump or photo, at least do an architecture diagram here

Organizational template

- Descriptive and compelling title
 - Abstract
 - Introduction and motivation
 - Related work (≈0-1)
 - Technical “meat” (≈3-4 pg)
 - Validation (e.g., studies) (≈1-2)
 - Discussion and/or future work (≈1/2)
 - Conclusions (≈1/4)
 - Acknowledgements & References (no limit)
- 
- ~End of page 2

Advice on related work

- May fold related work into background part of introduction or do it in later section
 - Very important to reference all the relevant material
 - Missing this is on the “gets you killed” list
 - Esp. missing the reviewer’s work 😊
 - But can’t spend much of this valuable space talking about somebody else’s work
 - If you need a longer discussion (usually) put it in separate section late in the paper

Technical “Meat”

- For CHI/UIST you MUST have the technical details
 - “I don’t know how you did it” or “not enough here to replicate this” is on the “gets you killed” list
 - And you had better have it right (as described, not just as implemented)
 - Goal: reviewer fully understands how it works so they can evaluate it (The devil is in the details)
- This doesn’t mean they necessarily want to hear the steps you went through to build it
- Usually don’t want code

Technical “Meat” Schema

- Overview of parts
 - How the decomposed parts (about to follow) fit together to make the whole
 - Architectural diagram if appropriate
- Part details 1
 - Now get down and dirty with the full details
- Part details 2, etc.
- [Optional] Technical wrap-up / discussion

Validation

- Good validation of invention work (particularly tools) is very hard
- Some approaches
 - Demonstrate “coverage of space”
 - examples that “span” a space (3 is typical minimum)
 - Re-implement prior system (but faster, simpler, better)
 - (Preferably real) use / experience
 - Performance testing
 - User testing
- Need for depth of validation is on a sliding scale: mature areas need a lot, new areas / highly innovative work needs less

Discussion and future work

- May have discussion in “Technical wrap-up” instead
- Future work needs to highlight the promise, but not of huge importance
 - Easy to say things here (but correspondingly they don’t carry a lot of weight)
 - Shooting yourself in the foot if reviewer thinks all the interesting stuff is here

Conclusion

- I tend to make this short
 - if its an important result/conclusion of the work, you should have summarized it up front and will only be repeating that summary here