Metadata Type System: Integrate Presentation, Data Models and Extraction to Enable Exploratory Browsing Interfaces

Yin Qu, Andruid Kerne, Nic Lupfer, Rhema Linder and Ajit Jain
Interface Ecology Lab, Texas A&M University
College Station, TX, USA
yin, andruid, nic, rhema, ajit@ecologylab.net

ABSTRACT

Exploratory browsing involves encountering new information during open-ended tasks. Disorientation and digression are problems that arise, as the user repeatedly loses context while clicking hyperlinks. To maintain context, exploratory browsing interfaces must present multiple web pages at once.

Design of exploratory browsing interfaces must address the limits of display and human working memory. Our approach is based on expandable metadata summaries. Prior semantic web exploration tools represent documents as metadata, but often depend on semantic web formats and datasets assembled in advance. They do not support dynamically encountered information from popular web sites. Optimizing presentation of metadata summaries for particular types of documents is important as a further means for reducing the cognitive load of rapidly browsing across many documents.

To address these issues, we develop a metadata type system as the basis for building exploratory browsing interfaces that maintain context. The type system leverages constructs from object-oriented programming languages. We integrate data models, extraction rules, and presentation semantics in types to operationalize type specific dynamic metadata extraction and rich presentation. Using the type system, we built the Metadata In-Context Expander (MICE) interface as a proof of concept. A study, in which students engaged in exploring prior work, showed that MICE's metadata summaries help users maintain context during exploratory browsing.

Author Keywords
Dynamic Metadata; Exploratory Browsing; Type Systems

ACM Classification Keywords
H.5.2 Information interfaces and presentation: User Interfaces: Graphical user interfaces.

INTRODUCTION

Browsing is a fundamental World Wide Web (WWW) activity [33]. According to Marchionini and Shneiderman, "browsing is an exploratory, information-seeking strategy that depends on serendipity" [32]. By *exploratory browsing*, we mean browsing when the task is open-ended and the user is unfamiliar with the information space. Exploratory browsing is key to *berrypicking* [3], the iterative process in which the user encounters new information, and her understanding and information needs evolve. Browsing and search are complementary strategies for exploring information [32]. In *exploratory search* [46], users engaged in learning and investigation iteratively refine information needs. While this paper directly addresses exploratory browsing, its implications also impact exploratory search.

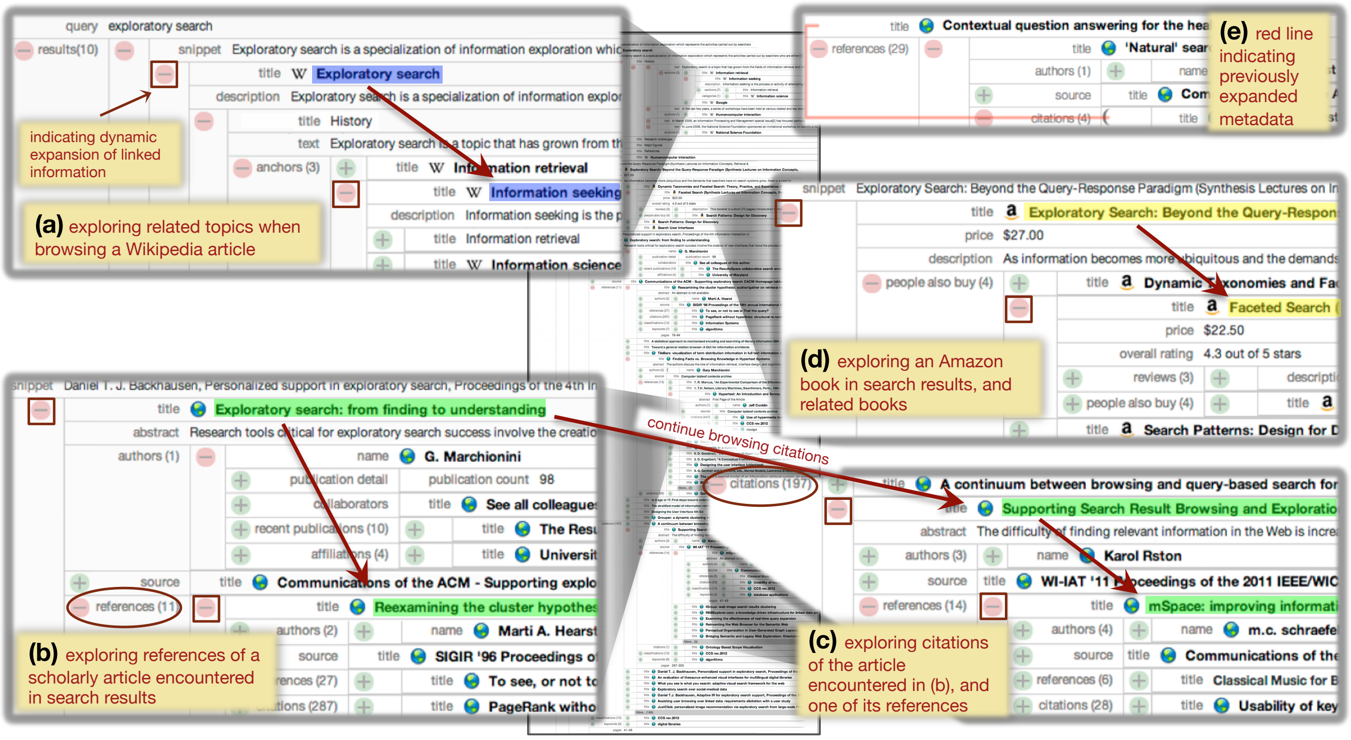
Users lose context while browsing, as new information is encountered [16]. Interlinked pages are shown in separate viewports, leading to *disorientation* [10], the problem of not knowing where you are or how to return to an encountered page in a network of information, and *digression* [17], the problem of going off track amidst many open windows or tabs. Disorientation and digression can grow acute during exploratory browsing. To counter, we design new interfaces that maintain context for the user during exploratory browsing.

Engineering exploratory browsing interfaces that maintain context requires building mechanisms for dynamically presenting trails [9] of linked documents. Since display and the user's cognitive resources, such as working memory [11], are limited, the interface must present documents as summaries, to reduce display space and cognitive load. Typically, summaries refer to manually or algorithmically extracted texts that reveal the chief points or substance of a document.

The present research hypothesizes that *metadata* – a data structure that describes a document – can function as a valuable form of summary. Many popular, useful web sites do not directly publish metadata. Thus, mechanisms for extracting metadata from ordinary web pages are required. Further, exploratory browsing involves encountering heterogeneous types of information, each of which is best represented with particular data models, layouts, and styles. The interface must dynamically derive and present metadata summaries particular to encountered information types at runtime, to reduce the cognitive load of browsing chains of linked information.

The present research operationalizes a metadata type system to address challenges of dynamically summarizing and presenting heterogeneous documents as expandable metadata in a unified context, to enable exploratory browsing across diverse websites. The type system integrates object-oriented mechanisms for describing: (1) *data models* of underlying

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
EICS 2014, June 17–20, 2014, Rome, Italy.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2725-1/14/06 ...$15.00.
http://dx.doi.org/10.1145/2607023.2607030


query exploratory search
results(10) snippet Exploratory search is a specialization of information exploration whic
title Exploratory search
description Exploratory search is a specialization of information explor
title History
text Exploratory search is a topic that has grown from th
anchors (3) title Information retrieval
title Information seeking
description Information seeking is the p
title Information retrieval
title Information science

title Contextual question answering for the hea
references (29) title 'Natural' sear
authors (1) name
source title Com
citations (4) title

snippet Exploratory Search: Beyond the Query-Response Paradigm (Synthesis Lectures on In
title Exploratory Search: Beyond the Query-Respon
price $27.00
description As information becomes more ubiquitous and the demands
people also buy (4) title Dynamic Taxonomies and Fa
title Faceted Search (
price $22.50
overall rating 4.3 out of 5 stars
reviews (3) description
people also buy (4) title
title Search Patterns: Design for D

snippet Daniel T. J. Backhausen, Personalized support in exploratory search, Proceedings of the 4th In
title Exploratory search: from finding to understanding
abstract Research tools critical for exploratory search success involve the creatio
authors (1) name G. Marchionini
publication detail publication count 98
collaborators title See all colleagues
recent publications (10) title The Resu
affiliations (4) title Universit
source title Communications of the ACM - Supporting explo
references (11) title Reexamining the cluster hypothes
authors (2) name Marti A. Hearst
source title SIGIR '96 Proceedings o
references (27) title To see, or not to
citations (287) title PageRank witho

citations (197)

title A continuum between browsing and query-based search for
title Supporting Search Result Browsing and Exploratio
abstract The difficulty of finding relevant information in the Web is increa
authors (3) name Karol Rston
source title WI-IAT '11 Proceedings of the 2011 IEEE/WIC
references (14) title mSpace: improving informati
authors (4) name m.c. schraefel
source title Communications of the
references (6) title Classical Music for B
citations (28) title Usability of key

indicating dynamic expansion of linked information
(a) exploring related topics when browsing a Wikipedia article
(e) red line indicating previously expanded metadata
(d) exploring an Amazon book in search results, and related books
continue browsing citations
(b) exploring references of a scholarly article encountered in search results
(c) exploring citations of the article encountered in (b), and one of its references

Figure 1: An overview of Mia's exploratory browsing with MICE. Snippets show close-up views of her session. Arrows denote browsing linked information.

metadata schemas; (2) *extraction rules* for deriving metadata summaries from particular web pages, using appropriate data models; and (3) *presentation semantics* to guide usable display of metadata summaries. When the user explores a web page in context, the runtime dynamically assigns the optimal metadata type, instantiates typed metadata summary objects, extracts metadata from the page to populate the instances, and passes appropriate semantics to the interface, along with instances, for type-specific presentation. The interface makes relationships between linked documents visible.

Our contributions include:

1. Demonstrating, in exploratory browsing, the unique value of metadata as a form for document summaries.

2. Integrating data models with extraction rules and presentation semantics for rich, usable presentation of metadata summaries that helps mitigate cognitive load.

3. Supporting heterogeneous information types through document subtype polymorphism, which promotes reuse.

4. Supporting encountering new information as needed for exploratory browsing and berrypicking, by dynamically binding documents, types, extraction, and presentation.

5. Making metadata available from popular web sites, such as Amazon, ACM Digital Library, and search engines.

Using the metadata type system, we built an **example** dynamic exploratory browsing interface, the Metadata In-Context Expander (MICE) (Figure 1) [24]. The visual appearance of MICE looks like a typical XML or RDF visualizer, but goes beyond by supporting dynamic acquisition, presentation, and exploration of new information.

We begin this paper with an exploratory browsing scenario that motivates development of the interface and type system, followed by a discussion of related work. Next, we use the scenario to contextualize an explanation of how the metadata type system enables exploratory browsing interfaces, such as MICE. We then present a user study in which computer science students browse and explore scholarly articles for prior work search and project ideation. The results show how MICE supports exploratory browsing in context. We finish by deriving implications for engineering exploratory browsing interfaces, and drawing conclusions.

SCENARIO: EXPLORATORY BROWSING WITH MICE

Mia is a computer science student who wants to conceptualize a research project about exploratory search. She starts by searching Google for "exploratory search". MICE presents search results in an expandable list. Each result is presented as a snippet followed by a collapsed document with only the title visible. Mia notices that the title shown in MICE is clickable, like the title links Google would present. She clicks on the plus button to expand the collapsed document in the first result, which is a Wikipedia article [47]. Information from that document is accessed in real time, converted to metadata summary, and presented in MICE (Figure 1a).

The metadata summary of the Wikipedia article introduces the concept of exploratory search, including its history, research challenges, and major researchers. Related topics are linked. Some linked topics are new to Mia. With MICE, Mia can easily expand linked topics, bringing metadata about them into the current context (Figure 1a). Further related topics are again linked as expandable metadata. Using this recursive information expansion, she explores topics, such as "information seeking", "faceted search", and "information foraging". She can still see the central topic, exploratory search, at the top, which helps her maintain focus without being disoriented by Wikipedia's many links. After a period of exploration in Wikipedia, she collapses these related topics in MICE and returns to the search results.

Wikipedia provides a good overview, but Mia wants to dig deeper. From the search results, she expands a scholarly article [31] from the ACM Digital Library. MICE extracts metadata for that article, including authors, abstract, references, and citations, and presents it in a concise form (Figure 1b).

Mia expands that article's references in MICE, seeking prior work it builds upon. She sees an interesting article [20] (Figure 1b) about exploring information, in which the user starts with a broad query and uses clustering to gradually refine the scope of information to explore [12]. She finds this idea inspiring. Another reference [32] distinguishes two different strategies for finding information: search and browsing. Mia keeps chaining references, discovering a seminal survey on experiential problems in hypertext [10], e.g., the cognitive load of seeing enormous amounts of information and maintaining context. By exploring prior work, Mia expands her understanding of the roots of research in this field.

With this understanding, Mia seeks new work in this field. She goes back to a scholarly article encountered in the search results, and expands its citations (Figure 1c). MICE shows 10 citations at a time, out of 197. Mia clicks on a button at the end of the list to reveal the next 10. She expands a title that catches her eye: [35] (Figure 1c). It discusses an interesting idea of applying zoomable user interfaces to clustering-based exploration. From the references of this article, she notices another one [27] by one of the major researchers introduced in the Wikipedia article. She expands it (Figure 1c).

As Mia continues exploring, she encounters information from sources including CiteSeerX, Google Books, and Amazon (on which she orders books on exploratory search and faceted search, Figure 1d). MICE supports her exploration process by showing rich, useful metadata summaries of browsed documents, and iteratively bringing linked information into context on click. When she encounters a previously expanded document, MICE displays a red line on hover, leading her to the previous occurrence (Figure 1e). Over an extended period of non-linear exploratory browsing, Mia learns a lot about the topic, including motivation, prior work, critiques, and new directions. Synthesizing what she learns, Mia conceptualizes a project about software architecture that supports multiple paradigms of exploratory browsing and search interfaces.

RELATED WORK

The metadata type system and exploratory browsing interface relate to prior work on the Semantic Web, metadata extraction, and exploration. It differentiates from them in making metadata available from popular web sites, extracting metadata of multiple types from heterogeneous sources, and supporting exploratory browsing with new information.

Metadata on the Web

The Semantic Web [2] effort develops standards and techniques to represent, query, and process metadata. RDF [42] is the primary information model. It represents metadata as *triples*, each consisting of a subject, a predicate, and an object. RDF can describe complex, interlinked metadata and relationships. However, many useful web sites and services, e.g. Google Search, Amazon, ACM Digital Library, and Twitter, do not publish RDF. RDF-S [43] and OWL [44] are Semantic Web technologies that specify metadata schemas using RDF. The focus is on inference rather than presentation. In consideration of contemporary web programming practice, we observe that presently out of the 10604 APIs indexed by `http://programmableweb.com`, only 74 use RDF. Thus, the Semantic Web representation of types and data seems unpopular with web developers. This problem is not new [1].

Microdata [45] embeds metadata into HTML pages using attributes denoting types and properties, making it easier for websites to publish formal semantics. Major search engines have been collaborating on a set of standard semantic types described in microdata, at `http://schema.org`. However, like RDF, many useful web sites, including Amazon and the ACM Digital Library, do not publish microdata.

Extracting Metadata

To overcome the scarcity of metadata on the web, prior systems extract metadata, for the user to collect and view.

Web Summaries [14] is a browser extension that allows users to create extraction patterns, extract metadata from web pages, and see collected metadata in different views. In a 10-week study [15], extracted metadata was found useful for both transient and long-term user tasks. Users liked a functionality called that takes a hyperlink on a page, extracts metadata from the destination page, and brings it into the context of the current page. They valued directly accessing linked metadata from within an initial context.

Piggy Bank [23] extracts metadata from web pages using a browser extension, stores it in an RDF database, and provides a faceted browsing interface. Exhibit [21] allows the user to publish metadata in a special JSON-based format in a faceted interface. Presentation is templated and customizable.

Marmite [48] and Vegemite [30] let end users create browser based metadata extraction scripts, or *mashups*. However, studies showed that users without programming skills experienced difficulty in authoring mashups. Thus, the applicability of these tools to general, unskilled users is unclear.

Clui [34] provides a browser plugin for users to collect metadata from the browser, represented with types called "webits". The type system lacks inheritance or polymorphism, and so is limited.

The present metadata type system actively extracts metadata from regular web pages. Different from scrapers that extract individual metadata records from open browser windows, the type system supports extensibility by enabling developers to reuse data models and presentation semantics, through an object-oriented programming language, as well as to reuse types, such as scholarly article, across different information sources. Further, this research addresses presenting *linked* metadata in one context. while making relationships visible, to support exploratory browsing.

Exploring Metadata

mSpace is a faceted browser for exploring a fixed repository of knowledge in the form of a metadata collection [27]. The user can re-order facets (dimensions) to re-organize presentation. When the user hovers over a facet label, mSpace shows associated snippets extracted from documentsl, bringing limited information into context. mSpace requires the knowledge to be encoded in RDF in advance [39], preventing exploratory browsing of newly encountered information.

CS AKTive Space presents a UK Computer Science research metadata collection [18]. The interface displays search results in a faceted list, supporting column sorting and preview cues, like mSpace. When the user selects a research group, person, or publication, details are shown beneath the faceted list, in the same page, maintaining context. However, only one detailed item can be shown at a time. Metadata is collected through *ad hoc* programs translating data to RDF, called *mediators*. They have been used "predominantly for large, comparatively static data sources" and "high-value data sources of general interest to the community" to populate the system with enough data, implying a scarcity of RDF data in the domain. Since knowledge acquisition precedes interaction, the ability for serendipitous browsing and exploration is limited. The system only addresses information in one domain, not an open-ended set of heterogeneous sources.

PGV [13] visualizes interconnected metadata in RDF as a graph; nodes are entities and edges are relationships. The user can expands linked nodes incrementally. The Atom Interface [37] improves visual presentation of such a graph using circles. X3S [40] reconstructs RDF query results in XML, which is further transformed to HTML styled with CSS for presentation. These interfaces operate on prepared RDF datasets, and thus do not support open exploratory browsing.

Tabulator [5] is a generic browser for linked RDF data. Its outliner mode shows metadata in a manner similar to MICE. Tabulator supports serendipitous browsing, differentiating from prior RDF interfaces. It is more generic. When the user expands a field, and the field is a link to another metadata record, it actively deferences the link and shows connected metadata in the same context. The authors emphasized such serendipity, since it supports "re-use of information in ways that are unforeseen by the publisher, and often unexpected by the consumer". However, Tabulator's scope is limited by the scarcity of RDF data on the web. The absence of type-based presentation semantics leaves issues of metadata's cognitive load un-addressed.

Parallax [22] enables "set-based browsing". The user starts with a set of metadata records, connected by facets. However, when browsing across facets, direct presentation of context is lost. The user views metadata linked to the current set in a new viewport. To ameliorate, Parallax maintains a linear trail of previously browsed sets. However, browsing and exploration processes may not be linear [29] [19]. Parallax works with a prepared dataset, available at `freebase.com`. Users thus cannot explore live information outside the prepared dataset, such as ACM Digital Library papers.

METADATA TYPE SYSTEM

Building upon the open source *meta-metadata* language and architecture [28], we develop a metadata type system to support exploratory browsing interfaces such as MICE. The type system addresses representing documents as metadata, dynamic metadata extraction, and presenting linked, heterogeneous metadata. We use the above scenario to contextualize our presentation of the metadata type system.

Figure 2 presents a procedural overview of the metadata type system. When the user encounters a document, the type system automatically selects the most appropriate type and binds it to the document. Data models and extraction rules specified by the type enable dynamic metadata derivation from the document. Then, the extracted metadata instance and the type, including presentation semantics, are sent to the interface. The interface dynamically binds data models and presentation semantics with extracted metadata and generates customized visual elements to present heterogeneous metadata in context.

Representing Documents as Metadata

Limits in human cognition form the basis of a need to concisely and consistently represent documents. In Mia's exploratory browsing session, she encounters diverse documents, such as articles, author profiles, and books. Some documents contain nested structures, such as a long list of citations. Presenting original documents with all the information in one context could overwhelm, since working memory is limited [11]. To mitigate this, the present research summarizes documents as *metadata*. Nested structures, such as citation lists, are broken down into constituent sub-objects, which the user can collapse and expand to focus use of attention and display. Figure 1b shows metadata for a scholarly article, e.g., title, authors, and references.

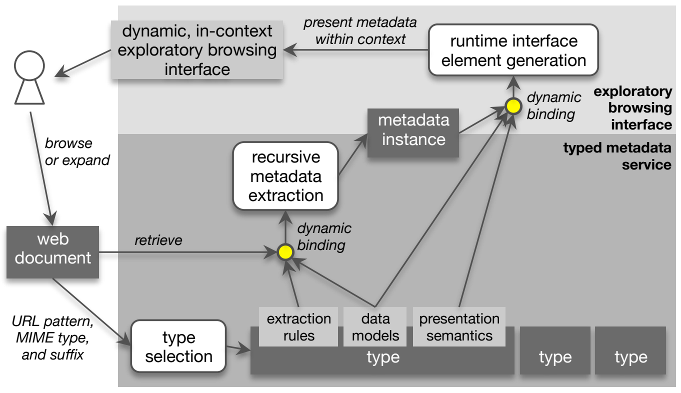

dynamic, in-context exploratory browsing interface
present metadata within context
runtime interface element generation
dynamic binding
exploratory browsing interface
metadata instance
typed metadata service
browse or expand
recursive metadata extraction
dynamic binding
web document
retrieve
URL pattern, MIME type, and suffix
type selection
extraction rules
data models
presentation semantics
type
type
type

Figure 2: Metadata type system: procedural view.

The present metadata type system specifies types in code blocks called *wrappers*. Figure 3 shows example wrappers used in Mia's scenario. Wrapper `creative_work` specifies a common type for creative work, which includes *fields* such as `year`, `references`, and `rating`. The type system supports three kinds of fields. (1) A *scalar field* defines a typed slot for scalars – values conveniently represented as a string. For example, field `year` in wrapper `creative_work` specifies a slot for an integer. (2) A *composite field*, such as `rich_media` in wrapper `creative_work`, defines a slot for an instance of a specified `type`. (3) A *collection field* defines a slot for a set of instances of a common type specified by `child_type`. In wrapper `creative_work`, field `references` specifies a reference list in which each reference must be an instance of `document` (or its subtypes by *polymorphism*, which we will explain later), and field `citations` specifies a citation list in which each citation is an instance of `creative_work`. A collection field can also hold a set of scalar values. Composite and collection fields can represent relationships between linked metadata, as `references` and `citations` do.

The type system supports *inheritance*, denoted by attribute `extends`, for reusing and extending types. For instance, as a form of creative work, we derive a type, `scholarly_article`, that inherits from `creative_work`, adding new fields such as `source` and `keywords` (Figure 3). Wrapper `acm_portal` further subtypes `scholarly_article` to extract metadata in the general scholarly article data model from a specific source (the ACM Digital Library). A common practice is to define a data model in a base type and use it for source-specific extraction in subtypes. The type system defines a common base type, `document`, for general web pages, which includes a `title`, a `location` (the URL), and a `description`.

The present metadata type system further enables representation of real world semantics involving multiple inheritance. We use *mixins* [8], which enable non-hierarchical incorporation of structures from another type without explicit inheritance to address this issue, achieving type flexibility on par to that of Freebase [6].

Extracting Heterogeneous Metadata From Documents

A major difficulty with representing documents as metadata is that many popular, useful web sites do not publish metadata. The metadata type system addresses this by actively extracting metadata of heterogeneous types from regular HTML pages published by these sites.

The extraction process begins when the user encounters a document. In Mia's case, when she clicks on the plus button to expand the encountered ACM Digital Library article, MICE makes a request to an underlying typed metadata service to extract metadata for that article. Since Mia could encounter many types of documents, the service needs to select the wrapper appropriate for the requested document. This is enabled by matching URL or mime-type features using *selectors* defined in wrappers. In Figure 3, the wrapper `acm_portal` specifies a selector for ACM Digital Library articles, with a URL pattern as the feature. Once matched, the wrapper associated with a selector is bound to the document for extraction.

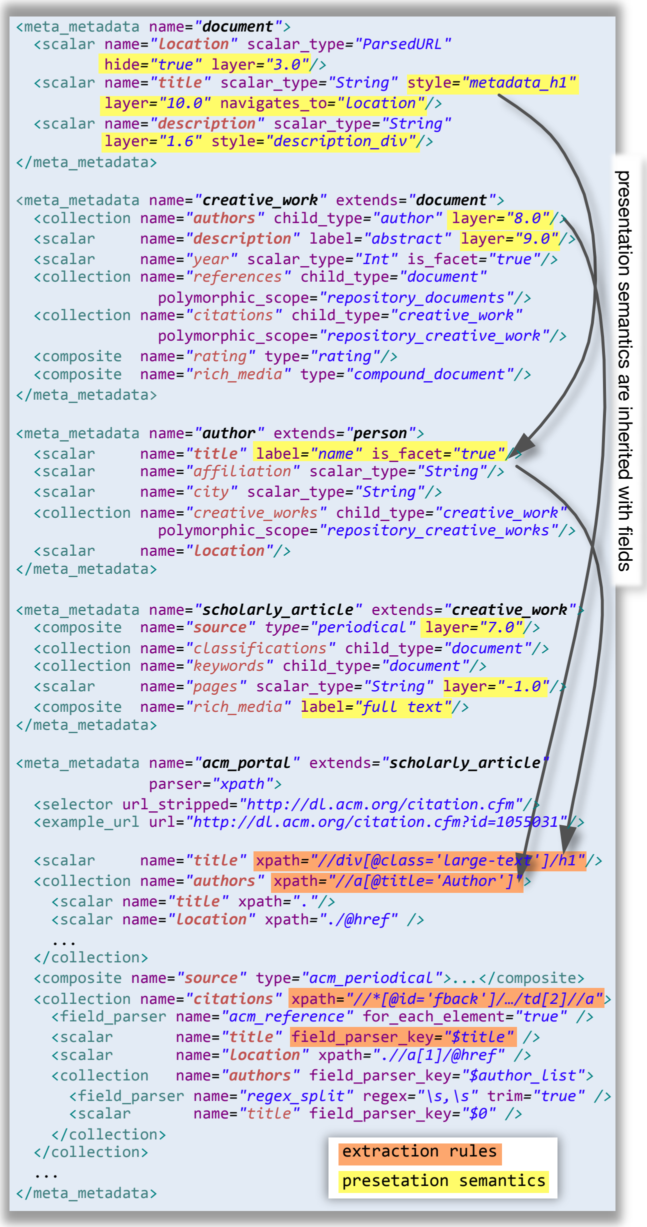
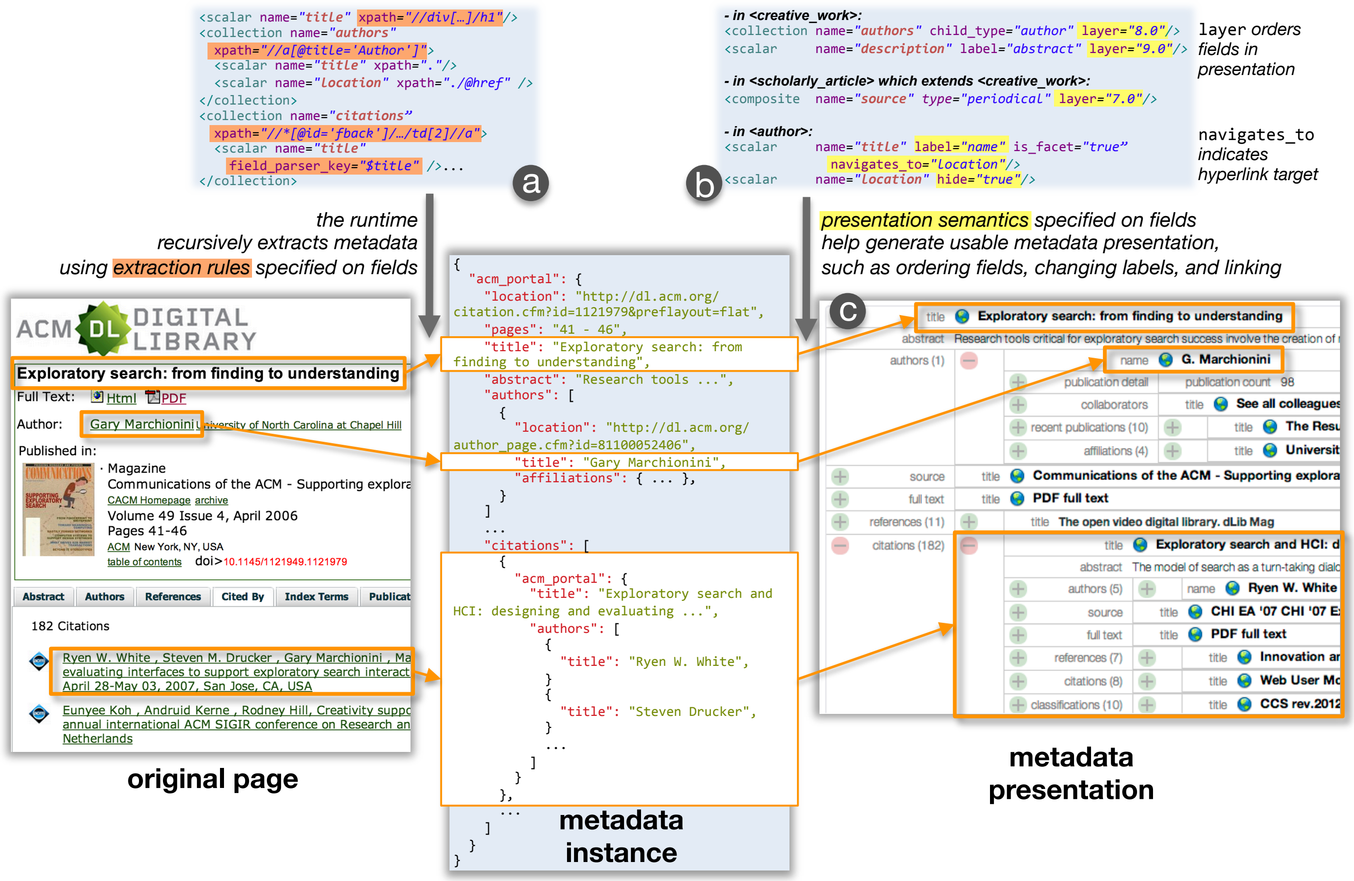

```
<meta_metadata name="document">
  <scalar name="location" scalar_type="ParsedURL"
          hide="true" layer="3.0"/>
  <scalar name="title" scalar_type="String" style="metadata_h1"
          layer="10.0" navigates_to="location"/>
  <scalar name="description" scalar_type="String"
          layer="1.6" style="description_div"/>
</meta_metadata>

<meta_metadata name="creative_work" extends="document">
  <collection name="authors" child_type="author" layer="8.0"/>
  <scalar     name="description" label="abstract" layer="9.0"/>
  <scalar     name="year" scalar_type="Int" is_facet="true"/>
  <collection name="references" child_type="document"
                polymorphic_scope="repository_documents"/>
  <collection name="citations" child_type="creative_work"
                polymorphic_scope="repository_creative_work"/>
  <composite  name="rating" type="rating"/>
  <composite  name="rich_media" type="compound_document"/>
</meta_metadata>

<meta_metadata name="author" extends="person">
  <scalar     name="title" label="name" is_facet="true"/>
  <scalar     name="affiliation" scalar_type="String"/>
  <scalar     name="city" scalar_type="String"/>
  <collection name="creative_works" child_type="creative_work"
                polymorphic_scope="repository_creative_works"/>
  <scalar     name="location"/>
</meta_metadata>

<meta_metadata name="scholarly_article" extends="creative_work">
  <composite  name="source" type="periodical" layer="7.0"/>
  <collection name="classifications" child_type="document"/>
  <collection name="keywords" child_type="document"/>
  <scalar     name="pages" scalar_type="String" layer="-1.0"/>
  <composite  name="rich_media" label="full text"/>
</meta_metadata>

<meta_metadata name="acm_portal" extends="scholarly_article"
               parser="xpath">
  <selector url_stripped="http://dl.acm.org/citation.cfm"/>
  <example_url url="http://dl.acm.org/citation.cfm?id=1055031"/>

  <scalar     name="title" xpath="//div[@class='large-text']/h1"/>
  <collection name="authors" xpath="//a[@title='Author']">
    <scalar name="title" xpath="."/>
    <scalar name="location" xpath="./@href" />
    ...
  </collection>
  <composite name="source" type="acm_periodical">...</composite>
  <collection name="citations" xpath="//*[@id='fback']/…/td[2]//a">
    <field_parser name="acm_reference" for_each_element="true" />
    <scalar       name="title" field_parser_key="$title" />
    <scalar       name="location" xpath=".//a[1]/@href" />
    <collection   name="authors" field_parser_key="$author_list">
      <field_parser name="regex_split" regex="\s,\s" trim="true" />
      <scalar       name="title" field_parser_key="$0" />
    </collection>
  </collection>
  ...
</meta_metadata>
```

Figure 3: Type inheritance and referencing in example meta-metadata wrappers from Mia's scenario.

After binding the wrapper `acm_portal` with Mia's encountered article, the runtime uses *extraction rules* integrated with data model fields to extract metadata from the document. Extraction rules can include (1) XPaths which operate on the HTML DOM tree, (2) names that directly map to elements in XML or JSON documents, (3) regular expressions for pattern matching and filtering, and (4) *field parsers* for injecting algorithms to parse strings in special formats. Figure 3 shows example extraction rules (XPaths and a field parser) for extracting article metadata from ACM Digital Library articles.

Algorithmically, the extraction process first instantiates an empty metadata object of the selected type, then populates the instantiated metadata object with extracted information



```
<scalar name="title" xpath="//div[…]/h1"/>
<collection name="authors"
  xpath="//a[@title='Author']">
  <scalar name="title" xpath="."/>
  <scalar name="location" xpath="./@href" />
</collection>
<collection name="citations”
  xpath="//*[@id='fback']/…/td[2]//a">
  <scalar name="title"
    field_parser_key="$title" />...
</collection>
```

```
- in <creative_work>:
<collection name="authors" child_type="author" layer="8.0"/>
<scalar     name="description" label="abstract" layer="9.0"/>

- in <scholarly_article> which extends <creative_work>:
<composite  name="source" type="periodical" layer="7.0"/>

- in <author>:
<scalar     name="title" label="name" is_facet="true”
              navigates_to="location"/>
<scalar     name="location" hide="true"/>
```

a
b
c
layer orders fields in presentation
navigates_to indicates hyperlink target
the runtime recursively extracts metadata using extraction rules specified on fields
presentation semantics specified on fields help generate usable metadata presentation, such as ordering fields, changing labels, and linking
original page
metadata instance
metadata presentation

ACM DL DIGITAL LIBRARY

Exploratory search: from finding to understanding

Full Text: Html PDF

Author: Gary Marchionini University of North Carolina at Chapel Hill

Published in:

· Magazine
Communications of the ACM - Supporting explora
CACM Homepage archive
Volume 49 Issue 4, April 2006
Pages 41-46
ACM New York, NY, USA
table of contents doi>10.1145/1121949.1121979

Abstract | Authors | References | Cited By | Index Terms | Publicat

182 Citations

Ryen W. White , Steven M. Drucker , Gary Marchionini , Ma
evaluating interfaces to support exploratory search interact
April 28-May 03, 2007, San Jose, CA, USA

Eunyee Koh , Andruid Kerne , Rodney Hill, Creativity suppo
annual international ACM SIGIR conference on Research an
Netherlands

```
{
  "acm_portal": {
    "location": "http://dl.acm.org/
citation.cfm?id=1121979&preflayout=flat",
    "pages": "41 - 46",
    "title": "Exploratory search: from
finding to understanding",
    "abstract": "Research tools ...",
    "authors": [
      {
        "location": "http://dl.acm.org/
author_page.cfm?id=81100052406",
        "title": "Gary Marchionini",
        "affiliations": { ... },
      }
    ]
    ...
    "citations": [
      {
        "acm_portal": {
          "title": "Exploratory search and
HCI: designing and evaluating ...",
          "authors": [
            {
              "title": "Ryen W. White",
            }
            {
              "title": "Steven Drucker",
            }
            ...
          ]
        }
      },
      ...
    ]
  }
}
```

title Exploratory search: from finding to understanding
abstract Research tools critical for exploratory search success involve the creation of
authors (1) name G. Marchionini
publication detail publication count 98
collaborators title See all colleagues
recent publications (10) title The Resu
affiliations (4) title Universit
source title Communications of the ACM - Supporting explora
full text title PDF full text
references (11) title The open video digital library. dLib Mag
citations (182) title Exploratory search and HCI: d
abstract The model of search as a turn-taking dialo
authors (5) name Ryen W. White
source title CHI EA '07 CHI '07 E
full text title PDF full text
references (7) title Innovation ar
citations (8) title Web User Mo
classifications (10) title CCS rev.2012

Figure 4: Metadata type system: semantic view. Types drive extraction from the source web page. The type is then joined with the resulting instance (seen as JSON), to drive presentation.

by iterating over data model fields. For each field, the integrated extraction rules are used to acquire information from the document (Figure 4a). For a scalar field, the extracted representation, a string, is converted into a value of the specified scalar type, such as integer or URL. For a composite or collection field, the process recursively instantiates and populates sub-object(s), using contextual DOM node(s) located by the extraction rule specified in the declaration of the encompassing composite or collection field.

In Figure 4a, the XPath on `citations` matches a list of contextual nodes, each of which corresponds to an anchored, formatted citation string (framed in the figure) in the original page. Formatted citation strings are parsed into key-value pairs, such as authors, title, and publication venue, using a field parser for ACM reference formats. Values are then assigned to the fields of a nested `creative_work` sub-object, such as `title` and `authors`, by `field_parser_key`. The anchor destination of a citation is extracted using a relative XPath and bound to the sub-field `location`, making the citation sub-object a pointer to linked metadata. Recursively extracting sub-objects is key to supporting nested or linked metadata of types, and experiences such as collapsing, and expanding details. The integration of data models and extraction rules enables this practical, field-by-field, recursive algorithm to derive rich metadata for heterogeneous types.

Heterogeneous Metadata and Presentation Semantics

In her task, Mia explores Wikipedia articles, research papers, and Amazon books. For Wikipedia articles, she follows links embedded in paragraphs to related concepts. For research papers, she uses references, citations, and authors to chain to related significant research. For Amazon books, she reads reviews to get others' opinions. Exploratory browsing involves encountering metadata of heterogeneous types. Each type may require *rich* presentation tailored to its specific structures and relationships, to make good use of the user's attention.

The metadata type system uses *presentation semantics* to address this heterogeneity. Integrated with data model fields, presentation semantics specify how a particular field in a particular type should be presented. Presentation semantics reference CSS classes to situate the details of presentation in abstractions, such as `metadata_h1`, separating low-level details and parameters, such as fonts, where designers can customize them. We developed a set of simple, yet effective presentation semantics, including hiding, ordering, positioning, collapsing, expanding, hyperlinking, concatenating, and changing labels for fields. In Figure 4b, `layer` decides the order of fields in presentation, and `navigates_to` specifies hyperlinking the field to a destination indicated by another field.

Presentation semantics can be inherited along with data model fields, and overridden as needed, promoting reuse. For example, `layer` specifications in wrapper `scholarly_article` will be inherited by subtypes such as `ieee_xplorer` and `acm_portal`, if not explicitly overridden. Thus, the field order specified in the base type `scholarly_article` will automatically apply to metadata extracted from any of these digital libraries.

Interfaces can render the same presentation semantics in different, yet consistent ways, to meet situated needs. The example, MICE, provides a default hierarchical HTML5 rendering, which will be explained in the next section.

Recursive Expansion of Heterogeneous Metadata

Being able to navigate to linked information with one click is crucial for web usability. By providing previously unanticipated information that evolves the user's understanding and information needs, links function as the basis for exploratory browsing and berrypicking. Mia encounters links that lead to new information, such as related concepts, names of recognized researchers, citations, and books that people also buy. Exploratory browsing interfaces must support such encounters with linked information, while maintaining context.

The example, MICE, uses *recursive expansion of heterogeneous metadata* to address this. A link, such as a citation, is initially presented as an abridged metadata object, with only the title; a plus button indicates further information. When the user clicks the plus button, MICE calls the underlying typed metadata service to extract detailed metadata from the linked document. After extraction, the service sends extracted metadata and the corresponding wrapper back to MICE, for presentation. Upon receipt, MICE recursively binds data model fields with extracted metadata values, and then iterates over these fields to generate HTML5 elements for presentation. Interface generation uses presentation semantics to customize display for each particular type, including sorting fields, hiding or changing labels, and hyperlinking.

For example, in Figure 4c, the scalar field `title` is presented as a header, anchored to the source ACM Digital Library page, as specified by `navigates_to`. The fields `authors` and `citations` are presented as lists of nested or linked metadata, initially with 10 items and a "show more" button. On expansion, the generated HTML5 elements are injected, replacing the abridged form with a detailed presentation. A sub-object whose `location` field points to a linked document, such as a citation, can be further expanded, which will recursively trigger the information expansion process.

The whole process of selecting the appropriate type, extracting metadata from the document, and presenting metadata with customized visual elements is dynamic, that is, executed in real time as the user encounters the document. Thus, the interface is able to dynamically present heterogeneous information as metadata that can be conveniently collapsed or expanded to the user in real time, while addressing characteristics of particular types.

Document Subtype Polymorphism

In programming languages, *subtype polymorphism* allows for general functions to operate on instances of different subtypes of a common type, enabling different behaviors at runtime and promoting reuse. In the metadata type system, *document subtype polymorphism* is a key to addressing heterogeneous information types and sources. The runtime provides general functions, such as metadata extraction and presentation, which operate on the general base type `document`. The type system and runtime then *polymorphically* operate on subtypes of `document`, such as `scholarly_article` and `amazon_product`, to extract heterogeneous metadata with different structures and contents, and consistently display them with rich presentation. This polymorphism is operationalized by dynamic bindings of documents to types integrating data models, extraction rules, and presentation semantics, and the invocation of extraction and presentation functions (Figure 2).

The type system comes with a wrapper repository [25] addressing a range of information types, including books, movies, patents, products, hotels, social media, and searches. As new polymorphic document subtypes are introduced, exploratory browsing interfaces building upon the type system, such as MICE, immediately support them.

USER STUDY

We designed and conducted a 2x2 within-subjects experiment to validate our hypotheses that: (1) metadata will serve as an effective form of summary, and (2) dynamic exploratory browsing interfaces like MICE will support exploratory browsing tasks better than a typical web browser. In the task context, students from an information retrieval class used *citation chaining*, the process of following references, citations, and authors for exploratory browsing, to conceptualize a project for the class. Independent variables we manipulated were *initial document set* and *interface*, each with two conditions. The instructor picked two topics for initial document set: *query log* and *PageRank*. Each initial set consisted of 7 scholarly articles from ACM Digital Library, IEEE, or CiteSeerX. The experiment interface condition uses MICE for exploratory browsing, while the control interface condition uses a regular web browser and hyperlinks.

We recruited 8 undergraduate (1 female, 7 male) and 5 graduate (all male) students who were taking or had taken the class. None of them, nor the instructor of the class, was affiliated with our lab. The study process for each participant consisted of a survey (5 min), an introductory video (5 min), two sessions of exploratory browsing (25 min x2) with different initial document set and interface conditions, and a survey (5 min). Conditions were counterbalanced. In each session, the participant spent 5 min on an interface tutorial video before engaging in exploratory browsing with papers. Participants used CiteULike to collect interesting papers in all conditions.

We recorded browser interactions and collected articles. A two way ANOVA shows students spent significantly less time directly browsing digital library web pages when using the MICE interface: .83 minutes compared to 16.43 minutes for the control ($p < .001$). This indicates that though the students could browse the original digital library web pages from MICE, they overwhelmingly did not need to, since metadata summaries presented by MICE were sufficient for them to perform the task. There was no significant difference in the number of collected papers between conditions.

The questionnaire asked participants about their experiences with both interfaces, gathering Likert scale quantitative and open-ended qualitative data. The Likert scale ranges from -4 (strongly preferring control interface) to 4 (strongly preferring MICE). Participants rated MICE better than the control in all four dimensions of experience. A single sample one-tailed t-test with $\alpha = .95$ and $\mu < 0$ as the alternative hypothesis showed statistical significance for each (Table 1).

Qualitative data analysis depicts aspects of user experience:

1) *Concise representation.* Users reported the concise representation of metadata summaries helped them browse while citation chaining:

> u8: [MICE] provides a much better method to chain documents by saving space and condensing the data for users to read and skim through.
>
> u12: The compactness of the UI makes it easier to go through a chain without losing track of where you started.

2) *Less digression.* Users said that the control interface often left them confused about how they got there:

> u3: With the web page [control] method, I quickly got off topic and had to keep multiple tabs open.
>
> u6: [MICE] better shows how papers are related and shows how I got to them.
>
> u2: [MICE] allowed me to traverse through documents while seeing where I was in relation to my past clicks. Whereas the [control] method required me to click the 'back' button anytime I wanted to backtrack on links.
>
> u4: Seeing how papers reference each other was much simpler in the tree view, as opposed to relying on memory and wondering how I got to the current paper from where I started.

3) *Supports comparison.* MICE supported knowledge formation that users thought would be missed while using the control interface:

> u7: It is easier to see all the surrounding papers, the ones cited by the paper, referenced by the paper, and the surrounding citations.
>
> u3: MICE definitely gave much more useful information than did the web page [control] method. Each factoid linked directly to other papers that shared some similarity through that particular fact.

4) *Integrated view mitigating disorientation.* Students found MICE's integrated view to be valuable and useful. Student u7 found MICE's visualization of cycles helped him understand which papers he had already seen.

> u10 : With MICE, I was able to see more diverse papers in the same viewing space, ... I discovered even more interesting papers from other topics. With the [control] method, interesting papers were more narrow in topic. I had to navigate further to find the next set of interesting papers.
>
> u1: MICE seemed quicker. I like using tabs for doing broad searches like this, but being able to see all the relations on one screen is very useful.
>
> u7: The red line linking the same paper... [helps you] see what papers you have already looked at.

Qualitative data further confirm that metadata works as a form of document summary. The concise representation of metadata summaries helped students read large amount of information and rely less on visual memory. Relationships between previously and newly browsed papers were visible through metadata fields, helping students keep their browsing sessions on track.

Question	Rating μ	p
(interesting) Which method helps you better find interesting papers along the citation chain?	1.46	.009
(overview) Which method helps give you a better sense of the referred or cited papers, before you actually read the paper?	1.70	.002
(overall) Which method do you prefer to use, overall?	1.46	.007
(citations) Which method is easier to use for citation chaining?	2.70	$< .001$

Table 1: Mean user ratings on a scale from -4 (strongly preferring control interface) to 4 (strongly preferring MICE), and t-test statistics.

Overall, MICE helped students understand context, browse related papers, and build knowledge through citation chaining. Participants preferred MICE for the exploratory browsing task. The time they spent in MICE instead of in the control interface shows that the metadata effectively summarizes digital library entries. The results show that the present interface supports exploratory browsing, while maintaining context for the user.

DISCUSSION

We need to discover new methods for making the world's vast, growing information resources more valuable to humanity. Consistently structured metadata representations of widely-used web documents enable summarization, usable presentation, and exploratory browsing experiences that maintain context. From our experiences with the metadata type system and the MICE interface, we derive implications for designing and engineering exploratory browsing and search interfaces supporting open-ended tasks:

Use metadata to represent summaries of web documents. Metadata summaries provide unique value to users browsing large collections of documents. Fields can function as facets, facilitating tasks that involve quick scanning, filtering, and comparison of multiple items, such as sorting products by price or finding most cited papers. Hierarchically nested structures enable collapse and expansion of details, helping users to better allocate their limited attention and make sense of information at different levels of abstraction. Representing linked metadata is an expandable field, in which the field name represents the relationship, helps users form mental models of citation chains, and so to acquire new knowledge.

In the study, participants with MICE spent 2 orders of magnitude less time (.84 vs. 16.43 min on average) viewing digital library pages, showing that metadata provided by MICE effectively summarized the source documents for the exploratory browsing task. While MICE presents metadata in a table-like structure, other interfaces can use different layouts driven by the same underlying metadata types.

Usable presentation of metadata summaries requires clearly presenting abundant details on the wild web, while managing redundancy and noise. Real world metadata is full of details and cross-references. Abundant details reveal the inherent complexity of the world. Details support various contextualized and personalized user tasks. Tufte wrote: "Detail cumulates into larger coherent structures ... To clarify, add detail." [41] Abundant details, when properly arranged, make full use of human capabilities of processing information, reduce the need of visual memory for switching contexts [41], and thus are essential to usable presentation of metadata summaries. In the MICE study, details such as references and citations, which are often absent in prior approaches to metadata summaries [14] [36], enable citation chaining.

On the other hand, inevitable redundancy and noise in real world metadata distract users' inherently limited cognition. The more information that is presented, the more cognition is consumed [38]. Presentation semantics can focus metadata displays toward usability. For example, DOIs [26] and URLs [4] of browsed papers are more useful to the machine than the user. Presentation semantics thus guide the dynamic exploratory browsing interface to avoid direct display of this information to the user, while using it as the destination of a hyperlink on the title field.

Integrate the meta-information of data models, extraction rules, presentation semantics, and type selectors to drive effective, usable metadata summary experiences. To provide value to the user, data models alone are not sufficient. Individual metadata summaries must be extracted and usably presented. Metadata data models, extraction, and presentation are inherently intertwined in user experiences. Integrating meta-information of data models, type selection, extraction, and presentation provides a general method for generating rich presentation. The structures of abundant details are expressed through metadata types, providing for consistency and variation, while enabling management of redundancy and noise. Extraction rules recursively acquire pieces of information from the DOM to form typed metadata instances. Presentation semantics enable hiding, relabeling, reordering, emphasizing, hyperlinking, expanding, collapsing, and concatenating fields, to generate type-specific rich presentations, addressing diverse use cases from the ACM Digital Library to Amazon and beyond.

The integrative metadata type system, with inheritance, helps scale metadata extraction and presentation to many information sources and types. The selector mechanism automatically picks the optimal type for each encountered document. This is essential for expanding serendipitously encountered metadata. Exploratory browsing interfaces, like MICE, operate on the base type of document, while using integrated types to drive presentation. For development and maintenance, the type system supports reuse and overriding of data model fields, extraction rules, and presentation semantics through inheritance [7].

Operate on popular and useful web information. Popular websites are, inherently, repositories of information that matters to people. The Semantic Web approach assumes that they will be published using standards for machine-understandable, linked data. Based on this assumption, many Semantic Web applications treat metadata as the result of preprocessing performed in advance. SPARQL queries then retrieve metadata for presentation. Alas, many useful web sites publish only semi-structured HTML, with human-oriented markup and styles, rather than RDF, OWL, or microdata. While a WWW 2007 paper articulated the need to connect semantic web and Web 2.0 approaches [1], `programmableweb.com` shows that six years later, RDF plays a role in less than 1% of registered APIs.

The present research enables extraction and presentation of metadata from a wide range of popular and useful web sites. This is important for building interfaces that provide immediate value to users by supporting everyday scenarios. Working with popular and useful web sites also enables investigation of *real world* use cases, which leads to holistic, deep understanding of people's practices with web information, especially semantics. This is crucial for driving research into the design and engineering of interactive information systems.

CONCLUSION

Based on object-oriented programming concepts and constructs such as inheritance, polymorphism, and dynamic dispatch, this research develops a novel approach to engineering usable exploratory browsing interfaces working with heterogeneous web semantics. Types integrate metadata data models, extraction, and presentation. Seemingly contrary to common practices of separating concerns, this integration is demanded by how these aspects of exploratory browsing are inherently connected in vital user experiences. Integrative metadata types operationalize dynamic exploratory browsing interfaces by enabling recursive extraction and usable presentation of heterogeneous metadata summaries from diverse sources.

The example exploratory browsing interface, MICE, enables browsing summaries of web pages through linked metadata, which can be dynamically expanded. The dynamic nature of such interfaces is essential to exploratory browsing. When the user serendipitously seeks to explore new information encountered through links, the interface dynamically expands, using types to customize metadata derivation and presentation. Further, newly published information can be dynamically incorporated. Thus, the metadata type system fundamentally differs from technologies that only operate on datasets assembled in advance.

The custom presentation semantics specified in types, such as ordering, formatting, hiding, and hyperlinking field values, enable type-specific emphasis that can mitigate the cognitive load inherent in browsing large amounts of information. The type system and MICE constitute a practical method for building web-scale dynamic exploratory interfaces. Study participants found MICE's concise presentation of linked metadata usable and valuable for exploratory browsing.

Disorientation and digression constitute deep rooted problems in popular user experiences of web browsing. A solution to this is to present summaries of multiple documents in a continuous space, maintaining context. Metadata summary representations produced by the type system enable reduced, yet expandable presentation of web pages. Presentation semantics enable the user to browse original web pages, as needed. Study participants found that MICE helped reduce disorientation and digression by displaying metadata in one context and making relationships visible, including to previously encountered information.

Dynamic interfaces based on the metadata type system have the potential to transform browsing experiences with web information for a wide range of open-ended, exploratory tasks. Exploratory browsing interfaces can be embedded into HTML pages to transform passive hyperlinks, enriching diverse, integral, 21st century information experiences, including digital libraries, shopping, social networks, messaging services, email clients, and newspapers. Our open source implementations of the type system and MICE [25] have the potential to facilitate the engagement of research, open source, and industry communities in engineering new interactive systems in diverse domains for exploratory browsing and search.

The metadata type system enables a new family of dynamic interfaces that help users browse the WWW. Support for exploratory browsing while maintaining context will be valuable in many sensemaking and berrypicking tasks. Future research can incorporate these techniques with query input and history to develop new support for exploratory search.

REFERENCES

1. Ankolekar, A., et al. The two cultures: mashing up web 2.0 and the semantic web. In *Proc of WWW* (2007), 825–834.
2. Antoniou, G., and van Harmelen, F. *A Semantic Web Primer*. The MIT Press, 2004.
3. Bates, M. The design of browsing and berrypicking techniques for the online search interface. *Online review 13*, 5 (1989), 407–424.
4. Berners-Lee, T. RFC 1738: Uniform resource locators (URL). *RFC* (1994).
5. Berners-Lee, T., et al. Tabulator: Exploring and analyzing linked data on the semantic web. In *Proc SWUI* (2006).
6. Bollacker, K., et al. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. of SIGMOD* (2008).
7. Booch, G., et al. *Object Oriented Analysis & Design with Application*, 3 ed. Addison-Wesley, 2007.
8. Bracha, G., and Cook, W. Mixin-based inheritance. In *Proc OOPSLA/ECOOP* (1990).
9. Bush, V., and Wang, J. As we may think. *Atlantic Monthly 176* (1945).
10. Conklin, J. Hypertext: an introduction and survey. *Computer 20*, 9 (1987), 17–41.
11. Cowan, N. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences 24*, 1 (2001), 87–114.
12. Cutting, D. R., et al. Scatter/gather: a cluster-based approach to browsing large document collections. In *Proc. of ACM SIGIR* (1992).
13. Deligiannidis, L., et al. RDF data exploration and visualization. In *Proc. of CIMS*, ACM (New York, NY, USA, 2007), 39–46.
14. Dontcheva, M., et al. Summarizing personal web browsing sessions. In *Proc UIST* (2006).
15. Dontcheva, M., et al. Experiences with content extraction from the web. In *Proc UIST* (2008).
16. Edwards, D. M., and Hardman, L. Lost in hyperspace: cognitive mapping and navigation in a hypertext environment. In *Hypertext: theory into practice*. Intellect Books, Exeter, UK, 1999, 90–105.
17. Foss, C. L. Detecting lost users: Empirical studies on browsing hypertext. Tech. rep., 1989.
18. Glaser, H., et al. CS AKTive Space: building a semantic web application. In *Proc. of ESWS*, Springer Verlag (2004), 417–432.
19. Greenberg, S., and Cockburn, A. Getting back to back: alternate behaviors for a web browser's back button. In *Proc HFWEB* (2002).
20. Hearst, M. A., and Pedersen, J. O. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *Proc. of ACM SIGIR* (1996).
21. Huynh, D., et al. Exhibit: lightweight structured data publishing. In *Proc. of WWW* (2007).
22. Huynh, D. F., and Karger, D. Parallax and companion: set-based browsing for the data web. In *Proc WWW*, ACM (2009).
23. Huynh, D. F., Mazzocchi, S., and Karger, D. Piggy bank: Experience the semantic web inside your web browser. In *Proc. of ISWC* (2005).
24. Interface Ecology Lab. Metadata In-Context Expander (MICE) Demo. **`http://ecologylab.net/mice`**.
25. Interface Ecology Lab. An open source metadata type system implementation. **`https://github.com/ecologylab/BigSemantics/wiki`**.
26. International DOI Foundation. The digital object identifier system. **`http://www.doi.org/`**.
27. Karam, M., et al. mSpace: interaction design for user-determined, adaptable domain exploration in hypermedia. In *Proc. of AH* (2003).
28. Kerne, A., et al. Meta-metadata: a metadata semantics language for collection representation applications. In *Proc. of CIKM* (2010).
29. Klemmer, S. R., et al. Where do web sites come from?: capturing and interacting with design history. In *Proc CHI* (2002), 1–8.
30. Lin, J., et al. End-user programming of mashups with vegemite. In *Proc. of IUI*, ACM (New York, NY, USA, 2009), 97–106.
31. Marchionini, G. Exploratory search: from finding to understanding. *CACM 49*, 4 (2006), 41–46.
32. Marchionini, G., and Shneiderman, B. Finding facts vs. browsing knowledge in hypertext systems. *Computer 21*, 1 (1988), 70–80.
33. McAleese, R. Navigation and browsing in hypertext. *Hypertext: theory into practice* (1989), 6–44.
34. Pham, H., et al. Clui: a platform for handles to rich objects. In *Proc. of UIST*, ACM (New York, NY, USA, 2012), 177–188.
35. Rástočný, K., et al. Supporting search result browsing and exploration via cluster-based views and zoom-based navigation. In *Proc. WI-IAT*, vol. 3 (2011).
36. Roy Rosenzweig Center for History and New Media. Zotero. **`http://zotero.org`**.
37. Samp, K., et al. Atom interface - a novel interface for exploring and browsing semantic space. In *Proc SWUI at CHI* (2008).
38. Simon, H. A. Designing organizations for an information-rich world. *Computers, communications, and the public interest 72* (1971), 37.
39. Smith, D. A. *Exploratory and faceted browsing, over heterogeneous and cross-domain data sources*. PhD thesis, U of Southampton, 2011.
40. Stegemann, T., et al. Interactive construction of semantic widgets for visualizing semantic web data. In *Proc. EICS* (2012), 157–162.
41. Tufte, E. *Envisioning Information*. Graphics Press, 1990.
42. W3C. RDF primer. Tech. rep., 2004.
43. W3C. RDF vocabulary description language 1.0: RDF schema. Tech. rep., 2004.
44. W3C. OWL2 web ontology language document overview. Tech. rep., 2009.
45. W3C. HTML5: A vocabulary and associated apis for html and xhtml. Tech. rep., 2012.
46. White, R. W., Kules, B., Drucker, S. M., and schraefel, m. Supporting exploratory search, intro to special issue. *CACM 49*, 4 (Apr. 2006).
47. Wikipedia editors. Exploratory search. **`http://en.wikipedia.org/wiki/Exploratory_search`**.
48. Wong, J., and Hong, J. I. Making mashups with Marmite: Towards end-user programming for the web. In *Proc. CHI*, ACM (2007).