NuSys: Towards a Document IDE for Knowledge Work

Authors omitted for submission
email@address.edu
Address Line 1
Address Line 2
City, State Zip Code

ABSTRACT
Knowledge workers consume and annotate digital documents such as PDF files, videos, images and text notes - in some cases collaboratively - to form mental models and gain understanding. An abundance of software solutions and utilities that were designed to assist users in stages of this process but not in the process as a whole, which makes knowledge work with documents unnecessarily inefficient. With NuSys we aim to streamline these tasks by creating an integrated development environment (IDE) specialized for end-to-end document-centric workflows. Specifically we are designing a small set of tools that span the four cornerstone tasks we have identified in knowledge work, including to retrieve, visualize & consume, externalize & lay out, and as to collaborate and share. The NuSys user interface design allows users to easily discover and compose these varied tools in order to tailor a customized, efficient user experience for their specific tasks. In this paper, we identify four tasks that capture peoples’ activities on a computer when performing document-based knowledge work, analyze how existing software supports these tasks, propose steps how they can be integrated in an IDE, and finally present and discuss a prototype that implements the core of these ideas.

CCS CONCEPTS
•Human-centered computing →User interface programming;

KEYWORDS
ACM proceedings, Bibtex, text tagging

ACM Reference format:
DOI: 10.475/123_4

1 INTRODUCTION
Many knowledge workers use websites, Google Docs, Office and PDF files, images, videos, and text files etc., which we henceforth refer to as documents, as their primary source of information. These workers typically use multiple different software solutions and utilities to retrieve, view, consume, organize, relate, annotate, and share their documents and fragments thereof. While these applications assist users in solving specific sub-tasks of knowledge work, they do not address the requirements of a complete workflow. As a result, users adopt inefficient strategies to cope with problems caused by context switches, lack of compatibility and incomplete functionality. Software vendors and various researchers have attempted to expand the depth of their target workflows by incorporating ever more comprehensive tool sets, but fall short of addressing the broad end-to-end needs of knowledge work. Adequate support is often missing even for elementary workflows such as collaboratively searching, gathering and freely arranging fragments of various media documents to gain understanding and then transforming emergent insights into interactive structured visualizations and presentations.

With NuSys, we draw on the concept of software IDEs which address workflow productivity by providing a broad set of tools with which users can visually compose and functionally coordinate to create efficient, interactive experiences for their specific workflows. In this paper, we first contrast our work to research efforts related to the problem of document handling in knowledge work in Section 2. In Section 3, we then identify four elementary knowledge worker tasks to define the scope of workflows that we believe software should support. In Section 4, we summarize the functionality of existing software solutions that support these tasks and concepts missing in these solutions. We then present the design and implementation a prototype for a knowledge worker document IDE, that supports all tasks in an integrated manner by combining a rich set of features that have been proven successful in related domains of document management, information visualization and retrieval, and knowledge externalization. Finally, in Section 6 we discuss our prototype and point to future work.

2 RELATED WORK
Some of the components we deem important for a document IDE already exist in isolation, i.e., as standalone applications. We therefore first contrast our work to research efforts in activity-based computing, window management and spatial hypertext, which have attempted to combine these components.

Activity-based computing is an alternative to the prevalent application and document-oriented approach that emphasizes user activities in the design of user interfaces more than applications and documents. It tries to better reflect how knowledge workers perform their tasks by providing easy access to activity-related documents and applications. For instance, activity-based systems
such as Giornata [26], Rooms [11], and GroupBar [23] allow users to view and arrange open windows in clusters. This makes it possible to logically group a set of documents in a new entity. Activities can either be defined manually, e.g., explicitly creating a new activity and tagging it, or automatically, where a system tries to infer the current activity through currently open documents, interaction patterns etc. [4].

There have also been efforts to provide user interfaces for collecting fragments of documents, such as The Sandbox [28], which provides a 2D workspace to mash up and annotate these clippings, and WinCuts [25], that allows for live application clippings of windows. NuSys differs from these approaches in that instead of merely providing easier access to documents, their views and editors, attempts to smooth the interplay between these systems by combining them in a single application, where all documents and clippings can be treated uniformly.

Spatial hypertext, or in our case spatial hypermedia systems, facilitate creating categories and relationships by providing means to spatially arrange information objects. Among many other benefits [22], the ease of moving an object makes spatial hypertext especially well suited to tasks where the information continually evolves [15, 21]. As [3, 19] found, using 2D spatial layouts to represent the relationship between documents is particularly useful because it takes advantage of human 2D spatial cognition. Many academic and commercial hypertext and hypermedia systems have emerged over the years, such as VIKI [21], or Storyspace [6] in combination with TinderBox [2]. However, many existing spatial hypertext applications are either simplistic or complex and correspondingly dialog-heavy. While NuSys can be seen as spatial hypertext system where relationships among collections of documents can be expressed by spatial arrangements, our system provides richer and more database-driven features to create, structure, and discover knowledge spaces.

3 TASKS IN KNOWLEDGE WORK

Through informal observations we identified a set of five iterative tasks that summarize the document handling activities of a knowledge worker.

3.1 Retrieve

A routine task of knowledge workers is to find new or previously consumed documents which satisfy some search criteria. They perform open-ended (browsing) or more targeted searches at different levels of detail, and progressively refine their queries and goals. For instance, they may first be interested in finding a summary on a certain topic and later look for more details. Document retrieval occurs iteratively and requires the effort of a user to parse and identify relevant pieces of information. Knowledge workers often want to recover and revisit intermediate work products, such as earlier states of a document, layouts of document working sets they previously generated or results of prior search filters.

3.2 Visualize & Consume

An integral part of knowledge acquisition is the act of informing oneself by consuming documents. This includes reading texts, watching videos, looking at images or figures, listening to audio recordings, etc. Documents are inherent multi-dimensional data containers that store information such as author, creation date, and specific user-assigned or automatically generated tags and metadata, as well as primary content; all of these can be visualized in different ways. Knowledge workers consume documents at custom levels of granularity, depending on the time budget for their tasks. For instance, text-based documents can be skimmed or pored over, images may be glanced at or inspected in detail and audio/video files can be viewed as visual and textual abridgements or snippets, or followed in their entirety. In many cases not all of these attributes are of interest at a given instant. Which attributes are relevant to a user is activity-dependent. Thus, users frequently choose to visualize a different subset of a document’s attributes/dimensions, or compute and render aggregated overviews. Consider an annotated PDF document, which is first displayed as a list item in a file explorer, then opened in a PDF reader, which renders a preview of each page in a grid, and upon selection displays a close-up of a specific page with all annotations superimposed on the document.

3.3 Externalize & Lay Out

In order to store or share insights gained from consuming documents, knowledge workers often express and formulate this knowledge as well as prior background by adding annotations and references, writing notes, constructing hierarchies, outlines, and summaries, and assigning metadata. These contextual artifacts help users to more easily discover, recover and share insights, which in some cases lead to more substantive creation of final work products. A particularly prevalent and informal way of externalization is to spatially lay out documents, i.e., create clusters or piles of related documents and their context [14]. A common example is how knowledge workers use their virtual desktop: this virtual space is often used as a temporary storage area for documents that users cannot yet categorize or gauge the importance of [13, 14]. Although desktops quickly become cluttered, in many cases the creator of the apparent mess is able to recover information with surprising efficiency.

3.4 Collaborate & Share

Knowledge workers collaborate and share knowledge. They often work in teams towards common goals and present intermediate and final results to stakeholders. Collaboration can happen synchronously or asynchronously, in co-located or remote settings, where teams of knowledge workers share and discuss material, request feedback and leave comments. An important property of collaboration in knowledge work is that communication is typically targeted towards known individuals, rather than the general public.

4 SOFTWARE SUPPORT

In this section, we provide examples how some ad-hoc tools are used by knowledge workers in practice to accomplish the tasks outlined in the previous section. We then discuss how the integration of such tools can be streamlined in a Document IDE.

4.1 Task Support

4.1.1 Retrieve. Ways to import and display documents in tools used by knowledge workers typically involve the execution of
multiple steps using copy/paste mechanisms or a file picker, or require the re-arrangement of windows to perform drag and drop interactions. To mitigate the "where was I" problem, applications like Microsoft OneNote [17] and Evernote [7] attempt to minimize the number of context switches between applications by providing means to clip and import entire documents or fragments directly from the web using a web browser plugin. Many applications have basic built-in search capabilities in the form of full text search. Some allow users to perform more targeted searches using filters on custom tags, other more database-oriented systems such as FileMaker [1] or Microsoft Access [16] allow for more complex query specifications using query languages such as SQL. Because finding a relevant document can be a lengthy and cumbersome task that can be thought of as a work product itself, some tools let users externalize this process for later reuse, e.g. GraphTrail [5], or the Windows File Explorer where searches can be saved. To recover previous states of a document, most applications provide undo/redo functionality or more elaborate forms to retrace the evolution of a document, such as the history feature in ShareLaTeX [20].

4.1.2 Visualize & Consume. The way documents are presented in these system varies greatly. File system-centric tools, such as Google Drive or the Windows File Explorer provide a set of predefined layouts, e.g., list and grid views, with the ability to customize such views for instance by displaying a preview for each document, or showing more information by adding columns, etc. Other tools only support a single non-customizable representation per document, but allow these documents to be freely arranged on a 2D canvas, for instance OneNote [17], Deon [12], Padlet [18]. Database-driven systems allow users to build their own custom views of documents and in some cases give users the ability to create interactive dashboards. Document analysis tools such as Jigsaw [24], on the other hand, provide summaries and aggregated views of documents and put an emphasis on visualizing relationships between them.

4.1.3 Externalize & Lay out. Examples of tools that help knowledge workers externalize their findings range from word processing applications, presentation programs, PDF readers and note-taking software, to drawing tools. Tools such as Mind Manager, iMapping [10], Tinderbox and Storyspace [2, 6] facilitate the creation of semantic relationships and the use of templates to generate HTML views. On the other end, tools like OneNote and Deon make it easy to informally structure documents in a 2D layout. Many of these applications provide some mechanism to assign tags and additional attributes in form of key/value pairs to documents.

4.1.4 Collaborate & Share. Support for collaboration varies greatly across all categories. Some of the systems that knowledge workers use were designed with collaboration in mind, ranging from synchronous collaboration features where changes to documents are reflected in real-time, to asynchronous reviewing and commenting features, e.g. in Google Docs, the online version of the Microsoft Office Suite or ShareLaTeX [20].

4.2 Towards a Document IDE

Despite the abundance of tools that knowledge workers use throughout their workflow, we know of no general purpose system that addresses the collective needs of even basic knowledge workflows. Instead, users piece together software to accomplish their tasks, for example, they may use a web browser to search for information, a notebook system to visualize and arrange clippings, a database to store structured relationships, and a screen sharing tool and email to communicate with colleagues. Managing the separate UIs, object models, data repositories and features of these multiple systems can be confusing and inefficient.

We believe that a better approach to knowledge work is to embody broad functionality with a small set of compatible building blocks within one uniform system. Specifically, we want to represent virtually all aspects of the workflow of knowledge workers as being operations on documents and their metadata. For example, the multiple notions of a view of a document, a search filter used to retrieve documents and a remote collaborator can all be represented in a straightforward way as documents themselves. By representing system functionality in terms of just a few basic document operations, the user experience can be driven by a small number of light-weight and fluid interactions that help users stay in the flow [8]; in turn, this can facilitate users in offloading interface reasoning to sub-conscious natural interactions [29].

We believe further that this uniform document-centric foundation not only facilitates interaction transparency, but also generates unique opportunities for reflecting on and reusing workflows. For instance, by treating collaboration, document viewing and searching all as operations on documents, knowledge workers will implicitly, as a byproduct of their natural workflow, be creating custom "dashboards", capturing visual search histories and blending synchronous and asynchronous interactions with their colleagues.

4.2.1 Fine-Grained, Heterogeneous Working Sets. Knowledge workers should be able to assemble heterogeneous working sets of information with no more effort than editing a text document. For instance, they may want to layout on a canvas either clippings or complete PDF, webpage, video, audio and image documents. Creating a clipping should be as simple as making an inline selection on any media type, without any need to locate and apply an external tool; and, the canvas that collects the working set of documents and fragments should itself be a document - essentially a dashboard for that working set. Collected documents should all be live, so that users can directly browse material as they collect it, without a context switch.

4.2.2 Externalize the Search/Find/Transform Process. Instead of treating searching as transient commands specified in a reusable search dialog, search queries can be viewed as dynamic documents that can be viewed, modified and persisted like any other document or annotation. For example, a search document can be created, alongside media documents, to represent the set of documents matching a query. However, since queries can be complex, each stage of a query can be represented as a document which is linked to the next stage, analogous to how comments in narrative thread can be linked to each other. The result of a query can then be considered a collection document with dynamic contents that update when any of its linked query documents are modified. Thus knowledge workers can externalize the thought process involved in creating a complex filter graph, and, in addition, traverse a found document's
links to recover how the search that led to the document. Transformations and computations can similarly be viewed as dynamic documents that can be linked to any other document including search chains.

4.2.3 Automated Document Summaries. When dealing with growing document collections, comprehending an overview of collective content becomes challenging. Unless users put in effort to summarize and structure a workspace, e.g., by organizing, tagging and summarizing documents, they themselves and others will have an increasingly hard time recalling relevant thought processes later. Knowledge workers tend not to create such thorough externalizations as they go because that would distract them from their primary focus on consuming information to form often abstract mental models. Even if a summary were created for each document, knowledge workers might each require a different summary with more or less detail. We believe that artificial intelligence can fill in some of the gaps left by manual externalization through automatically generated visual and textual document summaries that fit a target detail budget. We discuss ideas how this be implemented in practice in Section 6.

4.2.4 Flexible, User-Driven Spatial Layouts as Applets. Although documents and applications are often treated as being distinct things created by different types of people, typical applications can also be thought of as being documents containing content and UI elements. By exposing simple application building blocks, like search, transform, and layout elements, users can construct structured documents which in the limit behave like applications. For instance, grid and list view documents could be linked to a search document chain to create a stylized search output that presented only relevant document details in a customized layout; this resulting applet document could then be reused in other contexts with different search parameters. In general, as workflow tasks become more repetitive or complicated, the need to create custom applets that transform and display data can significantly improve user efficiency. The notion is not that every task requires an applet, but rather that every working set of documents is an applet; by removing artificial barriers, nothing prevents a user from evolving an ad hoc working set of active documents into a more refined, reusable task-specific applet.

4.2.5 Collaborators as Documents. Knowledge workers often need to communicate with colleagues regarding any aspect of their work, not just the simple text passage targets afforded by common annotation tools. By treating collaborative users as documents, users can be included in any working set of documents and directed (linked) to heterogeneous selections of documents and media. For instance, a icon of a user can be dragged, like any other document to a collection of documents. Users can also be added to search or transformation chains; for example, to specify the order in which users should be given documents to review and edit. In addition, since all document changes can be viewed synchronously, a message initially created when a colleague was offline can become a synchronous chat when multiple users start editing it simultaneously.

5 NUSYS

With NuSys, we implemented the core of the ideas outlined in the previous section. Our system features a collaborative workspace (Figure 1), which provides features to bring in, view, lay out, organize, annotate and share heterogeneous materials such as Office documents, multimedia files, and entire websites or fragments thereof. A workspace is an unbounded, zoomable, and pannable 2D canvas that can be shared among multiple users; changes to it are reflected among all connected clients in real-time through a cloud-based server. Following a popular “Post-It on a whiteboard” metaphor, a workspace allows users to import content as one or more (nested) documents that can be arranged, grouped, annotated, linked, and tagged in a free-form fashion for subsequent retrieval. In contrast to other information worker software, document and user-generated relationships such as hyperlinks or collections are first class citizens, meaning that they can be linked to any other items and can carry other attributes. Documents can be freely arranged into spatial groups, which can be highlighted using adornments (colored shapes or images indicating spatial regions on the workspace). It provides features to formulate and externalize complex queries, and offers functionality to create common and custom visualizations and layouts for documents. The workspace provides tools to visualize documents, allowing for images, PDFs and videos to be displayed directly in the workspace, instead of having to open them in a different application. Furthermore, NuSys’ user interface is optimized for, but not limited to pen/touch input. It incorporates digital ink, handwriting recognition and speech-to-text support, as well as various intuitive gestures that can be used to navigate and augments the documents and their attributes and relationships.

5.1 Introductory Use Case

Graduate student Ada applied for a software engineering internship at a large cooperation and got invited for a technical phone interview. The recruiters sent her a technical preparation guide in the form of a presentation that is intended to help applicants prepare for the interview. It states various general topics that will be covered in the interview, such as data structures, sorting algorithms and operating system concepts, and provides online resources that will help her re-familiarize herself with these topics. To brush up her knowledge on the material mentioned in the presentation Ada decides to browse the Internet for related material.

Ada’s workflow for gathering all the information is as follows. She looks up the name of a topic in the presentation that was given to her (e.g., Mergesort). She then uses web search to find relevant information on this topic on websites, lecture slides and in videos. To save this information, she uses the NuSys Importer, a plugin for Google Chrome, that allows her to store and tag web content that she will use later on (see Figure 2).

Switching over to NuSys, Ada now wants to look through the documents she collected in more detail. She uses NuSys’ search feature called “Tools” (see Figure 3) to create a working set of documents for each topic by first specifying a visual filter, and then dragging out the search results on the 2D workspace as a new documents. The workspace allows her to freely layout the documents and provides tools to arrange the documents in common structured layouts such as list and grids as well. Going through the
material, she adds annotations to documents by using text notes and digital ink.

Similarly to the interview topics, she also wants to build a repository of sample questions consisting of a title, a question and a possible solution that she can practice on. From the presentation she was given she opens links to sample coding questions in a web browser, browses sample questions, and uses the NuSys Importer to create blank documents and fill them with data. She starts off by using the “Create Document” option of the plugin (see Figure 2a). She then selects the title, answer and a solution to the question on the website, and assigns these attributes to the current document by typing the corresponding attribute names into the prompt box. Finally, she assigns a tag to the document that she will later use to retrieve the coding questions she has gathered (see Figure 2b).

Ada again uses Tools (Figure 3) to retrieve the set of questions she has collected. She starts creating her cheatsheet by first opening a detail view of one of the questions and dragging attribute names from the properties tab in the detail view, to create a field that will be filled with object’s property (Figure 4). To populate a list of all titles of question documents, Ada drags from an all-metadata tool to a title field, which triggers NuSys to create a new document for the result set of the current tool filter.

For each of these question titles, Ada now wants to create view for the corresponding question and solution. To do this, she drags these attributes from the detail view onto the workspace and transforms the to fields into a custom view by circling the nodes. The attributes defined in the view can now be displayed by double-tapping objects in the list of question titles.

5.2 Views as Documents
One of the key features of NuSys is its ability to view documents that knowledge workers commonly deal with inline. NuSys is capable of visualizing a variety of different formats such as images, videos, PDF files and even provides static views (via PDFs) of Office Documents such as Word and Powerpoint files with navigation controls. It allows users to flip through PDF documents, watch videos, browse through images, listen to audio files etc. Furthermore, it provides a means for users to create their own documents within the application, such as text, audio or video recordings. To some extent our system can be seen as a file explorer with no limitation as to which file types can be imported. It gracefully falls back to iconic representations for documents for which no visual representation can be rendered, and allows them to be opened from within NuSys with the operating system’s default application.

5.3 Externalization
NuSys provides various ways to externalize structures, for instance by creating hierarchies using collections, by adding document metadata in the form of tags and attributes, or relationships between
Figure 2: The NuSys Importer: a plugin for Google Chrome that allows users to seamlessly import text, images, PDFs, Youtube videos or entire websites, and even structured data to NuSys. a) depicts how the main menu of plugin is superimposed on a PDF b) shows how a user can add custom or frequently used tags after saving c) shows an example of how the plugin provides the ability to save images to NuSys with a single click d) depicts how a text selection can be assigned an attribute name that will be associated with a document.

documents using common knowledge mapping techniques, similar to the ones summarized in [9, 10].

Linking: Users can relate documents to each other by using bi-directional semantic, labeled or unlabeled, visual or non-visual hyperlinks, which can cross collections.

Collections: NuSys supports hierarchies in the form of collections, where documents and document collections themselves can be grouped in a collection. Collections are documents, and can be arbitrarily nested and behave equivalently to a document. That is, apart from their utility as hierarchical containers, collections provide a mechanism to create a new document out of existing ones. We elaborate on this concept in Section 5.5.

Attributes: Documents can be extended by adding custom attributes in the form of tags or key/value pairs.
Figure 3: A depiction of a visual filter chain created with NuSys’ Tools. In this example a user created a search that displays the document title for all documents that were created using the web import, have the generic type “Document”, have a “Problem” attribute associated with it and contain the keyword “Binary Search Tree”. Each filter can either be visualized as lists as shown in the center of the figure, or as bar or pie chart as shown on the left.

Tags: A special way to disambiguate documents is to assign user-defined tags to documents. NuSys also automatically suggests tags using its seamless ability to import results from the Microsoft Cognitive Services API.

5.4 Custom Document Visualizations
Similar to forms in data-driven systems NuSys has the ability to create custom visualizations for documents. While each document comes with a default visualization that contains the title and the document’s content, users can create custom visualizations from attributes that are associated with a document. An example of such a visualization can be found in Figure 4.

5.5 Creating Working Sets
To allow knowledge workers to mash up documents, NuSys enables users to define spatial or time-based regions for static (images, videos) or dynamic (video, audio) content respectively. Using simple interaction, users can define an arbitrary number of regions on a document without affecting the original document. Regions themselves are turned into a new document, giving them first-class document status and making it easy to assemble a new collection document from a set of regions.

5.6 Web Browser Interoperability
To assist users in information gathering, we developed a plugin for Google’s Chrome browser that simplifies the process of importing information found on websites, e.g., text, images, videos, PDF files, or entire websites as a document in NuSys. Our plugin also provides the ability to directly assign tags to selections in the web browser. Furthermore, the plugin stores information where the data was collected from, so that a user can easily find the origin of different document attributes. Plugins residing in third-party applications not only allow us to enhance content selection, as described above, but also enable our system to collect additional metadata, such as the surrounding content of imported document fragments or a browsing history encoding the path the user followed to find relevant information.

5.7 Complex Searches
Like most other tools, NuSys is equipped with a full-text search and text-based filters that allows users to search for content on workspaces or in the entire document repository. Similar to Graph-Trails [5], it features tools to visually explore the multivariate data of documents that would otherwise be difficult to do. Tools, as they are called in NuSys, let users group the documents in the repository or collections on the workspace by any attribute and visualize the results either in a list, or in a bar or pie chart showing their

1https://www.microsoft.com/cognitive-services
Figure 4: Shows an example of a custom layout created using sample coding questions, as described in Section 5.1. On the left is a list of document visualizations that only display the title of the two questions. The custom detail view on the right (shown in green) shows additional attributes of the currently selected document on the left, and updates when one of the titles is double-tapped.

By dragging out an attribute as a subordinate filter users can create result sets that satisfy multiple criteria. The visualization including the list is interactive, meaning that in order to change a filter criteria the user can simply modify their selection in the visualization. Each visualization represents a result set, which can be placed on the workspace as stack or collection of documents (see icons on the top right of each filter 3).

5.8 Collaboration Support

NuSys was built for small workgroups of up to a couple dozen concurrent users, and facilitates multiple ways for users to collaborate. With support for synchronous and asynchronous collaboration using multiple devices, our system follows a client-server model where workspaces can be accessed and manipulated in real-time by concurrent users. All changes to a workspace and documents are instantly reflected across all connected users. Collaborators can communicate with each other through a chat. Users can invite others to visit a location on the workspace, a document, or a specific location on a document by simply dragging a user icon to a place of interest. Doing so sends a chat message with a link to the location to the other user. This mechanism works not only for synchronous but also for asynchronous collaboration. Similar to invitations, users can drag their own icons onto someone else’s in order to join them at their location.

NuSys currently supports three different access permissions: private, public, and read-only that can be set on a per collection or per document basis. We may implement finer-grained permissions if our user base requests them.

5.9 Implementation & Scalability

NuSys consists of two main components: NuSys App, the client installed on user’s device, and NuSys Server, our server which resides in the Microsoft Azure cloud. Our client is a Windows 10 Universal App that can be deployed to various platforms, e.g., tablets, conventional desktops and large Interactive Whiteboards like Microsoft’s Surface Hub. In the future we may support smart phones but for now prefer to take advantage of larger screen real estate. We built our system with scalability in mind. In order to guarantee fast rendering performance even on large workspaces, we built a custom UI framework using Win2D that leverages the GPU whenever possible.

5.10 Additional Features

Form Factors: NuSys’ user interface was designed to scale from tablets and laptops to desktop computers and large screen displays. For more information, you can visit https://github.com/Microsoft/Win2D.
such as the Microsoft Surface Hub. NuSys’ user interface is optimized for, but not limited to pen/touch input. All operations within the workspace can be also be executed with mouse/keyboard input.

**History Capture:** To let knowledge workers keep track of their insights and ideas we incorporate two different features that help users maintain a history of their actions: Snapshots (a copy of the current state of a collection which itself can be displayed and interacted with as a collection on another workspace) and undo/redo functionality.

**Ink Annotations:** NuSys supports ink annotations on the workspace as well as on documents such as images, PDF files.

**Inline Presentations:** With NuSys users can present content within the application. Trails allow users to create a linear path through a document landscape which can be followed in a presentation-like manner. Users can use the keyboard or the displayed buttons to navigate back and forth in the presentation, which causes the system to smoothly transition to previous or next document.

6 DISCUSSION AND FUTURE WORK

Our current prototype has already proven useful in two simple cases: (1) where a scholar used NuSys to collect material and prepare research talks, and (2) another scholar used NuSys to do research on a topic that involved in-depth literature review of dozens of conference papers. Both scholars appreciated the ability to view, interact with and annotate their content on large canvas, the system’s responsiveness, and the collaborative features. They also pointed out that the ability to mash up documents within the system was a great time saver. While this initial feedback in encouraging, there are a number of limitations that need to be addressed before we can conduct more rigorous testing.

While we have invested a significant amount of time in developing a performant UI framework to render large corpora of documents on a single workspace, there are many details in the user interface and interactions we would like to improve. We aim to devote much of our attention to this “glue” that integrates our rich set of features to further improve the user experience.

Furthermore, there are a number of concepts and features we would like to improve or add, especially. Custom visualizations and layouts of documents in a free-form environment are a combination that we plan to further investigate. While the initial implementation works well for simple layouts, more complex views such as nested lists are not yet supported.

Similar to the web browser plugin, we have previously experimented with plugins for Microsoft Word and Powerpoint. Our goal was to update a static visual representation of such documents in NuSys anytime a Word or Powerpoint file was updated in their native environment. Although we have successfully implemented and tested the plugins, we shifted our focus to the more general goal of designing a plugin architecture such that code-savvy users could help extend the interoperability of our IDE.

Finally, we currently work on new ways to visually and textually summarize clusters and collections of documents on free-form workspaces. While we have made some progress using Latent Dirichlet Allocation (LDA) in combination with Ontology-based information extraction [27], we are still working on the algorithms robustness and are looking into expressive ways to present such summaries to a user.

ACKNOWLEDGMENTS

Omitted for submission.

REFERENCES

