NuSys: Towards a Document IDE for Knowledge Work

Philipp Eichmann  
philipp.eichmann@brown.edu  
Computer Science Department  
Brown University  
Providence, RI, United States

Robert Zeleznik  
robert_zeleznik@brown.edu  
Computer Science Department  
Brown University  
Providence, RI, United States

Trent Green  
trent_green@brown.edu  
Computer Science Department  
Brown University  
Providence, RI, United States

Andries van Dam  
andries_van_dam@brown.edu  
Computer Science Department  
Brown University  
Providence, RI, United States

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 insight. 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. In this paper, we introduce ideas on how to streamline common knowledge worker tasks, 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. Furthermore, we present NuSys, an integrated development environment (IDE) specialized for document-centric workflows, that implements the core of these ideas.

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

KEYWORDS
User Interface; Knowledge Work; Document Handling

1 MOTIVATION
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. 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. 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. Consider the simple case where a knowledge worker wants to examine information about Nobel Laureates. In a first step, she would collect information such as their name, age, origin, portrait image etc., from different sources and informally lay out her documents on the desktop, for instance. To better organize, visualize and compute the information she is interested in, she would extract fragments from her documents and bring it into a common structure, e.g., using a spreadsheet to apply filters, calculate the average age or display the number of Laureates by country. Lastly, to externalize her findings, she would create a stylized visualization of her findings and share it with others.

Despite the existence of tools that support sub-tasks of such a use case, we know of no general purpose system that addresses the collective needs of even basic knowledge 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 that can later be modified and re-used as intermediate work result.

2 DESIGN
With NuSys, we draw on the concept of software IDEs which address workflow productivity by providing a set of tools with which users can visually compose and functionally coordinate to create efficient, interactive experiences for their specific workflows. Contrary to existing solutions for knowledge work which are targeted towards specific sub-tasks, we believe that a better approach 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; in turn, this can facilitate users in offloading interface reasoning to sub-conscious natural interactions. We believe further that this uniform document-centric foundation not only facilitates interaction transparency, but also generates unique opportunities for re/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. To realize these ideas, we have built NuSys along the following four core concepts:

**Fine-Grained, Heterogeneous Working Sets (C1)**
Knowledge workers should be able to assemble heterogeneous working sets of documents with no more effort than editing a text document. For instance, they may want to layout on a canvas either clippings or complete PDF, website, video, audio and image documents, or custom visualization of documents such as just the author and title. This requires an application to display a broad set of common document types inline. Creating a clipping should be as simple as making an inline selection on any media type, without any need to switch tasks 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.

**Externalize the Search/Find/Transform Process (C2)**
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 search/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 (Figure 1).

**Flexible, User-Driven Spatial Layouts as Dashboards (C3)**
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 chain of interactive document filters to create a stylized search output that presented only relevant document details in a customized layout (Figure 1); this resulting dashboard 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 dashboards that transform and display data can significantly improve user efficiency. The notion is not that every task requires a dashboard, but rather that every working set of documents is a dashboard; 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 dashboard.

Collaborators as Documents (C4)
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, an 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.

3 NUSYS
Following the four core concepts C1-C4 outlined in the previous section, we implemented a prototype of a document IDE called NuSys. Our system features a collaborative workspace, which provides functionality 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 documents that can be arranged, grouped and nested, annotated, linked, and tagged for subsequent retrieval. A simple text editor allows users to modify text-based documents, and provides a mechanism to open/edit formats unknown to the system in their native application (e.g., Office files). 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 augment the documents and their attributes.

NuSys is capable of visualizing a variety of different formats such as images, videos, and PDF files. It also provides static views of Office Documents such as Word and Powerpoint files with navigation controls inline, and provides a means for users to create their own documents within the application, such as text, audio or video recordings (C1). Each document can hold an arbitrary number of user-defined attributes that can be added or modified through an editor. In addition, our system comes with a plugin for Google’s Chrome web browser that helps users bring in content, and piece together new documents with fragments found on the Internet.

To derive more fine-grained documents from existing ones, NuSys supports the creation of spatial or time-based clippings for static (images, videos) or dynamic (video, audio) content respectively, by marking regions of interest. Users can define an arbitrary number of clippings on a document without affecting the original document, lay out these clippings on the workspace, and, for instance, synthesize them into a new document.

NuSys applies a default visualization to common document types and additionally provides means to let users define their own layoutss of documents (C3). They can modify existing layouts or create new ones by visually arranging variables of attributes associated with a document (Figure 2). The main idea is that every document or document collection can be used as an input to another document, and filters and data transformations can applied if desired (Figure 1). On one hand, this gives users great flexibility in visualizing different attributes of their content, on the other hand, it enables them to externalize search queries by building visual query graphs, similar to GraphTrails [6] (C2).

NuSys comes with a set of predefined layouts for documents such as lists, grids and with a set of interactive layouts that can be used to perform search, called tools. Tools typically apply a transformation to the input documents, such as a grouping operation or a projection to a subset of attributes, and display the results in either one of the default layouts or visualization of aggregated values, e.g. document count per group (Figure 1).

We built our system for small workgroups with support for synchronous and asynchronous collaboration using multiple devices. NuSys 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. Connected users are represented with an icon in the lower left corner of the application. Like any other documents, users can be placed on the workspace, into collections or at specific locations on documents themselves (C4). Doing so enables them to query their and other users’ locations, which can, for instance, be summarized in a custom dashboard.

![Figure 2: Illustrates the concept of custom layouts for documents: A) Shows a default layout for an image. B) Shows how an image is displayed when brought into NuSys; the file name becomes the title attribute, the actual image becomes “content”. C) An example of a custom layout for an image document with additional, user-defined attributes “name”, “category”, and “year”. D) Shows layout C populated with an image document](image-url)
4 RELATED WORK

The idea of combining documents with user interfaces dates back to early publications such as [3, 4]. In the following, we contrast our research with more recent work in activity-based computing and spatial hypertext.

Activity-based computing attempts to encapsulate documents and applications as user activities. It tries to better reflect how knowledge workers perform their tasks by providing easy access to activity-related documents and applications. Examples include Giornata [14] and GroupBar [12]. Activities can typically either be defined manually, e.g., by explicitly creating a new activity, or automatically, where a system tries to infer an activity through currently open documents or interaction patterns [5]. There have also been efforts to provide user interfaces for collecting fragments of documents. WinCuts [13], for instance, is able to show live application clippings of selected windows. NuSys differs from these approaches in that it goes beyond providing easier access to documents, their views and editors. It treats documents, views on documents and searches uniformly as documents that can be flexibly laid out on a workspace. Furthermore, it provides a set of tools that operate on these documents, for example, to perform subsequent search, create clippings, and transform content.

Spatial hypertext and spatial hypermedia systems facilitate creating categories and relationships by providing means to spatially arrange information objects. Among many other benefits [11], the ease of moving an object makes spatial hypertext especially well suited for tasks where the information continually evolves [8, 10]. Many academic and commercial hypertext and hypermedia systems have emerged over the years, such as VIKI [10], or Storyspace [7] in combination with TinderBox [2]. A key aspect of insight formation is discovering and noting relationships. While NuSys can be seen as spatial hypertext system where relationships among documents can be encoded trough traditional (fine-grained) hyperlinks, spatial groupings and user-defined attributes, our system has less of a focus on semantics but more on the creation of flexible, user-driven spatial layouts. As such, NuSys can be seen as a mix of FileMaker [1], where users can create arbitrary data objects (documents), define layouts and add interactivity, and Microsoft OneNote [9], where content can be informally laid out in a free-form fashion.

5 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 did 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. Most of all, they appreciated the smooth interplay between structured and unstructured layouts and the ability to visually formulate search queries. While this initial feedback is encouraging, there are a number of limitations that need to be addressed before we can conduct more rigorous testing.

Although we have invested a significant amount of time in developing a performat 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. Especially designing the user interface around a visual language that allows users to populate layouts with other documents as input, and let them add custom functionality for input events is an ongoing challenge. Another area we are currently exploring is how to best present relationships between documents. In NuSys, relationships between documents are primarily expressed by spatial arrangements and hierarchical structures. This raises the question of how related documents, e.g., parent, child and neighboring documents within the same dashboard or elsewhere are presented to the user and which metric can be used to determine their relevance. Furthermore, the collaborative nature of knowledge work leads to an update dilemma; while real-time updates are desirable in many collaborative scenarios, altering or removing a document could confound other users. We are currently experimenting with different UIs that account for this problem. Finally, 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. Although we successfully implemented these plugins, we have shifted our focus to the more general goal of designing a plug-in architecture such that code-savvy users could help extend the interoperability of our software.

REFERENCES