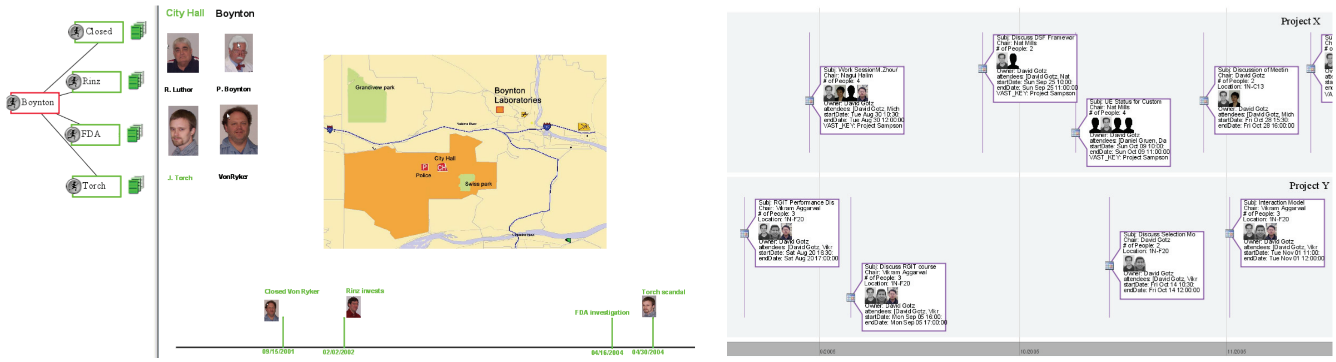
Interactive Visual Synthesis of Analytic Knowledge

David Gotz* Michelle X. Zhou† Vikram Aggarwal‡

IBM T. J. Watson Research Center



(a) (b)

Figure 1: (a) HARVEST allows users to interactively define newly synthesized knowledge. Here, Boynton is derived as a possible high-tech scam. (b) Synthesized knowledge ($ProjectX$ and $ProjectY$, inferred from calendar data) can be dynamically incorporated into the visual exploration environment to enable progressive analysis.

Abstract

A visual investigation involves both the examination of existing information and the synthesis of new analytic knowledge. This is a progressive process in which newly synthesized knowledge becomes the foundation for future discovery. In this paper, we present a novel system supporting interactive, progressive synthesis of analytic knowledge. Here we use the term "analytic knowledge" to refer to concepts that a user derives from existing data along with the evidence supporting such concepts. Unlike existing visual analytic tools, which typically support only exploration of existing information, our system offers two unique features. First, we support user-system cooperative visual synthesis of analytic knowledge from existing data. Specifically, users can visually define new concepts by annotating existing information, and refine partially formed concepts by linking additional evidence or manipulating related concepts. In response to user actions, our system can automatically manage the evolving corpus of synthesized knowledge and its corresponding evidence. Second, we support progressive visual analysis of synthesized knowledge. This feature allows analysts to visually explore both existing knowledge and synthesized knowledge, dynamically incorporating earlier analytic conclusions into the ensuing discovery process. We have applied our system to two complex but very different analytic applications. Our preliminary evaluation shows the promise of our work.

Keywords: Visual Analytics, Intelligence analysis, Problem-solving environments, Visual Knowledge Discovery

Index Terms: H.1.2 [Information Systems]: Models and Principles—User/Machine Systems H.5.1 [Information Systems]: Information Interfaces and Presentation—Multimedia Information Systems; H.5.2 [Information Systems]: Information Interfaces and Presentation—User Interfaces;

1 Introduction

Information analysis is a complex task in which analysts must make sense out of a sea of heterogeneous, potentially relevant information. In such a process, an analyst must examine existing information as well as develop analytic knowledge to progressively achieve their investigational goals. For example, a law enforcement analyst investigating an alleged case of political corruption would need to examine existing data, such as telephone records and newspaper articles, and use such information to synthesize new knowledge about potential suspects, including their social networks and political motives. Using their synthesized knowledge, the analyst can then begin to uncover the overall conspiracy.

Along with information exploration [3, 16, 22], therefore, analysts must also perform knowledge synthesis to gradually build an understanding of concepts or events that are only indirectly supported by the raw information. In this process, analysts must have tools for defining newly derived knowledge, as well as tools for modifying previously synthesized knowledge in response to new discoveries. As shown in Figure 1(a), the analyst investigating the alleged political fraud uses our tool to mark the bio-tech startup called Boynton as a potential scam, and associate it with supporting evidence, such as the FDA investigation event indicated on the timeline.

Furthermore, information analysis is a progressive process in which data exploration and knowledge synthesis occur iteratively as an investigation develops. Figure 1(b) is a snapshot of another investigation, where the analyst is analyzing business activities of an organization using employee communication data, such as calendar and email data. During the investigation, the analyst may gradually develop the knowledge of two *Projects*, *ProjectX* and *ProjectY*, and need to visually explore both the synthesized project information together with additional raw information, such as relevant employee meetings. Visualizing the two projects together with the

*e-mail: dgotz@us.ibm.com
†e-mail: mzhou@us.ibm.com
‡email: vikram@us.ibm.com

IEEE Symposium on Visual Analytics Science and Technology 2006
October 31 - November 2, Baltimore, MD, USA
1-4244-0592-0/06/$20.00

meeting data as shown here, for example, might expose the similar personnel working on both projects. Such a conclusion might never be reached if it were not possible to view synthesized knowledge (e.g. the project information) together with raw information within a single integrated visual display.

To assist analysts in interactively exploring information and synthesizing new analytic knowledge, we are building a novel, mixed-initiative visual analytic system, called HARVEST. In this paper, we focus on describing our techniques for supporting interactive, progressive synthesis of analytic knowledge. Our work in this area offers two unique contributions:

- **User-System Cooperative Visual Synthesis of Knowledge.** HARVEST allows analysts to visually define new concepts by annotating existing information discovered during an investigation. For example, users can define a new concept by directly marking one or more pieces of information (e.g., marking certain participants displayed within a visualization of phone call records as potential suspects). Users can also refine partially formed concepts by linking additional evidence or manipulating related concepts (e.g., linking two suspects). In response to an analyst's actions, HARVEST can automatically manage the evolving corpus of synthesized knowledge and its corresponding evidence. For example, when two partially defined concepts are merged together, the relevant sets of evidence are automatically combined.

- **Progressive Visual Analysis of Synthesized Knowledge.** Our system supports the integrated, visual exploration of both existing information and synthesized knowledge. This is achieved by dynamically incorporating synthesized knowledge together with raw information to form a single coherent visual exploration context. The integrated visual context allows analysts to perform progressive analysis of previously synthesized knowledge.

We have applied HARVEST to a pair of complex but different analytic problems: an organization analysis using realistic communication data (e.g., calendar appointments and meeting information) and a criminal investigation using the VAST 2006 Contest data. We have conducted informal case studies for each of these domains and our preliminary findings indicate that our approach can be useful in support of complex analytical tasks.

The remainder of this paper is organized as follows. We first provide an overview of related research. We then outline the overall HARVEST architecture and explain our approach to interactive visual synthesis. We then present our two case studies. Finally, we conclude and discuss directions for future work.

2 Related Work

Our work is closely related to a large number of projects on developing visual analytic tools. This extensive effort has led to a collection of interactive tools that help users explore massive and dynamic data, derive insights from examining data, and organize investigations [22]. However, much of this work has focused primarily on helping users visualize and examine existing data sets [3, 9, 15, 16, 21, 23]. In our work, we aim to support visual exploration of existing data together with newly synthesized analytic knowledge derived over the course of an investigation.

Moreover, existing tools are typically user-driven, where users manually specify the visual content and presentation details (e.g., [11, 12, 15]). In comparison, HARVEST supports a mixed-initiative visual analysis, in which the system works cooperatively with users by automating the management and visual presentation of information (e.g., raw data and synthesized knowledge).

Separately, several tools have been developed to facilitate the visual organization of an analysis. While these tools are designed to

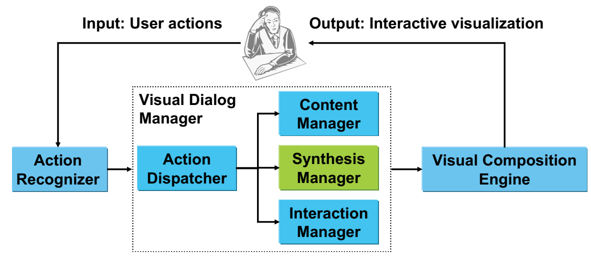


Figure 2: The HARVEST System Architecture Diagram

record and organize hypotheses and conclusions (e.g., [4]), they are often developed independent of information exploration tools. An exception to this trend is BAE's POLESTAR. This system provides both tool-sets within a single interface. HARVEST similarly supports both tasks within a single tool, but extends the integration so that synthesized knowledge can be directly exploited together with raw data during information exploration.

Our work is also related to the research activities in interactive knowledge discovery and data mining (KDD), particularly on visualizing the KDD process [5, 8, 14]. Similar to this work, HARVEST allows users to interactively manage both raw data and the derivation of analytic results. However, these systems focus only on relational data in databases and the use of automatic data mining algorithms to detect and extract data patterns (e.g., mining association rules). In comparison, HARVEST deals with heterogeneous data sets, including both structured data from databases and unstructured data, such as text documents and images. Moreover, HARVEST supports a mixed-initiative visual analysis where both systems and users can explicitly define and manipulate derived knowledge. More importantly, and in stark contrast to the typical behavior of KDD systems, HARVEST dynamically incorporates the synthesized knowledge back into the exploration process, together with the original data, for future analysis.

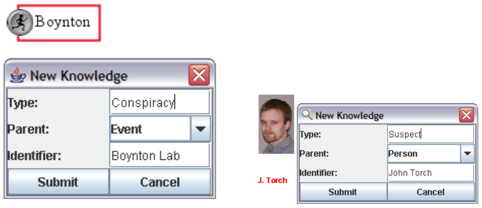
Our research is also related to a number of research efforts in the area of knowledge annotation, including social bookmarking tools for websites [7], text annotation [10, 12, 18], and image/video annotation [24]. These tools are designed for either coarse-grained annotation (e.g., keyword annotations in most social bookmarking tools [7]), annotating only existing knowledge [12, 18], or an offline annotation process separate from the human-driven analysis process [10, 24]. HARVEST supports interactive fine-grained, semantics-based annotation on both existing data and user-derived knowledge. In addition, HARVEST exposes annotation as a coherent part of an interactive analytic process.

3 HARVEST Architecture

The HARVEST architecture is composed of three main components as shown in Figure 2. Two of these components, the *synthesis manager* and the *visual composition engine*, are critical to the visual knowledge synthesis techniques described in this paper. We first provide an overview of HARVEST, and then describe the synthesis manager and visual composition engine in more detail.

3.1 HARVEST Architecture Overview

HARVEST's overall architecture is designed to support a mixed-initiative visual investigation. Analysts interact with the system via user *actions*. Given a user action, HARVEST uses the *action recognizer* to identify the type of a user action and the action parameters. Currently, HARVEST supports three types of user actions: data inquiry (e.g., searching for specific phone records), visual manipulation (e.g., directly manipulating events presented on a timeline), and



(a) (b)

Figure 3: (a) *Create* is used to define newly synthesized knowledge without direct evidence. (b) Annotation is used when direct evidence is available.

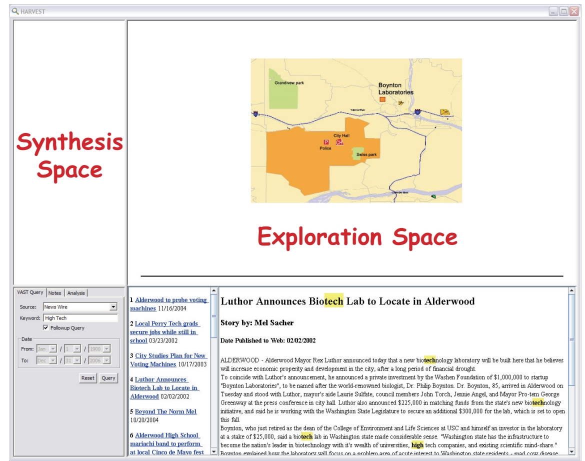


Figure 4: An overview of the HARVEST visual interface.

data synthesis (e.g., defining a suspect). Each type of action is associated with a set of parameters. For example, a data inquiry action has parameters, such as data content (e.g., searching for "Rinz") and data source (e.g., CNN news).

The recognized action is then sent to the *visual dialog manager*, which dynamically decides a course of corresponding HARVEST actions. Based on the type of the user action, the *action dispatcher* assigns the action to an action manager. Specifically, the *content manager* handles data inquiry actions by dynamically retrieving information relevant to the user's analytic context. The *synthesis manager* supports user data synthesis by automatically maintaining the evolving collection of newly derived user knowledge (e.g., a suspect identified in an investigation) and the corresponding evidence. The *interaction manager* responds to various user visual manipulations (e.g., changing the view of the displayed information). The interaction manager also handles exceptions (e.g., the desired information is not found) by suggesting follow-on user actions (e.g., modifying search criteria).

Based on the decisions of the dialog manager, the *visual composition engine* then synthesizes an interactive visual presentation [25, 26]. Such a presentation is used to either convey the data inquiry results, illustrate the synthesized knowledge with its evidence, or reflect the results of a user visual manipulation.

3.2 Synthesis Manager

The *synthesis manager* is responsible for recording and maintaining the analyst's knowledge developed over the course of an investigation. The input to the synthesis manager is a user-initiated knowledge synthesis action along with its associated parameters. Figure 3 shows two user knowledge synthesis actions, which are creating two new instances, respectively: a *Conspiracy* involving a company called Boynton Lab, and a *Suspect*, named John Torch. Given such an action, the synthesis manager processes the action and updates the corpus of synthesized knowledge as needed.

A piece of synthesized knowledge is either a new *Concept* in an existing data ontology or a new *Instance* of a new or existing concept in the ontology. We formally represent synthesized concepts and instances as follows:

$$\begin{aligned} synConcept &= \{Type, [ParentType]^*, [attribute, value]^*\} \\ synInstance &= \{Type, Identifier, [attribute, value]^*\} \end{aligned}$$

In the above representation, both the *Type* and *ParentType* denote a semantic category in a data ontology (e.g., *Person* or *Event*). The *attribute* and *value* pair denote a specific feature that can be used to describe the concept or instance. For example, concept *Conspiracy* may have features such as *participants* and *motive*. The *Identifier* in a synthesized instance is used to distinguish between multiple instances of the same type. For example, the identifier for a suspect could be the name of the suspect or the code name of the suspect if the real name is unknown. When defining a new concept or instance, the user is not required to supply the parent type and the attributes. Such flexibility is useful, since the user initially may not know how to connect the new concept with existing concepts, particularly during the early stages of an investigation.

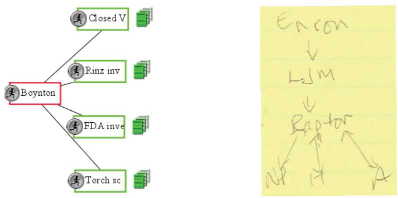
Each synthesized concept is associated with a list of instances of that concept, if any have instances been created. For example, concept *Suspect* is associated with a list of people identified as suspects. In addition, each synthesized instance is associated with a list of *evidence*, containing references to all information elements that are related to the instance. For example, after learning that the bio-tech startup Boynton Lab has close ties to both the city attorney Rinz and the city councilman Torch, the analyst may consider the company a potential scam. She can express this conclusion by creating a new instance of a new concept called *Conspiracy* which is derived from the existing concept *Event* in the ontology: $\{Conspiracy, Boynton\}$. Moreover, she associates this instance with supporting evidence, such as the Rinz and Torch connection (Figure 1a). As the investigation evolves, more evidence can be added later.

As users discover new knowledge or refine their understanding of existing knowledge, they may want to manipulate their synthesized concepts or instances. The synthesis manager allows users to manipulate synthesized concepts or instances via a set of synthesis actions, including modifying (e.g., modifying the type of an instance) and merging (e.g., merging two instances). Changes to synthesized knowledge are also forwarded to the visual composition engine to be reflected in the visual interfaces.

3.3 Visual Composition Engine

The visual composition engine is responsible for managing the composition of all visual spaces that convey information to users. This includes the choice of visual metaphors, automatic layout, and all other visual design decisions. This component is an extension of our previous work in automated visualization generation [25, 26].

There are two visual spaces in HARVEST that are directly related to the contributions outlined in this paper (Figure 4). First, there is a *synthesis space* which visually represents the corpus of synthesized knowledge developed by an analyst over the course of an investigation. The synthesis space contains iconic representations of all synthesized knowledge created during an analysis (Figure 5a). As new instances are created or manipulated by user actions, the changes are automatically reflected visually within this space. The visual design of the synthesis space is based on our observations of analyst behavior and their hand-drawn notes [6] as shown in Figure 5(b).



(a) (b)

Figure 5: (a) The HARVEST synthesis space visually represents the corpus of synthesized knowledge developed over the course of an investigation. (b) The visual design is similar to the hand-drawn figures used by some analysts to organize their investigation.

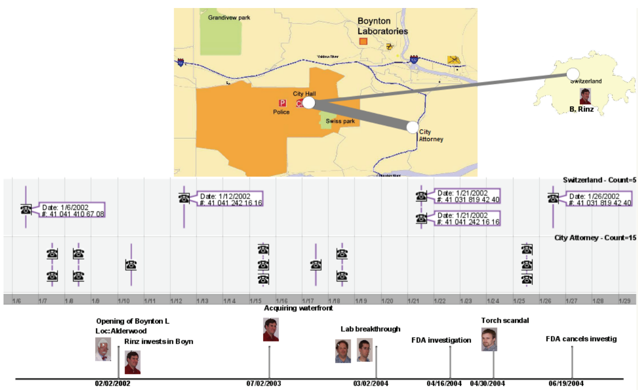


Figure 6: HARVEST's discovery space supports multi-metaphor visual composition as shown by this combination of timeline and map metaphors.

The second visual space is the *exploration space*. The exploration space is where various information is visually presented to support interactive exploration and examination. The exploration space is designed to work with both structured and unstructured data, and can display heterogeneous data collections using multiple visual metaphors (e.g., the timeline representation of events connecting to a spatial map representation as shown in Figure 6).

4 Interactive Visual Synthesis

HARVEST is an intelligent, mixed-initiative visual dialog system that supports interactive visual analysis. In comparison to previous work, one of the HARVEST's unique capabilities is its support of interactive, progressive synthesis of analytic knowledge. This functionality is made possible by two HARVEST features. First, HARVEST works with users cooperatively to help them visually synthesize and derive new analytic knowledge. Second, HARVEST can dynamically integrate user-derived knowledge with existing data to support progressive visual analysis. As a result, HARVEST enables an interactive investigation in which analysts can discover, synthesize, and progressively analyze information as they work toward an analytic conclusion. Next, we first discuss HARVEST's support for user-system cooperative visual synthesis. We then describe how HARVEST enables progressive visual analysis of synthesized knowledge.

4.1 User-System Cooperative Visual Synthesis

HARVEST supports user-system cooperative visual synthesis to leverage both human strengths (e.g., abstract reasoning and visual comprehension) and machine power (e.g., automated layout and record keeping). Specifically, HARVEST allows analysts to interactively define, manipulate, and visualize synthesized knowledge. Working cooperatively, HARVEST automates the management of an analyst's evolving collection synthesized knowledge and the dynamic visual presentation of such knowledge.

User-Driven Actions for Synthesis

Knowledge synthesis occurs throughout an investigation, concurrent with the exploration and visualization of gathered information. The synthesis of new knowledge can take many forms: the identification of investigative targets from existing information (e.g., marking a reported election event as an interest for further investigation); the formation of new hypotheses developed by instinct or from inconclusive evidence (e.g., identifying possible suspects from phone call records); or the refinement of previously synthesized concepts as an investigation unfolds (e.g., discovering a new alias for a previously discovered suspect).

Accordingly, HARVEST provides a set of tools that supports interactive knowledge synthesis in different situations. To achieve a systematic and consistent design of these tools [1], we categorize them based on different *user actions*. Currently, HARVEST supports six principal action-based tools for knowledge synthesis: *Create*, *Annotate*, *Modify*, *Merge*, *Remove*, and *Link*.

Create. The *Create* action is used to construct a piece of knowledge that does not exist previously in the knowledge base. In particular, a user can create a new concept. Consider the investigation of business activities of an organization using corporate communication data, including calendar and email data. Since the existing data themselves do not directly reflect higher-level semantic entities, such as *Project*, the *Project* concept does not exist initially in the knowledge base. As the investigation develops, the analyst may *Create* a new concept called *Project* to be linked to future findings, such as discoveries of different project instances. A user can also *Create* a new *instance* that does not exist in the knowledge base. In our example of investigating an alleged political scandal, the analyst may gradually develop a theory that a bio-tech startup called Boynton Lab is at the center of the conspiracy. To express her knowledge, the analyst would *Create* a new instance called $\{Conspiracy, Boynton\}$ (Figure 3a). A new instance instantiated via the *Create* action is initialized with no related evidence. The relevant evidence can be added later via the *Link* action described below.

The *Create* action is typically performed when an analyst synthesizes new knowledge (e.g., a new hypothesis or suspicion) that does not have *direct* support from any specific element of information. If such direct support is available, an analyst can use the *Annotate* action described below to synthesize a piece of new analytic knowledge.

Annotate. The *Annotate* action allows a user to attach additional semantic knowledge to a piece of information, which we call the *target*. In HARVEST, targets for the *Annotate* action are "raw" elements of information gathered from a user data query.

An analyst can annotate a piece of existing information to express his assessment of that information. For example, as more information is gathered, the analyst investigating the political fraud may consider councilman John Torch a suspect due to his close ties with Boynton Lab, which might help finance his mayoral campaign. Figure 3(b) shows the annotation GUI used by the analyst to label John Torch as a suspect. As a result of the annotation, a new instance $\{Suspect, John\ Torch\}$ is added in the synthesized knowledge base and is also iconically represented in the synthesis space.

Due to incomplete information, a user may perform a *partial annotation*. Suppose that an analyst is examining photos that capture the participants of a suspected terrorist planning meeting. Due to missing information, the analyst cannot identify one of the participants. She will then annotate this person with a partial annotation: $\{Suspect, Unknown\}$. In this case, the new instance with an *unknown* identity is added to both the knowledge base and synthesis space for further analysis throughout the investigation.

Due to a lack of information or inconsistent information, an analyst may also want to mark a piece of information as an item of interest instead of making an assessment. For example, knowing very little about the incumbent mayor Rex Luthor initially, the analyst may simply mark the mayor as an investigative interest in the synthesized knowledge base for future consideration: $\{PersonOfInterest, \textit{Rex Luthor}\}$

Modify. During an investigation, analysts often deal with incomplete and inconsistent information. It is likely that analysts might make wrong assumptions or vague assumptions at the beginning of an investigation. Thus, analysts may need to rectify or refine their initial assumptions as the investigation develops. The *Modify* action is provided to allow analysts to alter their synthesized knowledge as new evidence is gathered. Unlike a *Create* action, a *Modify* action must be performed on an existing concept or instance in the knowledge base. The modification can occur in two main ways. One is that an analyst alters his previous assumptions, and the other is that the analyst refines his previous assumptions.

As the investigation evolves and new information is gathered, an analyst may find his previous assumption is incorrect. For instance, in our example of investigating the political scandal, the analyst may initially consider the city councilman John Torch's involvement with a Boynton Lab employee a marital affair $\{MaritalAffair, \; Torch\ Affair\}$. However, as the investigation continues, the analyst discovers that the Boynton Lab employee involved in the affair is a highly visible figure and Torch is running for the city mayor but short of campaign funding. At this point, the analyst may want to alter her previous assumption by changing the *Torch Affair* to a potential political scandal: $\{PoliticalScandal, \; Torch\ Affair\}$. Note that the user cannot use a *Modify* to change the type of a synthesized concept, since the type change is equivalent to creating a new concept from scratch. However, a user can modify the attributes of a synthesized concept (e.g., adding attribute *Milestones* to the synthesized concept *Project*).

As more information becomes available, analysts may also want to refine their previous vague knowledge or assumptions. For example, during a criminal investigation, the analyst may discover an entity named *Blimpy*, though initially there may not be enough evidence for the analyst to conclude whether *Blimpy* is an object, an animal, or a person. In this situation, the analyst would simply indicate in the synthesized knowledge base that *Blimpy* is an object of interest: $\{ObjectOfInterest, \; Blimpy\}$. As more evidence is discovered, the analyst may develop a more refined theory that *Blimpy* is likely to be a weapon rather than anything else. The analyst now can refine his previous assumption to $\{Weapon, \; Blimpy\}$.

In addition to refining the type of a synthesized instance, the user can also refine other parameters of the instance. In our example of analyzing the business activities of an organization, the analyst has initially synthesized several *Project* instances based on the employee meeting data. As the investigation continues, the analyst is able to learn more about each project. Figure 7 displays two such project instances in the synthesis space and the refinement of *ProjectX*, including the information of its possible participants and sponsor.

In short, as demonstrated by the above examples, the *Modify* action allows users to perform progressive analysis by gradually amending and enriching the synthesized knowledge.

Merge. The *Merge* action is used to combine two synthesized instances together into a single logical entity. In HARVEST, *Merge* is only applicable to synthesized instances, since merging two concepts is considered creating a new concept. As part of the action, the user is asked to disambiguate between any conflicting information,

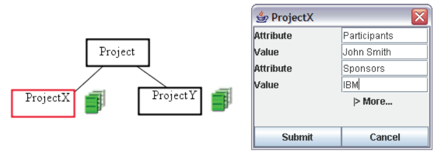


Figure 7: A *Modify* is used to refine the definition of *ProjectX*.

Figure 8: The *Link* action was used to connect the Boynton Lab Scandal to supporting evidence.

such as types and identifiers. The *Merge* action is particularly useful when an analyst discovers that they have unknowingly synthesized the same concept independently during two different stages of an investigation.

For example, the two projects, *ProjectX* and *ProjectY*, derived by an analyst, involve two different but overlapping sets of people (Figure 1b). As he continues to investigate, more evidence may indicate that these two projects are in fact the same project which happens to be named differently under different circumstances. The analyst would then select the two project instances as targets for a *Merge* action, combining their evidence lists under a single merged instance.

Remove. The *Remove* action is used to discard a user-synthesized concept or instance from the synthesized knowledge base. A *Remove* action could be initiated, for instance, if a particular conclusion proves unfounded after further investigation. When a user initiates a *Remove* action, the instance is removed from the knowledge base and any references to the instance found elsewhere in HARVEST are automatically removed at the same time.

In our political fraud investigation, the analyst may initially hypothesize a land grab event. However, as the investigation develops, the analyst may come to believe that the event never occurred. She would then remove all references to the debunked theory by selecting it from the synthesis space and issuing a *Remove* action. In practice, the *Remove* action should be used sparingly because even concept instances shown to be false may prove useful in future stages of analysis.

Link. The *Link* action is used to directly link evidence to a piece of synthesized knowledge, such as a new hypothesis or a suspicion. As an investigation develops, users explicitly link discovered evidence to substantiate their synthesized knowledge. Using our above example of hypothesizing the Boynton Lab conspiracy, the analyst will link evidence, such as the city Attorney Rinz acting as the lab's key investor, and city councilman John Torch's personal ties with one of the lab's employees, to support her hypothesis. HARVEST automatically maintains the synthesized knowledge with its linked evidence. Figure 1(a) shows the suspected Boynton conspiracy and its linked evidence.

Automated System Management for Visual Synthesis

Unlike typical visual analytic systems (e.g., [11, 13]), which require users to manually manage the details of their visual investigation, HARVEST works cooperatively with users by automating portions of record keeping and visual presentation responsibilities. Specifically, HARVEST automatically manages the evolving set of synthesized knowledge in response to user synthesis actions, and automatically composes visual presentations of both synthesized and "raw" information.

Automatic management of synthesized knowledge. Given a user's data synthesis action (e.g., *Create*), the synthesis manager responds to the action by automatically updating the corpus of synthesized knowledge to reflect the changes brought about by the action. Depending on the type of user action and its parameters, HARVEST updates the synthesized knowledge base accordingly.

In response to a *Create* action, HARVEST instantiates a new concept or instance node in the knowledge base. A newly created instance is also associated with an empty evidence list. HARVEST acts similarly when instantiating a new concept or a new instance except that when instantiating a new instance, HARVEST also automatically links the newly created instance to the specified concept. For example, when creating the new instance $\{Conspiracy, Boynton\}$, HARVEST will link the instance to the concept node *Conspiracy*. If the specified concept does not already exist in the knowledge base, HARVEST will automatically create such a concept first, then link the instance to it.

In a *Create* action, if the user specifies additional parameters other than the *Type* parameter, HARVEST performs additional operations. Specifically, if the *ParentType* is specified, HARVEST automatically links the newly created concept to the parent concept in the knowledge base. Similarly, based on the specified *attributes*, HARVEST also automatically links the new concept to other relevant concepts. For example, a user may specify that one of the attributes of a new concept *Project* is *participants*, of type *Person*. In this case, HARVEST will automatically link the concept *Project* with the concept *Person* via the relationship *participants*.

HARVEST's response to an *Annotate* action is quite straightforward. Based on the specified annotation parameters, HARVEST automatically creates a new instance in the knowledge base. In Figure 3(b), as the analyst annotates John Torch as a potential suspect, HARVEST creates a new instance $\{Suspect, John\ Torch\}$ and then links this instance to the existing instance $\{Councilman, John\ Torch\}$. Such linkage helps an analyst maintain a trace of knowledge sources, which in turn helps the analyst more easily recall where a piece of synthesized knowledge is derived from.

In response to a *Modify* action, HARVEST updates the modified concept or instance within the knowledge base to reflect its modified parameters (e.g., the attributes of a concept, or the type and identifier of an instance). If the change impacts the type of an instance (e.g., refining the type from *Thing* to *Person*), HARVEST automatically disconnects the instance from its original concept (e.g., *Thing*) and connects it to the new concept (e.g., *Person*) in the knowledge base. If the change is on the attributes of a synthesized concept or instance, HARVEST also automatically updates the relevant links (e.g., linking the modified instance to other relevant instances). For example, defining *John Smith* as the participant of *ProjectX* will automatically connect *ProjectX* to $\{Person, John\ Smith\}$.

HARVEST responds to *Merge* actions using a set of operations. A *Merge* action combines two instances together into a single unit with a type and identifier specified by the user. HARVEST responds automatically first by combining both evidence lists into a single list, removing duplicates. HARVEST will also try to combine the parameters of the two instances to form a combined list of parameters. If there is a conflict, HARVEST will prompt the user to resolve it. For example, when combining two synthesized *Project* instances together, HARVEST may detect that the two projects have different sponsors attached. It will then prompt the user to reconcile the difference.

In response to a *Remove* action, HARVEST simply removes the target concept or instance from the knowledge base along with its corresponding evidence list, if there is one. It also removes all links connecting to and from the target.

Finally, HARVEST responds to a *Link* action by simply adding a piece of evidence to the evidence list of a derived instance.

Automated visual presentation of raw information. Most existing visual analytic systems (e.g., [11, 15]) require users to specify visual metaphors and supply visualization details. In contrast, HARVEST leverages an automated graphics generation engine to dynamically decide the most suitable visual metaphors to convey the intended content (e.g., phone calls from city hall) [26]. To dynamically manage the graphical layout and maintain a coherent visual context, HARVEST also uses a dynamic space management algorithm based on [2] and an optimization-based visual context management algorithm [25]. As a result, HARVEST can automatically create an interactive visual presentation of information. To better customize a presentation according to users' personal interests and visual preferences, HARVEST also allows users to provide high-level presentation preferences. For example, a user may indicate to view the phone records on a timeline or spatially on a map (Figure 6). Like many other visual exploration systems (e.g., [3]), a user can also interact with the generated presentation to view the data at different levels of detail (e.g., zooming into a timeline to view employee meetings in a week's span versus in a month's span).

Automated visual presentation of synthesized knowledge. In addition to visually presenting the raw information, HARVEST also visually depicts user-synthesized knowledge. Once a new concept or a new instance is synthesized by the user, HARVEST visualizes the synthesized knowledge in two ways.

One way is to visualize the synthesized knowledge iconically in a graph-like structure in the synthesis space, which provides an analyst with an overview of the synthesized knowledge. Figure 5(a) displays the synthesized instance $\{Conspiracy, Boynton\}$ shown in the synthesis space along with its supporting evidence. To reduce visual clutter, the synthesized knowledge displayed in the synthesis space can be viewed at different levels of detail. For example, in Figure 5(a), the iconic display of the evidence could be collapsed and hidden from the user.

The other way is to visualize the synthesized knowledge in the exploration space as done with any other existing information. Figure 1(a) shows the synthesized instance $\{Conspiracy, Boynton\}$ along with its supporting evidence on a timeline. Depicting the synthesized knowledge graphically as any other information allows HARVEST to provide users with an integrated display with which users can perform progressive visual analysis (Figure 1b).

HARVEST also allows a user to visually link the synthesized knowledge depicted in the synthesis space with its evidence illustrated in the exploration space or vice versa. Figure 8 shows that after the user clicks on the hypothesized $\{Suspect, John\ Torch\}$, the corresponding evidence linked to the suspect is highlighted among other information shown in the exploration space. Since HARVEST's exploration space incrementally accumulates relevant information to maintain a coherent visual context [25], it is important to allow users to quickly check the correspondence of the synthesized knowledge and its evidence at any given point of the investigation.

In response to any changes made to a piece of synthesized knowledge, HARVEST checks the exploration space for any visual references to the altered knowledge. If any such knowledge happens to appear in the current exploration view, HARVEST updates the visualization to reflect the changed properties accordingly. Assume that the analyst decides to merge *ProjectX* and *ProjectY* shown in Figure 1(b). In this case, HARVEST not only will create a new iconic representation of the merged project instance in the synthesis space, but it will also visually merge the relevant evidence, such as various meetings, to form an integrated display for the merged project instance.

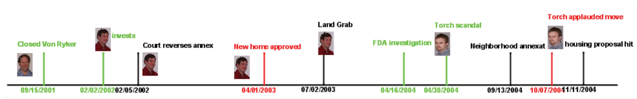


Figure 9: HARVEST allows users to view synthesized knowledge and raw data together within an integrated visual space.

4.2 Progressive Visual Analysis of Synthesized Knowledge

Unlike existing visual knowledge discovery systems [3, 12, 11, 17], where synthesized knowledge is not visually integrated with the existing "raw" data, HARVEST supports a *progressive visual analysis of synthesized knowledge* from the following two aspects.

First, HARVEST can dynamically incorporate previous analytic conclusions into ensuing stages of discovery by providing an integrated visual display. Such a display presents both synthesized knowledge and "raw" data together to form a coherent picture of an evolving analysis. Using our previous example, assume that our analyst has already formulated her hypothesis on a possible political conspiracy involving a bio-tech startup called Boynton Lab. As shown in Figure 9, the evidence (shown in green) supporting this hypothesis involves both the city attorney *Rinz* and city councilman *Torch*. Based on this information, the analyst would like to view additional activities with which both men are also involved in the context of the hypothesized Boynton Lab conspiracy. Figure 9 indicates such activities in black. As highlighted in the figure, Rinz is directly involved in several land transactions, while Torch applauded one of the transactions. While viewing the land transactions alone may not yield any clues, especially considering the city duties performed by both Rinz (city attorney) and Torch (city councilman), viewing them in combination with the discovered Boynton lab evidence however may signal correlated events and uncover additional plots in the conspiracy (e.g., the alleged land grab may be related to the Boynton Lab).

Second, using the integrated display, analysts can then perform progressive analysis by further investigating the synthesized knowledge in search of new analytical clues or conclusions [20]. To help analysts further examine and refine their derived knowledge, HARVEST allows analysts to visually explore synthesized knowledge along with their supporting evidence explicitly. For example, Figure 1(b) displays the information related to the two projects (*ProjectX* and *ProjectY*) that have been inferred by an analyst. By making such synthesized knowledge first-class citizens in the exploration space, the analyst is able to continue his investigation as he has done with any existing "raw" information. In this case, the analyst discovers the overlap between the two sets of people involved in the two projects. Without being able to visually examine the synthesized knowledge (e.g., the project instances), it would have been more difficult to discover the possible connections.

5 Case Studies

We have applied HARVEST to a pair of complex but different analytic problems: an organization analysis using realistic communication data, and a criminal investigation task using the VAST 2006 Contest data. We have conducted case studies for each of these domains but under two different settings to examine how our techniques impact analytical behavior and performance. In one setting, we disable the synthesis manager to observe how users work without HARVEST's cooperative help in performing knowledge synthesis. In the other setting, we enable the full HARVEST functionality to observe how it helps users in their interactive, progressive knowledge synthesis process.

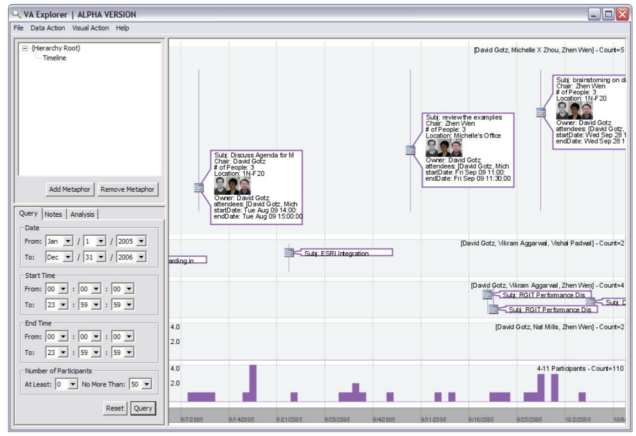


Figure 10: A screen shot from the HARVEST prototype for analyzing organizational communications data in our first case study.

5.1 Case Study One: Organization Analysis

In our first case study, we applied HARVEST to an organization analysis task in which we asked users to use employee calendar data to investigate the active projects taking place within an organization. We asked them identify the people involved, project names and milestones. Users were allowed to issue queries using a GUI form to an employee calendar database. Due to the nature of the calendar data, users were provided with an interactive timeline visualization tool for visually exploring the results returned from their database queries. A screen capture of our case study application is shown in Figure 10.

We recruited two users to perform the investigative task outlined above. In this study, we disabled the synthesis manager to explore how effectively a visualization tool *alone* can support the complex investigative pattern required to solve the analytic task. Following each session, we asked the user a series of questions to gather their feedback.

The users described our interactive timeline as "very helpful and useful" in exploring the calendar data. However, when asked about using it to perform a long-term analysis, participants listed a number of features that were lacking. For example, one of the users requested the ability of dynamically linking the new evidence to previous discoveries. Another commented on the lack of presentation on what has been discovered so far. In general, the users felt that they needed a way to record their progress during an investigation so that it could be used later in time. As one user said, we need a way to "keep the investigation context and save partial conclusions." The other user wanted a visual history of the analysis so they could "see what [they had] done before." Unsurprisingly, these observations are also consistent with the findings from our previous study [6].

Many of the missing features identified by the users in this case study are supported in the full HARVEST system. In particular, HARVEST allows analysts to easily track the investigation context by recording synthesized knowledge and discovered evidence for future review and analysis.

5.2 Case Two: Criminal Investigation

In our second case study, we exposed the full HARVEST capability to the same two users in a criminal investigation domain. For this study, we utilized the Alderwood political corruption story, as outlined in the VAST 2006 Contest challenge. The data used in this study is the VAST 2006 Contest data, including newspaper stories, telephone records, maps, and voter registration information. To enable visual exploration of events reported in the news, we augmented the data set by manually adding additional information (e.g., the geospatial locations of various places) and extracting the semantics of the unstructured information (e.g., the person and events reported in news). A screen capture of our second case study prototype is shown in Figure 4.

Once again, we concluded each session with a series of followup questions to gather user feedback. Both users responded with positive reviews of the synthesis space. They felt it could "potentially be very useful," "especially for long term investigations." Among the most important features, users identified annotation and automatic bookkeeping as especially useful, allowing them to "link knowledge with actual information" and "organize what you've found." While users commented that some surface aspects could be improved, the overall design and integration of the synthesis and exploration spaces was well received. In summary, although our studies of HARVEST are preliminary, we believe that HARVEST does show its initial promise.

6 Conclusion and Future Work

A visual analysis requires both the examination of existing information and the synthesis of new analytic knowledge. These tasks are performed progressively so that newly synthesized knowledge can become the foundation for future discovery. In this paper, we presented HARVEST, a novel visual system that supports both visual exploration of existing information and interactive, progressive synthesis of new analytic knowledge. In particular, HARVEST provides two unique features to support the interactive, progressive synthesis of knowledge.

First, HARVEST supports user-system cooperative visual synthesis of new knowledge from existing data. On the one hand, analysts can use a set of tools to visually define new knowledge by creating new concepts or instances, or indicate their investigative interests by annotating existing information. Analysts can also refine partially formed concepts by altering their definition, linking additional evidence, or manipulating related concepts. In response to user actions, our system automatically manages the evolving corpus of synthesized knowledge and its corresponding evidence.

Second, HARVEST supports progressive visual analysis of synthesized knowledge. In response to user data synthesis actions, HARVEST automatically updates the visual presentation of relevant information. Visual updates are applied to both the synthesis knowledge and its linked evidence. This feature allows analysts to visually explore both existing and synthesized knowledge, dynamically incorporating earlier analytic conclusions into the ensuing discovery process.

We have performed two case studies using the HARVEST system. In the first study, we observed users as they performed an organization analysis of communication data. In the second case study, we observed users during a criminal investigation task using the VAST 2006 Contest data. This preliminary evaluation of our work indicates that the HARVEST can be a significant aid in the analytical process.

While our current version of HARVEST shows promise, there remain several areas that can be improved upon. First, we are planning more comprehensive user evaluations to determine exactly which features are most helpful and how these features can be improved. In addition, we are exploring how to integrate HARVEST with an automated knowledge extraction system to support complex visual analysis involving large amounts of information [18, 19]. Currently, in order to visually illustrate unstructured information in a meaningful way, we have to manually extract structured semantics from the unstructured information (e.g., pictures and text documents) in advance. This would be impractical with large data sources. In the future, we would like to leverage automated information extraction technologies to extract these structured semantics on the fly while analysts are performing their analysis.

References

[1] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proc. of the InfoVis '05*, pages 111–117, 2005.

[2] B. Bell and S. Feiner. Dynamic space management for user interfaces. In *Proc. UIST '00*, pages 239–248, 2004.

[3] S. Card, J. Mackinlay, and B. Shneiderman. *Readings in Information Visualization*. Morgan Kaufmann, 1999.

[4] Mindjet Corporation. Mind Manager. http://www.mindjet.com.

[5] M. Cristina, F. de Oliveira, and H. Levkowitz. From visual data exploration to visual data mining: A survey. *IEEE Trans. on Vis. and Computer Graphics*, 9(3):378–394, 2003.

[6] D. Gotz, M. X. Zhou, and Z. Wen. A study of information gathering and result processing in intelligence analysis. In *IUI 2006 Workshop on IUI for Intelligence Analysis*, 2006.

[7] T. Hammond, B. Hannay, T. Lund, and J. Scott. Social bookmarking tools: A general review. In *Digital Library Magazine*, volume 11. Nature Publishing Group, 2005.

[8] J. Han, X. Hu, and N. Cercone. A visualization model of interactive knowledge discovery systems and its implementations. *Information Visualization*, 2(2):105–125, 2003.

[9] S. Havre, E. Hetzler, K. Perrine, E. Jurrus, and N. Miller. Interactive visualization of multiple query results. In *Proc. of IEEE InfoVis*, 2001.

[10] J. Heflin and J. Hendler. A portrait of the semantic web in action. *IEEE Intelligent Systems*, 16(2), 2001.

[11] i2 Incorporated. Analsyt's Notebook. www.i2inc.com.

[12] i2 Incorporated. Textchart. www.i2inc.com.

[13] Oculus Info Inc. http://www.oculusinfo.com/.

[14] Purple Insight. MineSet. www.purpleinsight.com.

[15] D. Jonker, W. Wright, D. Schroh, P. Proulx, and B. Cort. Information triage with trist. In *Proc. Intelligence Analysis '05*. http://analysis.mitre.org.

[16] D. Keim. Information visualization and visual data mining. *IEEE Trans. on Vis. and Comp. Graphics*, 7(1):100–107, 2002.

[17] J. Kolojejchick and S. Roth. Information appliances and tools: simplicity and power tradeoffs in the visage exploration environment. *IEEE CG&A.*, 17(4):32–41, 1997.

[18] J. Lin and B. Katz. Question answering from the web using knowledge annotation and knowledge mining techniques. In *Proc. CIKM '03*, pages 116–123, 2003.

[19] M. Naphade, R. Smith, and F. Souvannavong. On the detection of semantic concepts at TRECVID. In *Proc. ACM MM '04*, 2004.

[20] Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. of the Inter. Conf. on Intelligence Analysis*, 2005.

[21] Steven F. Roth, Peter Lucas, Jeffrey A. Senn, Cristina C. Gomberg, Michael B. Burks, Philip J. Stroffolino, John A. Kolojejchick, and Carolyn Dunmire. Visage: A user interface environment for exploring information. In *Proc. of the IEEE Symposium on Info Vis*, 1996.

[22] J. Thomas and K. Cook, editors. *Illuminating the Path: The R&D Agenda for Visual Analytics*. IEEE, 2005.

[23] J. Thomas, P. Cowley, O. Kuchar, L. Nowell, J. Thompson, and P. Wong. Discovering knowledge through visual analysis. *J. of Univ. Comp. Sci.*, 7(6):517–529, 2001.

[24] T. Volkmer, J. Smith, A. Natsev, M. Campbell, and M. Naphade. A web-based system for collaborative annotation of large image and video collections. In *ACM MM '05*, pages 892–901, 2005.

[25] Z. Wen, M.X. Zhou, and V. Aggarwal. An optimization-based approach to dynamic visual context management. In *Proc. InfoVis '05*, pages 187–194, 2005.

[26] M. X. Zhou and M. Chen. Automated generation of graphical sketches by example. In *Proceedings of IJCAI*, pages 65–74, 2003.