KBB: A Knowledge-Bundle Builder for Bio-Research

1 Research Area

We propose research into and development of a "Knowledge-Bundle Builder for Bio-Research." We direct our proposed research at the broad Challenge Area **04: Clinical Research** and the specific Challenge Topic **04-NS-102 Developing web-based entry and data-management tools for clinical research**.

The volume of biological data is enormous and increasing rapidly. Unfortunately, the information a bio-researcher needs is scattered in various repositories and in the published literature. To do activities bio-researchers need a system that can efficiently locate, extract, and organize available bio-information so that it can be analyzed and scientific hypotheses can be verified.

Currently, bio-researchers manually search for information of interest from thousands of data sources (either online repositories or publications) to achieve their goals. This process is tedious and time-consuming. As a specific example, to do a recent study about associations between lung cancer and TP53 polymorphism, researchers needed to: (1) do a keyword-based search on the SNP data repository for "tp53" within organism "homo sapiens"; (2) from the returned records, open each record page one by one and find those coding SNPs that have a minor allele frequency greater than 1%; (3) for each qualifying SNP, record the SNP ID and many properties of the SNP; (4) perform a keyword search in PubMed and skim the hundreds of manuscripts found to determine which manuscripts are related to the SNPs of interest and fit their search criteria;[1] and (5) extract the information of interest (e.g., the statistical information, patient information, and treatment information) and organize it.

In an effort to automate some of this manual tedium and speed up the search and extraction process, bio-information-retrieval researchers have worked on finding relevant documents (e.g., [BFS06, CJR+07]), but this work is directed only to specific topics (e.g., both [BFS06] and [CJR+07] focus on locating SNPs only). The research challenge for doing high-precision document filtering even for specific topics is huge, and it is even more of a challenge to generalize these ideas. How can a system do high-precision document filtering for any bio-research topic? Further, when found, how can we mitigate the tedium of extracting and organizing the relevant information so as to facilitate analysis and decision making?

The key is to answer both of these questions in unison—extract to identify and identify to extract. We take on this research challenge and propose here the idea of a *Knowledge Bundle* (*KB*) and a *Knowledge-Bundle Builder* (*KBB*). As we explain below, a KB includes an extraction ontology, which allows it to both identify and extract information with respect to a custom-designed schema. Construction of a KB itself can be a huge task—but one that is mitigated by the KBB. Construction of the KB under the direction of the KBB proceeds as a natural progression of the work a bio-researcher does in manually identifying and gathering information of interest. As a bio-researcher begins to work, the KBB immediately begins to synergistically assist the bio-researcher and quickly "learns" and is able to take over most of the tedious work.

As mentioned in the first paragraph, our proposed KB and KBB address the challenge area of clinical research and specifically address the topic challenge 04-NS-102. We quote from 04-NS-102 in terms of our research agenda: We propose "developing web-based entry and data-management tools for clinical research. The [tools] are to be open source [and to provide for] user-friendly, web-based data entry and data management. [They can] be customized by investigators [and could] serve as a core resource for the community." Further, the research we propose has the potential

[1] By the way, this took two domain experts one month just for one SNP for one disease.

to "facilitate the ability to combine datasets, facilitate data sharing, and [provide a platform for performing] data mining among clinical research datasets."

After summarizing the challenge and potential impact of our proposed research (Section 2), we describe our approach to meeting research challenge 04-NS-102 (Section 3). In describing our approach, we first give a bio-research scenario to show how our proposed KBB helps bio-researchers harvest and manage data for a bio-research study (Section 3.1). We then give the details of our research contribution by showing how we expect to be able to produce the results claimed in the bio-research scenario (Section 3.2) and how we expect to evaluate our research contribution (Section 3.3). Finally, we present the research plan for accomplishing our goals (Section 4).

2 The Challenge and Potential Impact

Achieving the specific research objective:

> Develop a system bio-researchers can use to semi-automatically build a knowledge bundle for use in bio-research.

is, by itself, a significant challenge. Gathering only and all the relevant knowledge into a useful form for analysis and decision making is a daunting task. Largely automating this task would empower bio-researchers to more easily perform the research for which they are trained. Not only would it speed up the task by shortening the length of time it takes to gather and organize information, but it would also provide study-specific knowledge repositories to analyze and augment.

The larger challenge of developing a system researchers in any area can use to semi-automatically build a KB for their research is not much beyond developing a KBB for bio-researchers. If we can develop a KBB for bio-research, one of the toughest challenges, it should be fairly easy to adapt it (possibly even just adopt it) for use in other scientific work and in general for harvesting and organizing information for any type of analysis and decision making.

Besides achieving these specific research objectives, the proposed research also addresses several challenging problems that, by themselves, can constitute research contributions: (1) the generalization of extraction ontologies, as knowledge bundles; (2) a general way to create accurate, personalized knowledge bundles; (3) a contribution to high-precision document filtering by using a KB as a sophisticated information-retrieval query; (4) further investigation into learn-as-you-go knowledge creation which allows the system to use the knowledge it has to increase its knowledge and improve its ability to perform; (5) further investigation into pay-as-you-go data integration which lets the system and user synergistically integrate schema and data incrementally; and (6) a vision of a large number of KBs as a web of knowledge superimposed over the current web of pages.

As for potential impact, the proposed research should enhance bio-research, making it easier for bio-researchers to gather and organize information. This should be particularly helpful for studies that require a custom framework of information and corresponding custom data. Further, a web of knowledge-bundles could provide the basis for a sophisticated bio-research repository.

Beyond bio-research, researchers in other disciplines can use the KBB to create customized knowledge bundles for any scientific study that requires gathering and organizing information for the study. In the larger vision of numerous knowledge bundles superimposed over the current web of pages, users should be able to query and receive direct answers to questions embedded in the web of knowledge and should also be able to directly access the web pages from which the answers to their queries have been taken.

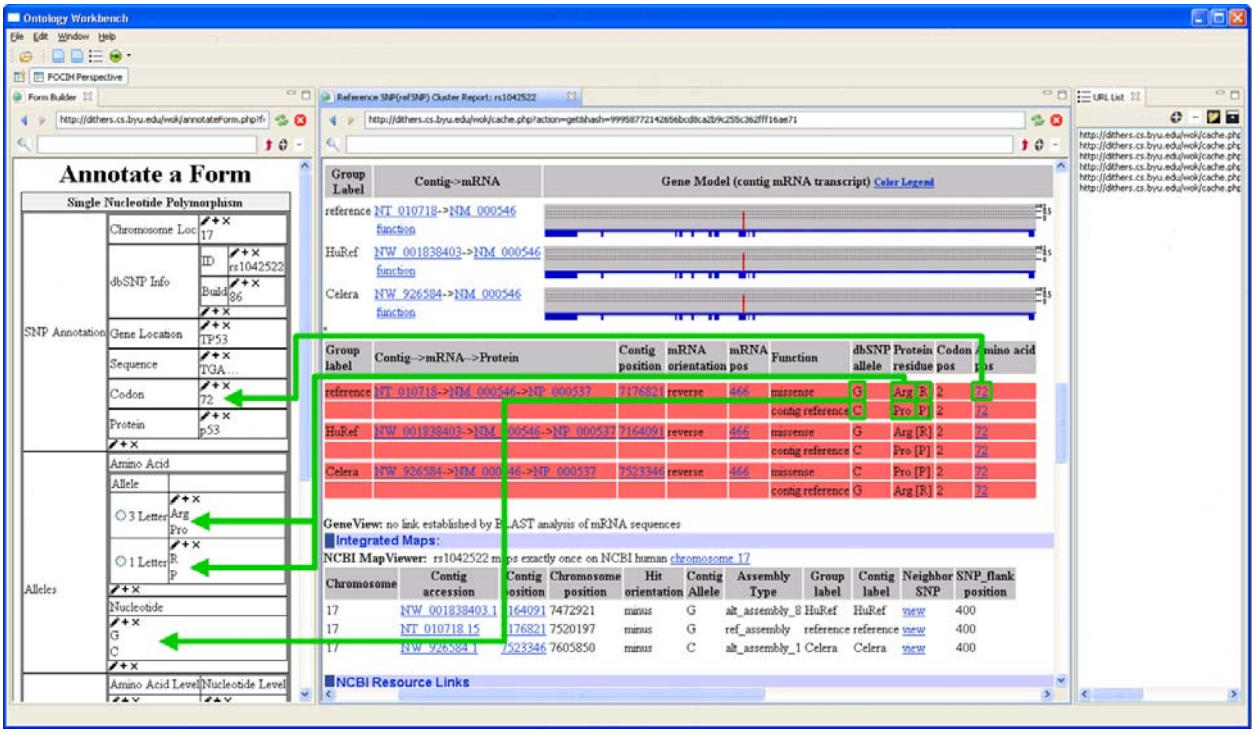

Figure 1: FOCIH Form Filled-in with Information From an SNP Page.

3 The Approach

3.1 Bio-Research Scenario

To explain our approach to the proposed research, we present a bio-research scenario that illustrates the KBB system we intend to develop. The research scenario is real enough to illustrate the KBB's usefulness in in actual clinical work, but is reduced in scope to focus attention on the critical components of the KBB. (The KBB scales to accommodate larger KBs—in principle, there is no limit to the size of KB the KBB can handle.) Further, as part of the research, we propose to design a convenient interface for bio-researchers. The interface and figures we show here are screenshots of tools we have already constructed or are mock-ups showing possibilities, as opposed to the intended final KBB interface.

Suppose a bio-researcher B wishes to study the polymorphism and lung-cancer association. The objective is to find SNPs that may indicate a high risk for lung cancer. To do this study, B wants information from NCBI dbSNP about SNPs (the chromosome location, the SNP ID and build, the gene location, codon, and protein), about alleles (amino acids and nucleotides), and about the nomenclature for amino acid levels and nucleotide levels. B also needs data about human subjects with lung cancer, including X-ray images of lungs of these subjects, and needs to relate the SNP information to human-subject information.

To gather information from dbSNP, B constructs the form in the left panel in Figure 1. Observe that the form contains form fields for the data items B wishes to harvest. B next finds a first SNP page in dbSNP from which to begin harvesting information. B then fills in the form by cut-and-paste actions, copying data from the page in the center panel in Figure 1 to the form in the left panel.

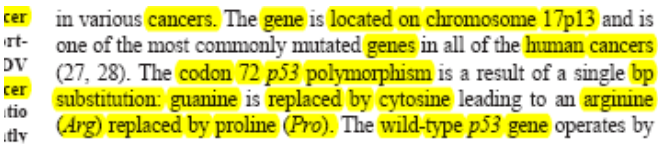

Figure 2: Paper Retrieved from PMID Using a Generated Extraction Ontology.

To harvest information from other dbSNP pages, B gives the KBB a list of URLs, as the right panel in Figure 1 illustrates. The KBB automatically harvests the desired information from the dbSNP pages referenced in the URL list. Since one of the challenges bio-researchers face is searching through the pages to determine which ones contain the desired information, the KBB provides a filtering mechanism. By adding constraints to form fields, bio-researchers can cause the KBB harvester to gather information only from pages that satisfy the constraints. B, for example, only wants coding SNP data with a significant heterogeneity (i.e., minor allele frequency $> 1\%$). Because of this filtering mechanism, B can direct the KBB to search through a list of pages without having to first limit them to just those with relevant information.

For the research scenario, B may also wish to harvest information from other sites such as GeneCard. B can use the KBB with the same form to harvest from as many sites as desired. Interestingly, however, once the KBB harvests from one site, it can use the knowledge it has already gathered to do some of the initial cut-and-paste for B. In addition to just being a structured knowledge repository, the KB being produced also becomes an extraction ontology capable of recognizing items it has already seen. It can also recognize items it has not seen but are like items it has seen. The numeric data values or DNA snippets need not match precisely with those previously seen; they only need to be numeric values in a proper range or valid DNA snippets.

Using KBs as extraction ontologies also lets bio-researchers search the literature. Suppose B wishes to find papers related to the information harvested from the dbSNP pages. B can provide the KBB with a list of papers to divide into two piles—those that are relevant to the study and those that are not. Using the KB as an extraction ontology provides a highly sophisticated query of the type used in information retrieval resulting in high-precision document filtering. For example, the extraction ontology recognizes the highlighted words and phrases in the portion of the paper in Figure 2. With the high density of not only keywords but also data values and relationships all aligned with the ontological KB, the KBB designates this paper as being relevant for B's study.

For the human-subject information and to illustrate additional capabilities of the KBB, we suppose that a database exists that contains the information needed. The KBB can automatically reverse-engineer the database to a KB, and present B with a form representing the schema of the database. B can then modify the form, deleting fields not of interest and rearranging fields to suit the needs of the study. Further, B can add constraints to the fields so that the KBB only gathers data of interest from the database to place in its KB. Figure 3 shows an example of a form reverse-engineered from INDIVO and altered to fit our research scenario.[2]

In addition to human-subject information contained as text and data values in a standard database, B also desires X-ray images of lungs for as many of the subjects as possible. With the human-subject information in the KB, B can make use of it to filter image repositories to find needed images—both for only the human-subjects being studied and for only the type of images being studied. Further, the KBB allows B to annotate the images as Figure 4 shows. The KBB stores both the graphical part of the annotation and the commentary about the graphical annotations.

[2]Since the INDIVO schema has more tables and attributes than B wants, B selects only those tables and attributes relevant to the study before reverse engineering and tailoring the form for B's needs. In many similar instances, preselecting before reverse engineering may be necessary to make the task of tailoring the resulting form reasonable.

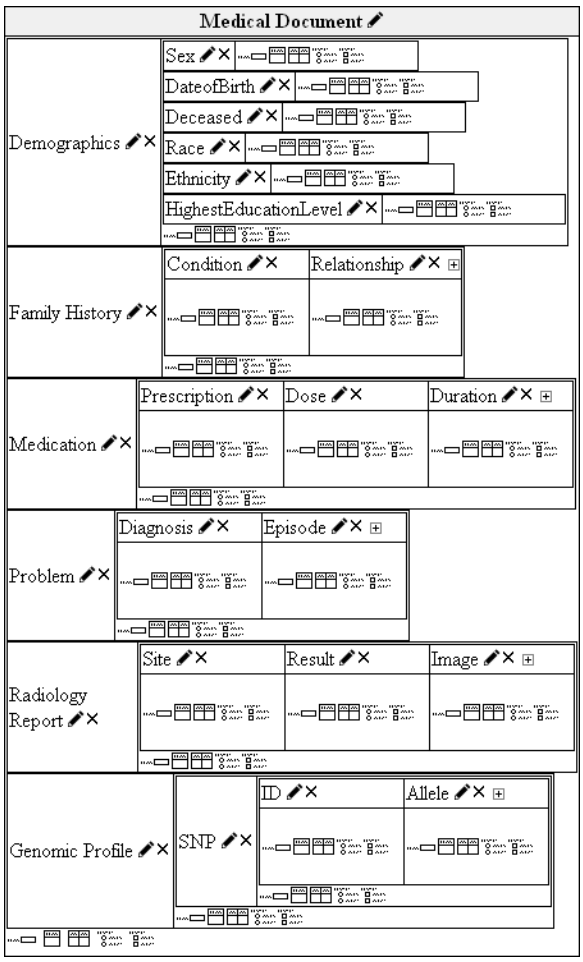

Figure 3: Human Subject Information Reverse-Engineered from INDIVO.

With all information harvested and organized into an ontology-based knowledge bundle (the KB), B can now do some interesting queries and analysis with the data. Figure 5, for example, shows a SPARQL query requesting the SNPs associated with any one of four amino acids: *Arg*, *Gly*, *Leu*, or *Trp*. For our example, we query based on information harvested from the six URLs listed in Figure 1. The query finds three SNPs and for each, returns the dbSNP ID, the gene location, and the protein residue it found. In our prototype, users may click on any of the displayed values to display the page from which the value was extracted and to highlight the value in the page. As Figure 5 shows, users may alternatively click on one or more checkboxes to access and highlight all the values in a row. Observe that *rs55819519*, *TP53*, and the *His Arg* values are all highlighted in the page in the right panel of Figure 5.

3.2 Expected Research Contribution

We propose the following specific research objective: Develop a system that bio-researchers can use to semi-automatically build a knowledge bundle (KB) for use in bio-research. We call our proposed system KB-Builder (KBB). A stipulation for KBB is that a bio-researcher must always be able to complete the task of building a KB and that the task should never be harder than creating a form with possible constraints and filling it in. We direct research, however, toward automating as much of the process as possible. Further, although we specifically target bio-research, we point out that nothing in our approach limits us to the domain of biology. Thus, we propose as a general research objective: Develop a KBB, a system that researchers, investigators, or decision makers can use to semi-automatically build KBs to aid in analysis and decision making.

We define a *knowledge bundle* (KB) as a 6-tuple (O, R, C, I, A, F).

- O is a set of object sets—sometimes called concepts or classes; they may also play the role of properties or attributes. (Examples: *Person*, *Amino Acid*, *Sample Number Color*.)
- R is a set of relationship sets among the object sets. (Examples: $Person(x)\ has\ Disease(y)$, $Sample(x)\ taken\ on\ Date(y)$.)
- C is a set of constraints that constrain the objects and relationships in O and R. (Examples: $\forall x \forall y((Person(x)\ has\ Disease(y)) \Rightarrow Person(x))$, $\forall x(Sample(x) \Rightarrow \exists^1 y(Sample(x)\ taken\ on\ Date(y)))$
- I is a set of object and relationship instances embedded in O and R that satisfy C. (Examples: $Color('\text{green}')$, $Sample('\text{SMP9671}')\ taken\ on\ Date(2009\text{-}03\text{-}25)$.)

- A is a set of annotations for object instances in O; specifically, each object o in O may link to one or more appearances of o in one or more documents.
- F is a set of data frames—one for each object set and, as extended here in this proposal, also one for each relationship set. Data frames include recognizers for object and relationship instances as they appear in documents and free-form user specifications, instance converters to and from internal representations, operations over internal representations, and recognizers for operation instantiations as they appear in documents and free-form user queries.

We can, and usually do, think of the triple (O, R, C) as an ontology.[3] Among other representations,[4] we use OWL as our main representation language for ontologies. And among other representations,[5] we use RDF as our main representation language for storing instances with respect to OWL ontologies. Because a KB is a populated ontology, it is a database and thus can be queried and mined for knowledge nuggets. Because a KB includes annotations, usually for each object instance, it provides a simple type of provenance—a link back to documents from which object instances are extracted. And because a KB includes data frames for object and relationship sets, it is an extraction ontology—an ontology that can recognize object and relationship instances in structured, semi-structured, and unstructured documents.

Figure 4: Annotated Lung Cancer Image.

A *KB-Builder* (*KBB*) is a tool used to build KBs—more specifically, it is a tool to largely automate the building of KBs. The KBB has the capability to fully and automatically build a KB by reverse-engineering structured and semi-structured information sources into KBs. Perhaps more often than not, however, users need custom-built KBs. Using the KBB, a user U can start from scratch and build a KB from the ground up. As U begins to build a KB, the KBB watches and learns what U wants. Often, after only being shown how to harvest a handful of instances from machine-generated documents, the KBB can harvest and organize instances from hundreds of additional machine-generated, sibling, source documents. Further, as it collects more knowledge into its knowledge bundle, the KBB can create or identify domain-specific instance recognizers and use them to extract knowledge from as-yet unseen, non-sibling, and even non-machine-generated source documents. As a result, the KBB can also do high-precision document filtering to find additional relevant documents for the study. The KBB is synergistic, with as much of the burden shifted to the machine as possible. It allows users to check and fix any mistakes it makes, and it learns to do better as it is corrected.

We note that the KBB we propose is not for everyone's information-finding needs. Although semi-automatic with much of the burden being shouldered by the KBB, the overhead in establishing KBs from scratch may outweigh the advantages gained. Individuals just wanting a quick answer to a question would not, indeed should not, use the KBB to build a KB from scratch to answer their question. On the other hand, it is not hard to envision a web of thousands of KBs

[3] Pundits disagree about the definition of an ontology, but we adopt the view that an ontology is a formal theory and a specification of a shared conceptualization, both of which can be captured in a model-theoretic view of data within a formalized conceptual model. Since the elaboration of our triple (O, R, C) is a predicate-calculus-based formalized conceptual model [EKW92], we present it as an ontology.

[4] XML Schema and OSM [EKW92]

[5] XML and OSMX—populated OSM model instances represented in XML

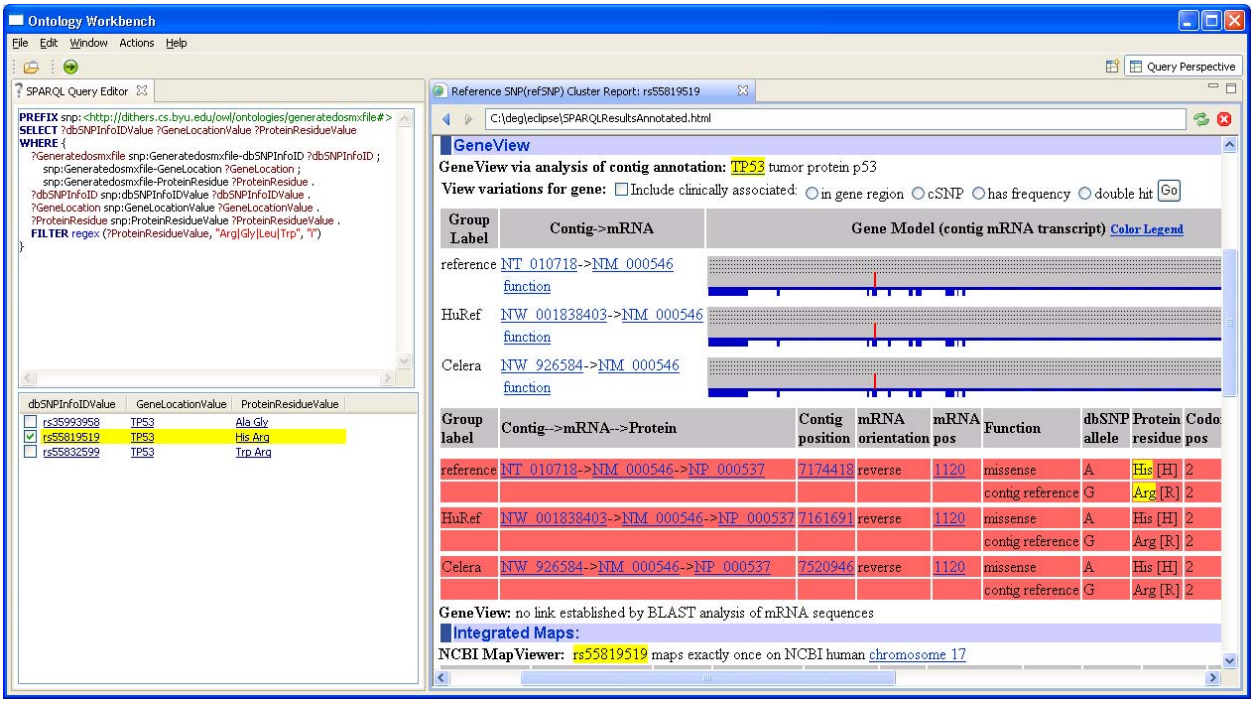

Figure 5: Screenshot of our Current Web of Knowledge Prototype System.

already built, interlinked on identical data items, and publicly available. In this case, the web of KBs would directly support question answering—returning answers to questions and links to pages to verify answers. Indeed, as we have explained elsewhere [TEL09a], users can successfully pose free-form questions in an interface to a web of KBs. We further note that when building a KB is appropriate, none of the overhead is wasted. Users simply start harvesting and organizing the information they need under the "watchful eye" of the KBB, which takes on ever more of the task.

We now explain how the proposed KBB works. We begin by describing work already accomplished, but which we must solidify, enhance, piece together, and scale up.

- *Form-based Ontology Creation*. While we do not assume that bio-researchers and other decision-making researchers are expert users of ontology languages, we do assume that they can create ordinary forms for information gathering. The KBB interface lets users create forms by adding various form elements: single-entry and multiple-entry form fields, as well as coordinated columns of multiple-entry form fields and form fields indicating concept generalization / specialization and component aggregation. The form-building interface also lets users nest form elements as deeply as they wish. The clickable icons in the data and label fields of the forms in Figures 1 and 3 let users control form creation. Users can specify any and all concepts needed for a study and can specify relationships and constraints among the concepts. Thus, users can customize and organize their data any way they wish.

  From a form specification, the KBB generates a formal ontological structure, (O, R, C). Each label in a form becomes a concept of O. The form layout determines the sets of relationships in R among the concepts, and the constraints in C over the concepts and relationship sets. For example, between the form-title concept *T* and each top-level single-entry form element *S*, the KBB generates a functional binary relationship set from *T* to *S*. Thus, for the form in Figure 1, the KBB generates functional relationship sets from *Single Nucleotide Polymorphism*

to *SNP Annotation*, to *Alleles*, and to *Nomenclature* (which, in Figure 1, is scrolled off the bottom of the panel). Similarly, between each form element E and a single-entry form element S nested inside E, the KBB also generates a functional binary relationship set from E to S. Thus, among others, for the form in Figure 1, the KBB generates a functional relationship set from *SNP Annotation* to *Chromosome Location*.

- *Information Harvesting.* How well the KBB harvests information from a particular site depends on how regular the pages are. Most pages are uniform enough that the KBB can harvest information without user intervention.[6] For each item to be harvested from an HTML page, the KBB generates an xpath to the node in the DOM tree in which the item is located. For data items within the node, the KBB automatically infers the left and right context information it needs as well as delimiter information for list patterns. When the pages are not as uniform as might be expected, the KBB works interactively with a user U, allowing U to cut and paste any data items missed or mistakenly entered in its automated harvesting mode. The KBB learns from these corrections and makes adjustments as it continues to harvest from multiple pages in the site. Details about how the KBB infers context and delimiters and about how the KBB operates when it is corrected are contained in [TEL09a, TEL09b].

  While harvesting information, the KBB builds the I and A components of a KB. Since the KBB harvests concept value instances and relationship instances with respect to the defined ontology, it is able to immediately populate the ontology with these harvested values and thus build the I component of the KB. Constructing the A component is a matter of keeping links to the pages and location within the pages from which the value instances are extracted. The KBB records the xpath to the node in which the value appears and the offset within the node. Because web pages tend to change, we cache pages when we harvest them. This ensures that provenance links remain valid. It also means, however, that the KBB may need to reharvest information from the page whenever its harvested content changes.

  Our annotation implementation allows us not only to annotate simple instance text values and phrases, but also to annotate parts of values or to annotate values in parts. For example, a bio-researcher may want to view start and end positions of genes on chromosomes as a single value while the web page displays two separate values or vice versa. Our implementation also allows us to annotate images. Users can select either the entire image as the unit of annotation or any subpart of the image. Thus, a bio-researcher can directly annotate the cancerous part of the lung in Figure 4.

- *Extraction-Ontology Creation and Usage.* Building the F component of a KB turns the populated ontology into an extraction ontology. Ontologies augmented with instance recognizers are called *extraction ontologies*. Instance recognizers, contained in *data frames*, are regular expressions that recognize common textual items such as dates, email addresses, and DNA sequences. They can also contain lexicons that match with items such as company names and protein names. Much can and has been said about data frames and instance recognizers embedded within extraction ontologies (e.g., [ECJ+99]).

  To build an extraction ontology, and thus the F component of a KB, the KBB needs to be able to create instance recognizers. The KBB creates instance recognizers in two ways as it harvests information: (1) by creating lexicons and (2) by identifying and specializing data frames in a data-frame library. Lexicons are simply lists of identifiable entities—e.g., lists of protein names. As the KBB harvests protein names or any other named entity, it simply

[6] In preliminary evaluations we have conducted, the system has often been able to achieve 100% precision and recall.

stores the unique names in a lexicon. Thus, when the KBB encounters the name again, it can recognize and classify it. We initialize a data-frame library with data frames for common items we expect to encounter. The library contains, among many others, data frames with recognizers for all types of numbers and dates. For bio-researchers we would want to add data frames to recognize DNA sequences, nucleotides, amino acids, and other data instances with regular patterns. When recognizers in these data frames recognize harvested items, they can classify the items with respect to these data frames and associate the data frames with concepts in the ontology. For special cases, such as numbers with units, the KBB should be able to classify the numbers (type and range) and find the units and thus should be able to create new data frames by specialization. We have had some implementation experience building data frames [TE09], but exploring additional possibilities for the KBB to automatically construct data frames is future work.

- *Reverse-Engineering Structured Data to KBB Forms.* Structured repositories (e.g., relational databases, OWL/RDF triple stores, XML document repositories, HTML tables) may contain much of the information needed for a bio-study. A reverse-engineering process can turn these structured repositories into knowledge bundles. Figure 3 shows an example of a generated KB form resulting from a reverse-engineering process. Bio-researchers can use the generated forms in the KBB to custom-tailor reverse-engineered KBs, restructuring the metadata to the (O, R, C)-ontologies they want and limiting the data to the I-values they want. They can even generate a database snapshot by having the KBB fill in the form with the I-values they have selected and annotate these documents, and thus produce A-annotations for their KB. And, they can employ the same techniques mentioned in the previous bullet to produce F-component data frames for extraction ontologies.

  We have had some implementation experience reverse engineering HTML tables, specifically sibling tables from machine-generated web sites, into extraction ontologies [TE09]. We have also implemented reverse-engineering for relational databases [EX97] and XML document repositories [AKEL08] into conceptual-model instances in OSM, and we are currently implementing reverse-engineering for OWL/RDF. As part of the the reverse-engineering of XML documents into OSM, we have identified and proven properties about an XML normal form that ensure the generation of redundancy-free XML-document structures [ME06]. Because of the structural isomorphism between KBB form structures and and XML-schema structures, these properties guarantee that we can generate generate KBB forms, however complex and however deeply nested, such that they are redundancy free.

- *KB Usage for Analysis and Decision Making.* As the KBB harvests information, it stores harvested information in its KB, which is a knowledge base.[7] Users can query and mine the KB in standard ways. Query results, however, have an additional feature beyond standard query results—each result data value v is "clickable" yielding the page with v highlighted as Figure 5 shows.

  We have implemented an initial prototype as part of our web-of-knowledge project [ELL+08, TEL09a]. Currently, users can pose queries over RDF triple stores using the SPARQL query language and a version of our extraction-ontology-based free-form query processor. Figure 5 is a screenshot of our prototype.

[7] Note that KB, the acronym for a knowledge bundle, is the same as the acronym for a knowledge base. This is intentional—a knowledge bundle includes a knowledge base, usually however, a personalized knowledge base for some specific research agenda.

Although some of our work is complete, we still have much to do to solidify and enhance what we have already implemented and to extend it to be a viable bio-research tool. We plan to further our research as follows. Our objective is to shift as much of the burden as possible to the KBB—definitely a non-trivial research task.

- Much of what we have accomplished has been done as part of the work of master's theses and doctoral dissertations. We have begun to piece together these student projects into a unified prototype, but given the diversity of project tools and their independent nature, integrating them together into a unified whole is challenging.

- We have defined and implemented data frames for concepts corresponding to nouns and adjectives. To accommodate relationships explicitly, rather than implicitly as we do now, we plan to define data frames for relationships in connection with verbs. In addition to these atomic structures, we plan to define molecular-size ontology snippets that include a small number of interconnected data frames that together could recognize meaningful aggregates of atomic concepts. Further, although we have worked some with high-precision filtering [XE08], we have not yet applied our work to literature repositories. Extending extraction ontologies with data frames for relationships and molecular-size ontology snippets should help, but we also believe that to be as successful as might be required for bio-researchers, we will likely also need to adapt techniques from natural language processing and probabilistic grammars.

- We have not yet added field constraints to forms. Adding these constraints makes our form-based harvester act as a filter. More generally, form harvesters with constraints behave as database queries. We have earlier defined and implemented NFQL (a Natural Forms Query Language) [Emb89], which we intend to use as a foundation for adding constraints to forms.

- Reverse engineering structured data is a field of research unto itself. Our efforts have proven to be successful [AKEL08, EX97, ME06, TE09]. We point out, however, that they need not be perfect since we expect bio-researchers to customize generated forms for their own use. Also, we observe that bio-researchers may often only want small parts of large structured repositories. Automatically finding and cutting out these small parts is possible, but challenging [LED$^+$09]. Improving on previous work will be necessary to help bio-researchers successfully harvest just the desired information from structured sources.

- Schema integration, record linkage, data cleaning, and data fusion are each, by themselves, fields of research. Fortunately, we need not resolve all, or even any, of the issues in each of these fields of research to be successful. However, opportunities abound if we can provide some basic data integration within KBs and some basic data linkage among KBs. Data integration within KBs can reduce uncertainty and mitigate the need for users to assimilate data from different sources manually. Data linkage among KBs can lead to opportunities for cross KB data mining and for serendipity by presenting bio-researchers with (perhaps unexpected) connections among research studies. We have considerable experience with schema integration (e.g., [BE03, XE06]), which we intend to adapt for use in our KBB.

- Automatic classification and annotation of multimedia objects is another field of research unto itself. Our particular interest in multimedia objects is merely to provide a way to collect them and manually annotate them. Our annotation schema currently allows users to collect images as objects in object sets and to record simple annotations about these objects.

Currently, we only annotate entire images or rectangular subimages. We plan to add additional facilities for annotating images such as pointer icons and fine-grained polygonal boundaries, even more fine-grained than the polygonal boundary in Figure 4. Other than adapting available tools (e.g., intelligent scissors developed for Adobe at BYU), we plan to do no research in multimedia.

The field of information extraction is at the heart of our proposal. For more than a decade researchers have proposed various ways and means to do information extraction. Recently, several major overview papers and research monographs have appeared [DGM03, EBSW08, Sar08, WKRS09]. In relation to this work, we merely claim that none of the approaches, with the possible exception of [WKRS09] claims, or even aims at, the kind of high-precision information extraction we propose here. And, although high-precision in its results, the approach in [WKRS09] is not directed toward harvesting custom KBs of the type needed by bio-researchers.

Our KB/KBB approach proposed here is unconventional: it supports directed, custom harvesting of high-precision technical information. And our KB/KBB approach is innovative: its semi-automatic mode of operation largely shifts the burden for information harvesting to the machine, and its synergistic mode of operation allows research users to do their work without intrusive overhead. Our KB/KBB approach is a helpful assistant that "learns as it goes" and "improves with experience."

3.3 Evaluation

We intend to evaluate our proposed KBB in two fundamental ways: (1) precision and recall on KBB tools and (2) field tests.

We can use precision and recall results to assess the accuracy of automatic harvesting. Once a page has been marked up, how well can the KBB harvester recognize and extract similar data from subsequent pages? To a large extent these precision and recall measures will be more a measure of page regularity than a measure of the KBB to harvest accurately. More interesting is to determine whether the KBB can recognize when it needs assistance from its human operator. We can also use precision and recall results to assess the accuracy of initial form fill-in by an extraction ontology. Further, since the evolving extraction ontology should improve, we can run regression tests to ensure that it does not "regress" as it "improves." Finally, we can use precision and recall, as initially intended by the information-retrieval community to determine whether the literature-filtering processes of the KBB can identifying relevant literature and reject close-but-irrelevant literature.

For field tests, we intend to have bio-researchers[8] use the KBB to create KBs for actual research scenarios. If sufficient backend analysis tools (e.g., statistical packages that run on RDF data repositories) are available or can easily be adapted for use with RDF, bio-researchers may be able to use the KBB in their actual research work.

4 Timeline and Milestones

The code to all of our tools will be open-source. Initially, we will publish the code on our own servers, but as more outside people become involved and as the project continues to develop,

[8]Yan Asmann, a bio-researcher at Mayo Clinic, has agreed to field-test KBB. She is on the team of PI's specifically for this reason, as well as to be a consultant representing bio-researchers and to provide us with real-world test-case scenarios as we develop KBB.

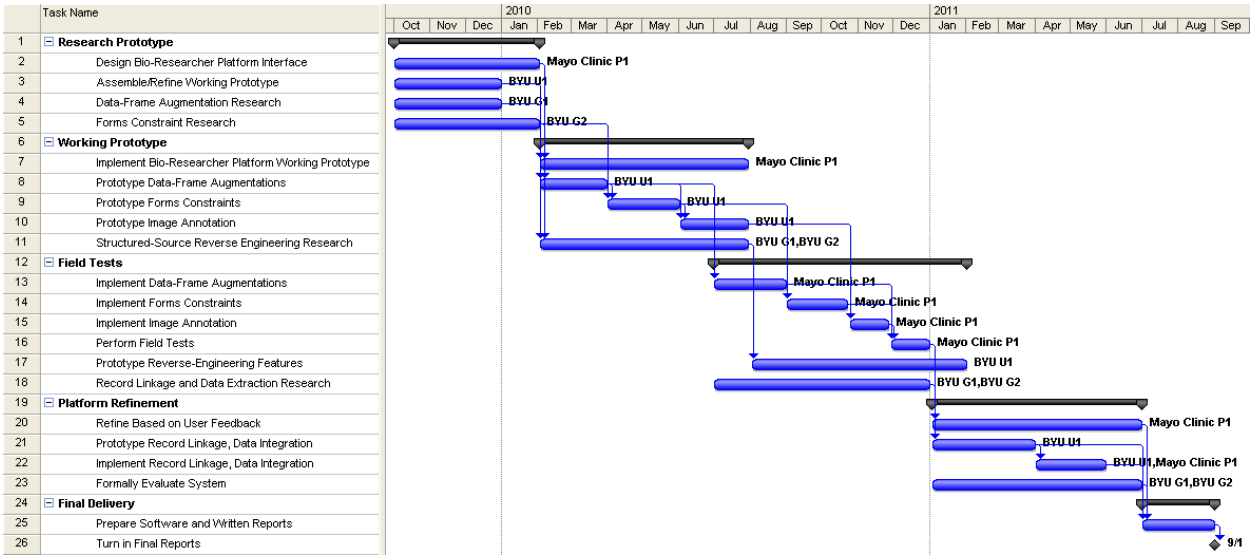

Figure 6: Project Plan Gantt Chart.

we will move the entire code base to a public repository such as SourceForge. The development tools and libraries we use include standard open-source components: Java, XML, PHP, HTML, CSS, JavaScript, and Eclipse (including the Eclipse graphical framework as the foundation for our workbench). We use an agile software development methodology (Scrum, with two-week sprints).

Personnel involved in the development include: three faculty, two graduate-student researchers, one undergraduate programmer, all at BYU; and two researchers and one programmer at Mayo Clinic. The Gantt chart in Figure 6 shows the schedule for the development team. We have divided the project into five milestone deliverables as identified in Figure 6. By year-end 2009, we will have integrated our existing tools into a research-quality prototype, and we will have a design in place for the working prototype of the envisioned bio-research KB / KBB platform. By mid-year 2010, the basic working prototype will be in place and ready for field testing. By year-end 2010, additional features will be in the system and field testing will have begun in earnest. We plan to refine the system during the first half of 2011, with final delivery ready for September 2011.

The pipeline strategy throughout this process is first to perform research on additional techniques or features needed, then prototype them at a research level, and next implement them to a level of "production quality." We will refine according to user feedback at any point where such feedback has been gathered.

Embley will direct the overall project. Lonsdale will supervise work on high-precision filtering of plain-text literature and natural language text in semi-structured documents. Liddle will supervise software development. Tao will be the BYU / Mayo-Clinic liaison and will supervise the Mayo Clinic programmer. Asmann will serve as bio-researcher and field tester. All will participate as authors in the various research papers produced during the course of the project.

References

[AKEL08] R. Al-Kamha, D.W. Embley, and S.W. Liddle. Foundational data modeling and schema transformations for XML data engineering. In *Proceedings of the 2nd International United Information Systems Conferences (UNISCON'08)*, pages 25–36, Klagenfurt, Austria, April 2008.

[BE03] J. Biskup and D.W. Embley. Extracting information from heterogeneous information sources using ontologically specified target views. *Information Systems*, 28(3):169–212, 2003.

[BFS06] J. Bonis, L.T. Furlong, and F. Sanz. OSIRIS: A tool for retrieving literature about sequence variants? *Bioinformatics*, 22(20):2567–2569, 2006.

[CJR+07] J.G. Caporaso, W.A. Baumgartner Jr., D.A. Randolph, K.B. Cohen, and L. Hunter. MutationFinder: A high-performance system for extracting point mutation mentions from text? *Bioinformatics*, 23(14):1862–1865, 2007.

[DGM03] J. Davies, M. Grobelnik, and D. Mladenić, editors. *Semantic Knowledge Management: Integrating Ontology Management, Knowledge Discovery, and Human Language Technologies*. Springer, Berlin, Germany, 2003.

[EBSW08] O. Etzioni, M. Banko, S. Soderland, and D.S. Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, 2008.

[ECJ+99] D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith. Conceptual-model-based data extraction from multiple-record web pages. *Data & Knowledge Engineering*, 31(3):227–251, November 1999.

[EKW92] D.W. Embley, B.D. Kurtz, and S.N. Woodfield. *Object-oriented Systems Analysis: A Model-Driven Approach*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[ELL+08] D.W. Embley, S.W. Liddle, E. Lonsdale, G. Nagy, Y. Tijerino, R. Clawson, J. Crabtree, Y. Ding, P. Jha, Z. Lian, S. Lynn, R.K. Padmanabhan, J. Peters, C. Tao, R. Watts, C. Woodbury, and A. Zitzelberger. A conceptual-model-based computational alembic for a web of knowledge. In *Proceedings of the 27th International Conference on Conceptual Modeling*, pages 532–533, Barcelona, Spain, October 2008.

[Emb89] D.W. Embley. NFQL: The natural forms query language. *ACM Transactions on Database Systems*, 14(2):168–211, June 1989.

[EX97] D.W. Embley and M. Xu. Relational database reverse engineering: A model-centric, transformational, interactive approach formalized in model theory. In *DEXA'97 Workshop Proceedings*, pages 372–377, Toulouse, France, September 1997.

[LED+09] D.W. Lonsdale, D.W. Embley, Y. Ding, L. Xu, and M. Hepp. Reusing ontologies and language components for ontology generation. *Data & Knowledge Engineering*, 2009. (in press).

[ME06] W.Y. Mok and D.W. Embley. Generating compact redundancy-free XML documents from concptual-model hypergraphs. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1082–1096, August 2006.

[Sar08] S. Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.

[TE09] C. Tao and D.W. Embley. Automatic hidden-web table interpretation, conceptualization, and semantic annotation. *Data & Knowledge Engineering*, 2009. (in press).

[TEL09a] C. Tao, D.W. Embley, and S.W. Liddle. Enabling a web of knowledge. Technical report, Brigham Young University, 2009. (submitted for publication—draft manuscript available at deg.byu.edu).

[TEL09b] C. Tao, D.W. Embley, and S.W. Liddle. FOCIH: Form-based ontology creation and information harvesting. Technical report, Brigham Young University, 2009. (submitted for publication—draft manuscript available at deg.byu.edu).

[WKRS09] G. Weikum, G. Kasneci, M. Ramanath, and F. Suchanek. Database and information-retrieval methods for knowledge discovery. *Communications of the ACM*, 52(4):56–64, April 2009.

[XE06] L. Xu and D.W. Embley. A composite approach to automating direct and indirect schema mappings. *Information Systems*, 31(8):697–732, December 2006.

[XE08] L. Xu and D.W. Embley. Categorization of web documents using extraction ontologies. *International Journal of Metadata, Semantics and Ontologies*, 3(1):3–20, 2008.