

Semantic Data Integration Environment for Biomedical Research (Ontological Framework for e-Science & Imaginizer)

V. Astakhov, B. Sanders, Jeffrey S. Grethe, E. Ross, D. Little, A. Gupta, T. Astakhova.

BIRN Coordinating Center, University of California San Diego, La Jolla, CA 92093-0715, USA
Contact: {astakhov@ncmir.ucsd.edu}

Abstract

This paper presents current progress in the development of Ontological Framework on top of the data integration environment which is a part of the Biomedical Informatics Research Network (BIRN; <http://www.nbirn.net>) project which is sponsored by the National Institutes of Health (NIH). The goal is the development of a ontological framework for biomedical research that support advance data acquisition, data storage, data integration, data visualization and other computing and information processing services over the Internet. Ontologically enabled data integration system performs semantic integration of data sources distributed over multiple research centers. It enables researchers to perform analyses based on larger and broader semantically relevant datasets than would be available from any single institution's data. This paper describes the system architecture, capabilities of the semantically based data integration and a new way to animate semantics of texts. Also, it demonstrates a way to represent ontology through holographic reduced representation (HRR) and describe ontological confabulation as a way to optimize semantics browsing.

Introduction

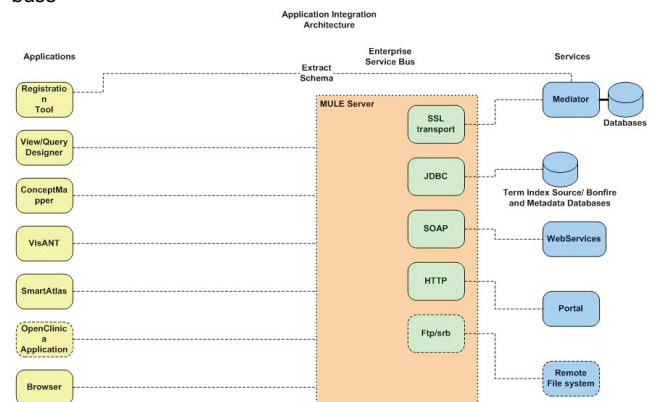
We are developing an Ontological Framework on top of data integration environment (mediator) at the Biomedical Informatics Research Network (BIRN) that fosters large-scale collaborations in biomedical science by utilizing the capabilities of the emerging cyber infrastructure such as high-speed networks, distributed high-performance computing and the necessary software and data integration capabilities [1]. Currently, this is a consortium of more than 40 research groups over US and UK participating in three test bed project centered on brain imaging and genetics of human neurological disease. These groups are working on large scale, cross-institutional studies on Alzheimer's disease, Autism, depression, and schizophrenia. Others are studying animal models relevant to multiple sclerosis, attention deficit disorder, and Parkinson's disease through MRI, whole brain histology, and high-resolution light and electron microscopy. There are a number of integrated schemas over different combinations of these sources designed for different study groups. The data to be integrated ranges from 3D volumetric data of nerve components, to image feature data of protein distribution in the brain, to genomic data that characterize the anatomical anomalies of

different genetically engineered mouse strains. The goal of the BIRN application integration team is to develop a general-purpose Ontological data integration framework that neuroscientists can use in their research.

Architecture

Ontological Framework interact with the data integration environment. It provide semantic interface for underlying infrastructure. The BIRN's data integration environment provides set of tools and system infrastructure for all phases of the data integration process from registration of sources to semantic queries [2,3,4,5,6]. This infrastructure consists of an API for data registration and ontological annotation, a data integration server and a number of domain-specific clients which provides the necessary semantic information, tied to external ontologies, so that integrated queries can be executed across the sources. Currently, we are rewiring system components to make them accessible through Enterprise Service Bus (ESB) architecture. Figure 1 illustrates ESB server which provide multi-protocol communication bus over all services.

Figure 1: All applications connected through enterprise Service buss



One major service is Mediator which provides integration over relational data bases. Figure 2 shows major components of the mediator service.

Gateway: Mediator entry point is gateway which handles all communication with Mediator clients. The Gate-

way's principal functions are to: listen on a secure socket for a client XML request; authorize, validate and classify the request; process the request; and return an XML response.

Access control is a very important aspect of a practical information integration system. Mediator allows user credentials to be passed to the sources for authentication and authorization.

Registry: The Registry is the Mediator's internal database of data types, sources, relations, foreign-key relationships, functions, integrated views, users, permission policies, and related data. The relations and functions are source-specific and reflect the portion of a BIRN-participating institution's research data they are sharing with the BIRN.

Planner: The planner's purpose is to take a query and produce a logical query plan which tells in what order to gather data from the different sources, which data to transfer between what sources, and from which relations or functions to gather the mentioned data. To produce a plan, for a particular query, the first action of the planner is to construct a tree representation of that query. This tree has, as leaves, all base relations and functions of the query, and, as internal nodes, all the Boolean operators that relate these base components to one another. The next step involves transforming the tree into its equivalent disjunctive normal form (DNF). The result is a two level tree representing a disjunction of conjunctive queries. Finding a feasible plan for the original query reduces to finding properly ordered plans for all the constituent conjunctive queries of the DNF. The algorithm iterate through a list of all elements such as relations and functions found in a conjunctive query. As soon as a suitable element is found, it is removed from the list and placed in the next position of the prospective plan; whereupon the process repeats itself with the now smaller list. The algorithm stops when no more elements can be placed in the plan, either because none remain in the list, or because none of the remaining elements is suitable for placement in the plan. This algorithm runs in time $O(n^2)$, where n is the number of elements in the conjunctive query.

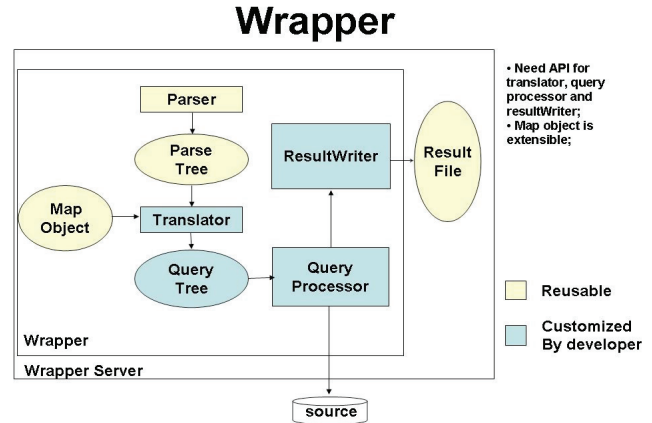
Executor: The execution engine translates the object that it receives from the Planner into the specific set of instructions (XML) and sends it to the wrapper server. The wrapper will convert this XML message to the appropriate language of the source, execute the query at the source and returns the results back.

The results retrieved from the wrapper are stored temporarily in the execution engine database (MySQL) to be used following source conjunctions in the plan. At the end of the execution, the results from each disjunction in the plan are unioned with each other and are sent back to the client.

Wrapper: The Wrapper web service is the middleman between the mediator and a source. It accepts the query from the mediator, translates it into source query language,

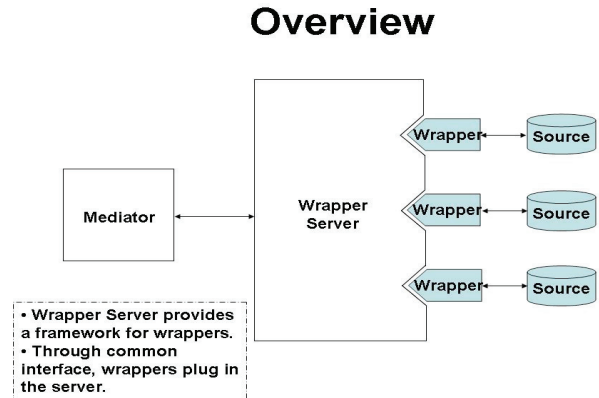
executes it against the source, and finally returns the results back to the Mediator in the Mediator's format.

Figure 2: Wrapper web service



Wrapper Service: The Wrapper server is standalone software which acts as a container of wrapper instances. Each source has its own dedicated wrapper instance.

Figure 3: Wrapper Server based on Tomcat/Axis freeware.



The server accepts three kinds of requests: 1) Requests to add or remove a wrapper instance. 2) Requests to execute a query and 3) Requests to retrieve results.

And we have designed source-specific wrappers for sources such as Oracle, Oracle Spatial, and PostgreSQL, where a generic query can be translated into the appropriate flavor of SQL and functions supported by the specific sources' systems.

Ontological Framework

Ontological Framework utilizes underlying architecture and built on top of Data Integration environment. It employs the model where each data source treated as a relational source. For a data source which disclose a set of functions those functions are internally treated as relations with the binding pattern (\bar{b}, f) where \bar{b} represents a set of bound arguments and the single f is the free output variable of the

function. Using this model also enables us to treat computational sources such as statistical packages as a “data sources” which contributes functions but no relations. Every relations and attributes has a descriptor for keyword search, and a so-called semantic-type that can be used to map the element to an ontology.

Ontology: Ontology is a term-graph whose nodes represent terms from a domain-specific vocabulary, and edges represent relations that also come from an interpreted vocabulary. Many of the terms are drawn from existing terminologies and ontologies such as the Unified Medical Language System (UMLS), the Foundational Model of Anatomy (FMA), Neuronames, the Gene Ontology, the Functional Genomics Ontology (FUGO) and the Phenotype and Trait Ontology (PATO). These terms will form the basis of a more formal ontology describing the multiscale investigation of neurological disease. The BIRN Lex provides terms, utilized by BIRN scientists in the context of their research, covering neuroanatomy, molecular species, behavioral and cognitive processes, subject information, experimental practice and design, and associated elements of primary data provenance required for large-scale data integration across disparate experimental studies.

The nodes and edges are typed according to a simple, commonly agreed upon set of types produced by test bed scientists. The most common interpretation is given by rules such as the transitivity of *is-a* or *has-a* relations, which can be used to implement inheritance and composition. However, there are also domain specific rules for relationships such as *volumetric-subpart* (brain-region → brain-region) and *measured-by* (psych-parameter → cognitive-test) that need special rules of inference. For example, if a brain-region participates-in a brain-function (such as “working memory”), and the brain-function is measured-by a cognitive-test, then the cognitive-test functionally-tests the brain-region.

In the current framework, ontologies are represented as a set of relations comprised of a set of nodes and a set of edges, with appropriate attributes for each node and edge. Other operations, including graph functions such as path and descendant finding and inference functions like finding transitive edges are implemented using an API of functions. BIRN ontological browser is accessible from the web site <http://mediator.nbirn.net:8080/knowme/bonfire.html>

Data to Ontology annotation: The ontology is described using OWL (Ontology Web Language) which is built on top of RDF (Resource Description Framework), which can be edited using a tool such as Protégé. An OWL ontology is constructed with a hierarchical vocabulary of terms to describe concepts in blended space. Each concept in the ontology maps to a Data Object, which is a set of fields and values stored in a database. There is also the capability to retrieve related data fields via sequences of foreign key/primary keys and map those field values. The full set of concept-data object mappings is saved to a pro-

ject which can be loaded into a database. It is then possible to perform a join query using concept terms to retrieve all relevant data stored in the database as a result set. This method allows ontological concepts to be used as a generalized query vocabulary for database information retrieval. To provide data annotation using pre-defined ontology, the system has additional *mapping relations*. Currently there are three kinds of mapping relations.

The *ontology-map* relation: that maps data-values from a source to a term of a recognized ontology such as BIRN-Lex, used within BIRN for annotation data sources.

A *joinable* relation: links attributes from different relations if their data types and semantic types match.

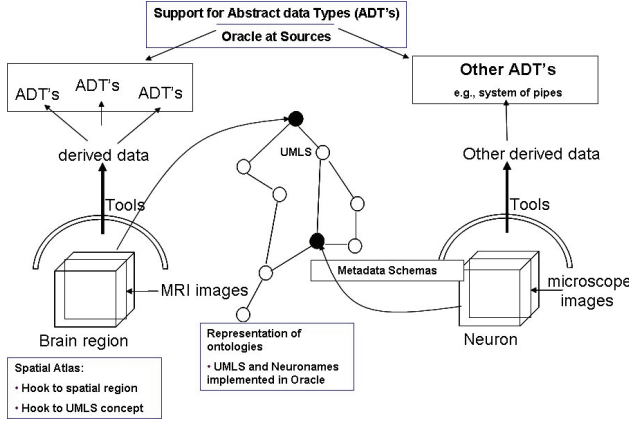
The *value-map* relation: maps a Mediator-supported data value or a Mediator-supported attribute-value pair to the equivalent attribute-value pair disclosed by the source. For example, the Mediator may disclose a demographic attribute called “gender” with values {male, female}, while a source may refer to “sex” with values {0, 1}, while another source, with its own encoding scheme may call it “kcr_s57” with values {m, f}. We have a preprocessor module that uses a look-up function to make a substitution.

Data Annotation: To support mapping to ontology, semantically based integration and the ability to query through ontological concepts, we have developed several architectural extensions. These extensions include a Data annotation tool which helps in generating a map between the data and ontology, a Term-Index-Source that keeps this mapping information, and query pre-processor and post-processor modules.

The data integration environment has a specific relational source called the *term-index-source* that maintains a mapping between data from registered sources and ontological terms defined by various ontologies. This is used by the system to preprocess queries. We have also developed a *Concept Mapping* client to assist in the creation of mapping entries in the *term-index-source* at the time of source registration. This tool allows the user to select ontology such as BIRN Lex [12-13], browse for relevant concept term identifiers, and associate them with *data objects*. The client also connects to an arbitrary relational database source using JDBC, and by retrieving the schema metadata, makes it possible for the user to select one or many table-field-value triples to comprise the *data object*. The *data object* is expanded to include related data via primary-foreign key relationships inherent in the metadata. There are also other automation features such as mapping templates to facilitate the mapping of large sources. The result of mapping the ontology to the registered source is a controlled query vocabulary, incorporating ontological terms, that is made available immediately to all search clients that make use of the *term-index-source*.

Figure 4 demonstrates how data mapped to two different ontological concepts can be linked through the shortest path.

Figure 4: Data to ontology mapping process. Data can be linked through



This Figure also demonstrates that two concepts can be connected through various paths.

We have implemented an algorithm to support relevant data retrieval using ontology concept to concept paths, ontology to data mapping, PK-FK constraints, and ontological data joins that are based on concept ids rather than actual data.

Automatic query generation algorithm

At the first step, a user is expected to provide a list of key words defining areas of interests. The Term Index Source (TIS) engine takes the list and performs a key word search over pre-defined ontological sources and returns a list of ontological concepts. The user reviews the concept list and picks up sub-set that reflect the areas of the interest.

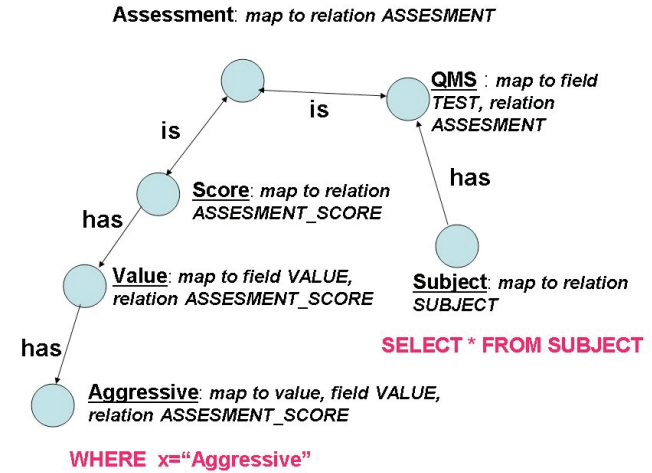
Next, the TIS engine uses this sub-set of terms to extract a list of sources and relations from the maps stored in the TIS. Sources are ranked based on concepts mapped to the source data. Sources which have data mapped for each concept in the list will be ranked as top candidates. For each of these candidate sources, the engine extracts a list of mapped tables/fields. The algorithm starts from fields mapped to the concepts and creates a minimal spinning tree constructed to generate a source local view based on PK-FK constraints. For each source s in these sources:

For each pair of relations that are mapped in the *term-index-source* $\langle u, v \rangle$ in s we compute the path of the foreign key-primary key between them from the database constraints: if such a path exists, then for each relation-column in the fk-pk path we add an edge to the directed graph g , then compute the shortest path sp on g for $\langle u, v \rangle$ using the Dijkstra algorithm, create a view v representing the join across each relation-column in sp , and map the view to the concept ID associated with s . Such views assemble all data relevant to provided ontological concepts. The PK-FK constraints are used to join mapped tables directly or through intermediate tables. Intermediate tables are not included in the final view.

Lastly, the user can request data from individual views generated at each candidate source or it can request data to

be returned from a join across all those views. This join is done based on the ontological markup rather than string matching. For example, the value “subject” may be joined with “aggressive” if they are both marked as being the same concept id.

Figure 5 Ontological query that retrieve and link distributed data annotated to two different concepts: Subject and Aggressive



Concept Based Queries for the Researcher

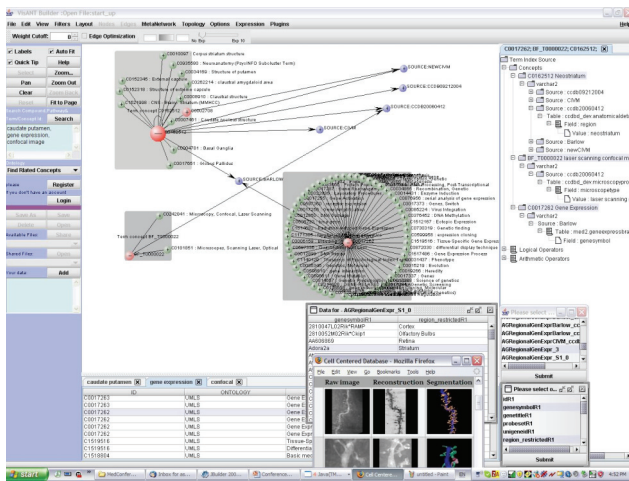
The BIRN portal (www.nbirn.net) provides set of tools and web services for end-to-end functionality. That is the ability to register a new source, annotate it and issue semantic queries against the source.

The VisANTBuilder is one of the system client developed as a discovery tool for the researcher through the use of concept based keyword searches and a graphical display of ontological concepts and data sources that contain data relevant to these concepts. A user types a comma-separated list of terms in the text field and clicks the “Graph Term-Related Concepts” button. A tabbed pane of concept ids appears under the “Terms” tab, and the concept relationships are simultaneously graphed in the “Graph pane.” The graph displayed as a result to the user’s concept query shows the inter-relationship of related concepts in the ontology and the sources where there is data mapped for the specific concept. **Red** nodes are mapped to relations and fields in the Term Index Source and the size of the nodes is relative to number of mapped sources (for example “brain regions” is larger due to multiple site mappings to that concept) Green nodes represent related concepts, which are not mapped to data fields in the Term Index Source. **Blue** nodes represent mapped mediator database sources in the Term Index Source.

After selecting the mapped concepts for the query, a right mouse click brings up a contextual menu with node-related items. The “query selected” item will query the Term Index Source for data sources, fields, and data types which are mapped for the selected concept(s). Selecting the Term Index Source tab will display a tree view of all selected

concepts and allow the user to drill down to the data type, source, table, and field where relevant data can be queried via the mediator. Finally, the “Query Mediator for Table...” menu item returns a result set of all data the selected table. If multiple concepts are selected, an automatic inter-source “smart” join feature will merge data from the sources mapped to those concepts and display the results in a table. Additionally, if the data has the ability to be displayed by the original source it can be rendered to a web browser via URLs (e.g. data from the Cell-Centered Database (CCDB) can be displayed this way). This is critical when returning data to researchers as many sources provide rich user interfaces with a lot of contextual information that a data integration system can not replicate.

Figure 6: “VisAntBuilder” GUI. Provide an ability to see an ontological sub-graph and data associated with selected ontological terms.



Text Animation (Imagenizer)

Many psychological and brain related cognitive studies produce their results in form of assessment texts which require additional processing. To accommodate a huge amount of text information, we developed an extension for VisANTBuilder which called *Imagenizer*. We extended the concept of data visualization and developed experimental environment for *data animation*. That implies animation of short texts collected as psychological reports. We speculate that text animation can provide a quick way to visualize semantics provided by textual information.

To construct an algorithm, fundamental theoretical units must be chosen. We propose the term ontological concept² for an object occupying space and time, an object with attributes specifying what an object is or does and what relations exist between objects. Example: concepts “woman”, “walk”, “beach” can lead to the conceptualization “A woman walked on the beach” that will lead to an animated image of a woman walking on the ocean beach. This conceptualization will imply many “beliefs”. One possible belief here can be - “the woman wears something”. Many of us intuitively “believe” that people usu-

ally wear something when they “walk” if the opposite is not mentioned. That “belief” will cause the imagination of many people to provide an image of the woman walking on the beach and wearing clothing, even if nothing was mentioned about clothing. It is a totally different case where we have the sentence: “A naked person is walking on the beach”.

The following relevant rules can exist in the system:

IF X is like Y then X seeks Y.

IF Y disturbs X then X avoids Y

Universal Structure of objects in the scene

A participant in the scene entity is assumed to be in one of the three states (Active Actor, Passive Actor and Action) with a binding pattern for every disclosed relation. Every entity’s element (relations and attributes) has a descriptor for a keyword search, and a so-called semantic-type that can be used to map the element to its ontology. For example: Relation – behind (far behind) has a type-position / orientation. Another example: Attribute – red has type-color.

Further, an entity may disclose a set of functions that are internally treated as relations with the binding pattern (b, f), where b represents a set of bound arguments and the single f is the free output variable of the function. For example, “take the ball” can be treated as a human specialized function, which is used to raise the human actor hand in a set of specified scenes. Such functions will depend on sets of binding parameters “b” that they characterize, or the position of the ball and return “f”-position of the hand. Such a model lets us treat animation as Petri Net dynamics with computations where actors-nodes take different states in time.

Mental Space: We use a term “mental space” as small packets of concepts, which are constructed as we think and talk. Mental Space can be seen as an ontological sub-graph. Also, we have *Conceptual integration* as a critical part of imagination. It connects input *mental spaces*, projects selectively to a blended imaginary space, and develops emergent structures through composition, competition, and elaboration in the blend.

For example, a set of sentences: “The blue ball was left on the beach. A woman walks on the beach,” imply two input concept spaces “Woman walks on the beach” and “ball was left on the beach”.

We perform *Cross-Space mapping* which connect counterparts in the input mental spaces and then construct *Generic Space* that maps onto each of the inputs and contains what the inputs have in common: beach, ocean, and horizon. The final *blending* does the projection of the ocean beach from the two input mental spaces to the same single beach in blended imagination space.

Such blending develops emergent imaginative structures that are not present in the inputs like “woman walk toward the ball” or “woman walk relatively close to the ball”. It seems intuitively obvious that imagination can create an integrated scene with all the mentioned objects as a result of those two sentences.

Ontological dK-series and dK-graphs

An ontological graph for a dictionary of English words is extremely complex. Here, we need a way to approximate properties of a generic ontological graph, that can be built on a limited text cohort but that can capture topological properties of generic English text.

To capture such complexity of graph properties, we use the dK-graphs approach [14]. This approach demonstrates that properties of almost any complex graph can be approximated by the random graph built by set of dK graphs :0K, 1K, 2K and 3K where “K” is the notation for a node degree and d-for joint degree distribution that d node of degree “k” are connected.

Based on our text cohort, we estimate 0K-average node degrees as the average frequency of a word, 1K –node degree distribution as frequencies for the words in the text cohort, 2K and 3K – joint degree distribution were extracted as pair-wise and triple-wise frequencies of having two/three words in two/three sub-sequent sentences. Those values were assigned for each dK graph of our ontology and later used during operation of *ontological confabulation*.

Term-Object-Map

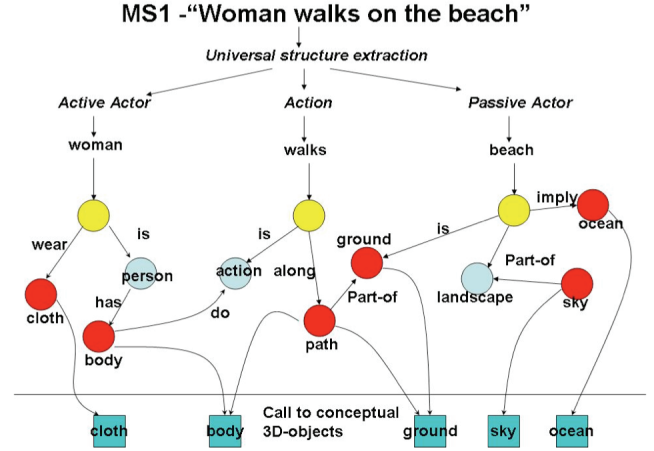
We perform a specific mapping between ontological terms and 3D-animation objects library, which was developed for the Maya animation environment. These objects are used by the system to build animation.

Text filtering

We developed experimental prototype *-Imagenizer* which uses simple English text as an input and generates output animation. First, it performs text processing to extract semantic relations among words in the sentences. Mental space is created for each sentence. As an example, we consider the simple text of three sentences taken from psychological assessment report collected during depression study: “A woman walks on the beach. The blue ball was left on the beach. A woman takes this ball”.

Figure 7 represent details of the first mental space that are built from the sentence “Woman walks on the beach”. The sentence was processed and its *Universal structure* was extracted: “Active actor (woman)-action (walk)-passive actor (beach)”. Instances of the universal structure (woman, walk and beach) were anchored (colored by yellow) to an ontological graph that represents relations among concepts. We perform graph expansion operations to integrate all relevant objects required for the mental space such as ocean, sky and the woman’s clothing, which are not mentioned in the sentence (colored by red). Each concept such as “beach”, “woman”, and “ball” represents a sub-graph that connect all concepts relevant to the specified term.

Figure 7 .Part of ontological graph that represent mental space for sentence “Woman walks on the beach”



Same operations were performed for the second and third sentences.

Generic Space and holographic reduced representation

Generic space was constructed by the extracting of objects such as “woman”, “beach” and “ball”, which co-occurred in different *mental space*.

Starting from those concepts we perform node expansion as a procedure of finding neighbor concepts connected to generic concepts; for example, the “color” of the ball and “body” of the woman. We perform node expansion by using various relations (represented by edges on the ontological graph) such as analogy/disanalogy, cause-effect, representation, identity, part-whole, uniqueness, similarity, and various properties. All these classes of relations are represented by various grammars constrains in the English texts. Follow the ideas of “Universal grammar” and “Distributed Reduced Representation” proposed in papers of Paul Smolensk [15], we define any semantic space as a convolution of some vectors-concepts providing the space reduced representations. Rather than using Smolensky’ Tensor [15] that has the variable length we decided to use the Holographic reduced representation techniques [16]. Each node and edge were assigned to a random vector from 512-dim space then any combination of connected nodes and edges were defined as a result of circular convolution on the vectors [16]: $z = x @ y$, where $z_j = \sum x_k * y_{jk}$.

And indexes represent coordinates in 512 dimensional space. Such representation will provide coding schema to any complex scene.

Example: sub-graph “woman-wear-clothing” was represented as $z = x @ y = x @ (m @ n)$; where “m”-represents vector “woman”, “n”-“wear”, “clothing” and convolution of “m” and “n” gave us “woman-wear-” open-end sub-graph that has one node and one edge.

Renormalization as dimension reduction

Due to the high complexity of the ontological graph we performed a “graph compression” that resulted in elimination of some redundant links. We decided to perform compressions just over the vital relations such as Time, Space, Identity, Role, Cause-Effect, Change, Intentionality, Representations and Attributes. Mathematically, that operation was implemented as circular correlation: $y = x \# z$, where $y_j = \sum x_k * z_{k+j}$. Taking the previous example, that operation is equivalent to “woman-wear-”#”woman-wear-clothing”=”clothing” and will return the node “clothing”.

Holographic reduced representation let us quickly compute “generic space”.

The final ontological sub-graphs blending and generation of resulting imaged space were performed as a confabulation on the *generic space*.

dK- Ontological Confabulation

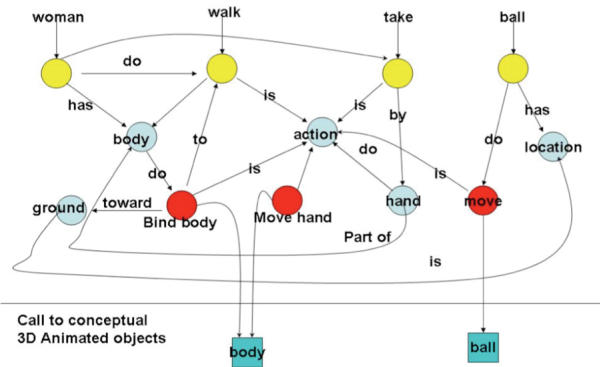
We extended the operation of Confabulation previously proposed in paper [17] and developed an extended version of this operation for ontological graphs. This operation extracts concepts not mentioned in the text message. Consider our example: “A blue ball was on the beach. A woman walks on the beach. She takes the ball and kicks it”. It seems clear that our imagination should build the picture of the woman that binds her body to take the ball by hands, even though nothing in the text mentioned that bio-mechanical process. To do that, *Imaginizer* will use *ontological confabulation* for extracting knowledge associated with provided concepts. Confabulation was defined [17] as a maximization of probability to start from nodes a, b and c and get node d: $p(abc|d) \sim p(a|d) * p(b|d) * p(c|d)$. We start from any input node A, and then randomly walk and calculate its probability to get to node B. That probability obviously depends on the order degree for node B. The more nodes connected to B through some path, the higher probability it is to get there. The initial algorithm proposed in [9] uses only the weight of the edges that were calculated from pairwise frequencies of two nodes in some text.

Here we propose a new algorithm to calculate the probability of transitions by using node degrees and joint probabilities of dK-series (0K, 1K, 2K, 3K) that was extracted from the cohort of the text during building of the ontological graph. Due to analysis [14] that any properties of the complex graph can be reconstructed by a random graph with identical statistical properties for 0 to 3K sub-graphs, we suggest calculating the probability of a transition from A to B through some intermediate nodes as a sum over degree distributions for all intermediate 0K,1K,2K and 3K sub-graph between A and B.

If we start from several input nodes, then the total probability to get to the node B is sum over all probabilities calculated for each input node. The B node with highest probability will be taken as a part of new blended space. The next less probable node was taken as a part of the imagined space if its probability was higher then some threshold. That threshold was estimated heuristically.

Figure 8. Final imaginary space emerges as a result of ontological confabulation.

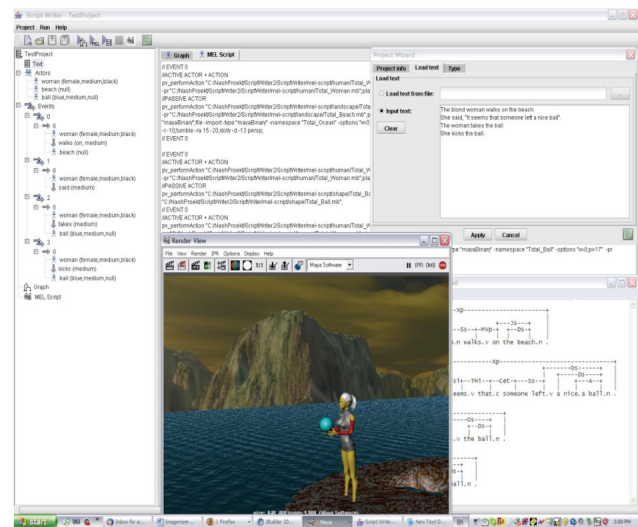
Confabulation – blended imaginary space



Imaginizer

Imaginizer is an attempt to visualize semantics imbedded in the simple English text. It is done in such a way to emulate a human reading a story and imagining it from a first person perspective. The first limited version of the software was called *ScriptWriter* (now renamed to *Imaginizer*). It provides us with the ability to animate short stories with primitive objects, humans, and several landscapes. *ScriptWriter* performs text processing, semantic extraction and animation planning. It provides a connection to sensory-motor system of agents – “Actors” and support multi-agent coordination. *ScriptWriter* scenario was organized as a collection of the behavioral actions of the actors or simply “Actions”. During the animation process some of actors and their actions encapsulate some other actions in the nested way that will produce sequential behavior. An example of sequential behavior is shown below: “A woman walks on the beach. There was a blue ball on the beach. She kicks the ball.” This text can be animated by *ScriptWriter* with minimal human intervention.

Figure 9. Imaginizer (ScriptWrite) screen shot



Current Work and Conclusions

A key goal for our work is providing biomedical researchers, who do not have any expertise in ontologies or query formulation, the ability to easily query a distributed collection of data resources. The researcher will not know all the particulars of each data source or what data is contained in them. Therefore, we have been building this system for researchers to be able to easily perform key word type queries. This system allows researchers to navigate through hierarchically organized ontological concepts which are mapped to actual data. The main benefit of this approach, is that it lets us abstract the details from the user so that they do not need to know the detail of sources and be an expert in query formulation, which they are not. The process of using an ontological based query vocabulary provides the necessary framework for the user relating to the nature and extent of data available from the federated database. We continue to work on the ability to formulate a natural language query as well as on technology to present requested data in different visual modalities.

Acknowledgements

This work is supported by NIH BIRN-CC Award No. U24 RR019701-01 (NCRR BIRN Coordinating Center Project BIRN005). The authors thank their colleagues and students involved in the BIRN project for their contributions, in particular, Willy WaiHo Wong, Joshua Tran, Asif Memon and Ilya Zaslavsky.

References

1. Grethe JS, Baru C, Gupta A, James M, Ludaescher B, Martone ME, Papadopoulos PM, Peltier ST, Tajasekar A, Santini S, Zaslavsky IN, and Ellisman MH (2005) Biomedical Informatics Research Network: Building a National Collaboratory to Hasten the Derivation of New Understanding and Treatment of Disease, *Stud Health Technol Inform.* 2005;112:100-9
2. B. Ludäscher, A. Gupta, M. E. Martone, "Model-Based Mediation with Domain Maps", 17th Intl. Conference on Data Engineering (ICDE), Heidelberg, Germany, IEEE Computer Society, 2001.
3. Gupta, B. Ludäscher, M. E. Martone, Registering Scientific Information Sources for Semantic Mediation, 21st International Conference on Conceptual Modeling, (ER), Tampere, Finland, pp. 182-198, October 2002.
4. M.E. Martone, A. Gupta, B. Ludaescher, I. Zaslavsky and M.H. Ellisman, , Federation of brain data through knowledge-guided mediation. In R. Kotter(ed.), *Neuroscience Databases: A Practical Guide*, Boston: Kluwer Academic Publishers, pp. 275-292, October 2002.
5. Gupta, B. Ludäscher, J.S. Grethe, M.E. Martone, "Towards a Formalization of Disease-Specific Ontologies for Neuroinformatics", *Neural Networks*, 16(9), pp. 1277-1292, 2003.
6. M.E. Martone, A. Gupta and M.H. Ellisman, "e-Neuroscience: challenges and triumphs in integrating distributed data from molecules to brains", *Nature Neuroscience*, 7(5), May 2004, pp. 467-472.
7. A. Gupta, B. Ludäscher, M.E. Martone, A. Rajasekar, E. Ross, X. Qian, S. Santini, H. He, I. Zaslavsky, "BIRN-M: A Semantic Mediator for Solving Real-World Neuroscience Problems ", *ACM Conf. on Management of Data (SIGMOD) (Demonstration)*, page 678, June 2003.
8. V. Astakhov, A. Gupta, S. Santini, J.S. Grethe, "Data Integration in the Biomedical Informatics Research Network (BIRN)", *DILS 2005*: 317-320 Springer, Data Integration in the Life Sciences.
9. V. Astakhov, A. Gupta, J.S. Grethe, E. Ross, D. Little, A. Yilmaz, X. Qian, S. Santini, M.E. Martone, M.H. Ellisman: Semantically Based Data Integration Environment for Biomedical Research.19th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2006), June 2006, Salt Lake City, Utah, USA.
10. A. Prince and P. Smolensky* "Optimality: From Neural Networks to Universal Grammar". *Science* vol. 275 14 March 1997.
11. Tony A. Plate "Holographiv Reduced Representation." CSLI Lecture Notes Number 150, 1997.
12. Gupta A., C. Bean, W. Bug, C. Fennema-Notestine, M. E. Martone, J. A. Turner, J. S. Grethe. Creating a community-based knowledge management framework for integrating neuroscience data via ontologies. Society for Neuroscience Annual Meeting (2006).
13. Turner J. A , C. Fennema-Notestine, M. E. Martone, A. R. Laird, J. S. Grethe, W. Bug, A. Gupta, C. Bean. You say potato, I say potahto: Ontological engineering applied within the Biomedical Informatics Research Network, Society for Neuroscience Annual Meeting (2006).
14. Priya Mahadevan_ Dmitri Krioukov† Kevin Fall‡ Amin Vahdat: A Basis for Systematic Analysis of Network Topologies.<http://www.krioukov.net/~dima/pub/index.html>
15. Fauconnier, G., Turner Mark Book: The way we think.
16. Alan Prince* and Paul Smolensky* Optimality: From Neural Networks to Universal Grammar. *Science* vol. 275 14 March 1997.
17. Robert Hecht-Nielsen.:ATheory of Cerebral Cortex. UCSD Institute for Neural Computation Technical Report #0404