IDACT Query Manager for Heterogeneous Dataset Assimilation

Brian Hay
Kara L Nance
University of Alaska Fairbanks

Abstract: There exists a wealth of data in heterogeneous formats related to NASA science missions. The Query Manager (QM) for the IDACT System (funded under AIST-02-0135) minimizes the efforts required of the scientific researcher to obtain and format datasets relevant to the scientific research domain. The QM accepts a general topical request from the researcher; this request is analyzed in the context of the Data Source Registry which maintains domain-specific content information. If a topical match is found, the QM generates the appropriate query process and obtains the data for the user through the registered data sources. The QM uses complex selection criteria including function and specialized operations on different data types to build and manage query selectors. These query selectors are used to generate data-source specific queries. In some cases, the queries will be SQL statements for access to a relational database system, although in other cases the queries may utilize alternate interfaces, such as web service calls, a connection to a data stream such as a telemetry feed, or an FTP server for files. The query results are passed to the IDACT Data Transformation (TM), which is tasked with transforming the resultant data sets into single homogeneous view as desired by the user in the format the format requested. The major benefit of this approach is that the scientific researcher can have easy and useful access to the voluminous and complex scientific data sets, while reducing the associated analysis and operational times and costs.

IDACT

The IDACT system [2, 9, 10] provides a modularized approach to aid scientific users in identifying, collecting, and synthesizing diverse, geographically distinct, heterogeneous datasets in order to better investigate scientific phenomenon. The solution facilitates a query to a middle-layer system that allows data consumers to locate, retrieve, and transform datasets from multiple data sources. It incorporates intelligent automated processes to provide data access to a wider range of data consumers, with fewer data processing skills, and is built as a modular system, to ensure that it can continue to be used as technology changes in the future. The benefits of this approach to the scientist and modeler include seamless autonomous data collection, data system operation, and management of heterogeneous entities in support of scientific analysis and modeling, thus reducing the associated operational time and costs

IDACT Components

The IDACT Query Manager (QM), Datasource Registry (DR), and Transformation Manager (TM) [3, 8, 9, 10] components operate in concert to allow data consumers to easily identify, acquire and transform data to a format that meets their needs. In order to facilitate such operations, data sources are first registered with the DR

Datasource Registry

The IDACT Datasource Registry (DR) [2, 3, 9, 10] provides a mechanism for registering datasources and storing appropriate metadata and the capability to define and store information about relationships between datasources through the process of relating the datasource contents to registered common fields. In addition the DR provides these services to a potentially distributed platform in a flexible and extensible system that will facilitate developer modifications. The structure of the datasources within the DR is hierarchical. Information is registered about the datasources as a whole. In addition, the DR retains information about the contents of the datasources, primarily as fields within datasources. Further the DR provides a mechanism for defining relationships between fields in distinct datasources through the use of domain-specific registered common fields.

During the registration process, an attempt is made to map fields in the data source to previously or newly defined common fields, allowing the QM to build suitable queries, and the TM to define new transformations as necessary. In order for this process to be accessible to the typical data owner, several automated approaches have been applied, allowing the DR to generate an initial field mapping which can then be accepted or modified by the user. Furthermore, the DR learns from experience, so that its performance in future registration attempts requires less user interaction. The goal is to perform as much of the mapping work as possible in an automated fashion, then allow the user to modify or augment those mappings as necessary, while recording any changes made by the user to provide better performance in future efforts.

Determining Name-Based Mappings Using the Association Knowledgebase

The DR first searches previously identified associations in order to locate potential associations in a new data source. Suppose that a new data source is a database containing a table named *Locations* which has the previously unencountered field names *NorthSouthCoord*

and *EastWestCoord*, and that the common fields have been registered in the DR for *Latitude* and *Longitude*. Since no associations for the field *NorthSouthCoord* exist in the association knowledgebase, the DR may request that the user provide a suitable mapping. Once that mapping has been made, the DR creates an association in its knowledgebase, and when the same field name is encountered in new data sources in the future the DR would automatically be able to generate an association, which the user would then be free to modify or accept.

As the association knowledgebase grows in size it is likely that a source field name may be mapped in the association knowledgebase to several different common fields, depending on the data source in which it appears. Consider the source field name *Location*, which is very frequently used to contain geographic coordinate information. The association knowledgebase may contain a default rule for handling such a field, which describes a *split* association resulting in the field data being associated with both the *Latitude* and *Longitude* common fields [9]. In data source *A* owned by Alice, the source field *Location* was associated with the common field *City*, and in data source *B* owned by Bob the field *Location* was associated with the common field *Country*. At this point, the relevant subset of association knowledgebase is represented by figure 1.

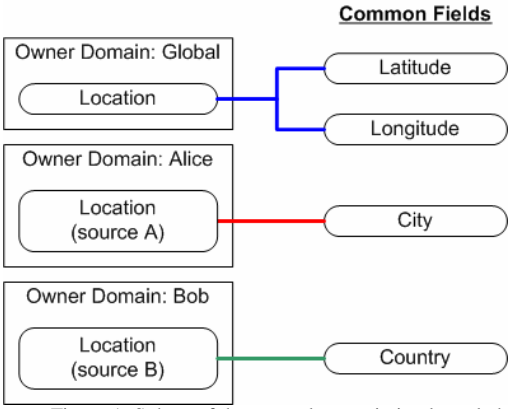

Figure 1: Subset of the example association knowledgebase.

When faced with a new data source that contains a field named *Location*, the DR builds an ordered list which rank potential matches in the association knowledgebase as follows:

- The first search is made for matches in the same owner domain (i.e. with the same data owner) as the new data source, and any matches are added to the list. For example, if Bob was the owner of the new data source, then the highest ranked association in the list would be to the *County* common field, whereas if Alice was the owner of the new data source, then the highest ranked association in the list would be to the *City* common field. However, if Claire was the owner of the new data source, then no association would be found at this step and the list would remain empty.
- The second search is made in the *Global* owner domain, which represents very common associations that are likely to be encountered. In this example, there is a split association defined in the *Global* owner domain for the source field *Location*, so that association is appended to the list.
- The third search is made in any owner domains not previously considered. For example, if Bob was the owner of the new data source, then all owner domains other than *Bob* and *Global* are searched at this stage, and resulting matches are appended to the list.

If the ordered list is non-empty after the search process has completed, then an automatic association is proposed by selecting the first association in the list. In the event that the user modifies the association, they are presented with the remainder of the list as the most likely associations, reducing the amount of effort required on the part of the user. Once the user finds a suitable association, a new entry is added to the association knowledgebase.

If Alice submits a new data source, *C*, that she owns, in which there is a field named *Location*, the resulting ordered list would be as follows:

1. *Location* associated with common field *City*, from the *Alice* Owner Domain.
2. *Location* associated with common fields *Latitude* and *Longitude*, from the *Global* Owner Domain.
3. *Location* associated with common field *Country*, from the *Bob* Owner Domain.

The DR would then propose an association from the *Location* field to the common field *City* using the first item from the list, and provide Alice with an opportunity to approve or modify it. Once Alice decides on the correct association, a new entry is added to the association knowledgebase that describes the new association for data source *C*. For example, if Alice accepts the proposed association, then the resulting relevant subset of association knowledgebase is shown in figure 2.

If Alice now submits another data source, *D*, that she owns, in which there is a field named *Location*, the resulting ordered list would be unchanged. However, in this case assume that Alice chooses to associate the *Location* source field with the *Country* common field. The resulting relevant subset of the association knowledgebase after such an operation is shown in figure 3.

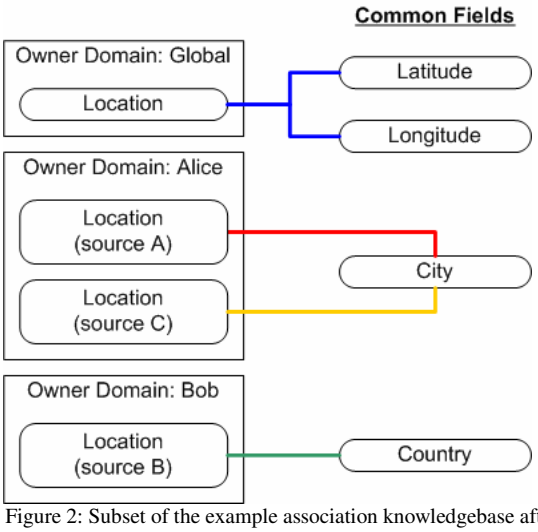

Figure 2: Subset of the example association knowledgebase after Alice accepts the proposed association for data source *C*.

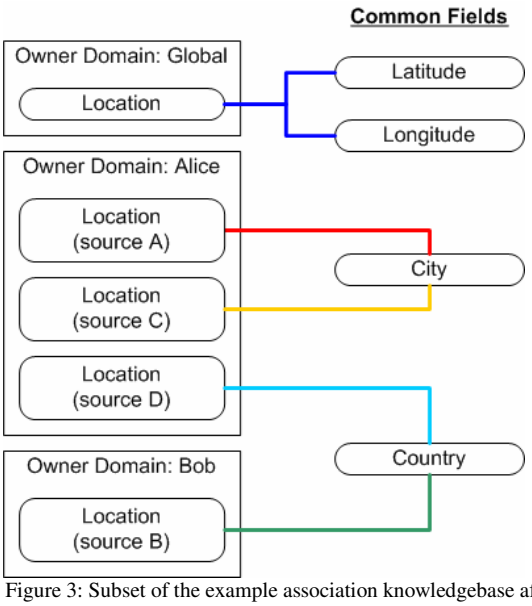

Figure 3: Subset of the example association knowledgebase after Alice modifies the proposed association for data source *D*.

Conflict Resolution in the Association Knowledgebase

If Alice submitted another data source with the association knowledgebase in the state depicted in figure 3 a conflict would arise, in that when searching the *Alice* owner domain for the field *Location*, two possible associations would be discovered, one mapping to the *City* common field, and the other to the *Country* common field. Conflicts such as this are resolved using the following strategy:

1. If possible, preference is given to the association whose context most closely matches the context of the source field. This may or may not be possible, depending on the type of the data source, and is discussed in more detail later in this paper.
2. If two or more associations cannot be ordered using a contextual approach, preference is given to the association which appears most frequently. In this example, the mapping of *Location* to *City* for Alice's new data source would be ranked higher than the *Location* to *Country* mapping.
3. If two or more associations appears equally frequently in a search stage, preference is given based on the most recently created association.

Determining Name-Based Mappings Using Field Name Comparison

If the search of the association knowledgebase returns an empty list, there are some additional operations that may allow the DR to propose associations to the user. The first approach has been used successfully by the SIMON agent [1, 4, 5, 6, 7, 11, 12, 13] and is, in its most basic form, based on a comparison of field names. For example, consider a data source based on the database tables shown in figure 4, in which there are several locations defined as a latitude/longitude pair, and associated with each location are one or more dated measurements.

For each of the fields in the data source an initial attempt is made to find a registered common field of the same name. For example, if we have registered common fields named *Latitude*, *Longitude*, and *Date* these associations could be made quickly and easily, and these matches could be appended to the ordered list of possible associations for those particular source fields.

However, it is likely that field names will typically not exactly match the available registered common field names. For example, consider the similar scenario shown in figure 5, where the field names have been modified. As a result, the exact match approach for creating association between the data source fields *Lat*, *Long*, and *MeasurementDate*, and registered common fields such as *Latitude*, *Longitude*, and *Date* would obviously not work. When this problem was encountered during the development and use of SIMON, substring matching was found to be quite effective in identifying candidate associations, which the user could then accept or modify.

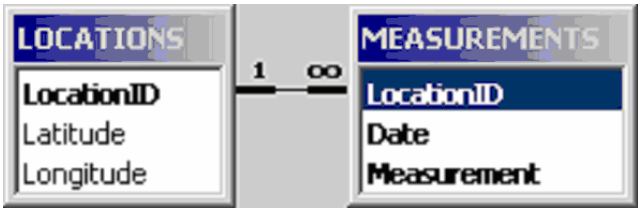

Figure 4: A basic database data source.

For example, the source field name *Lat* would result an association with the registered common field *Latitude* due the substring match. As with exact matches, any partial matches found are appended to the ordered list of candidate associations.

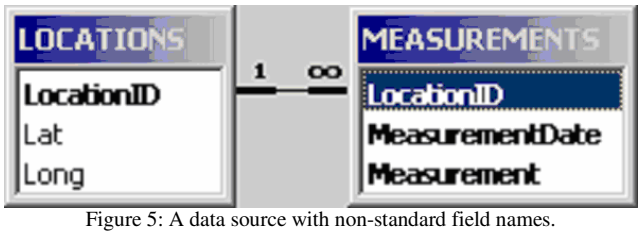

Figure 5: A data source with non-standard field names.

Context-Based Approaches to Name-Based Associations

In the event the source data is organized in hierarchical fashion (such as in XML data sources, for example), the DR will attempt to determine the context of a source field in order to resolve ranking conflicts between multiple potential associations, or to narrow the choice of possible registered common fields in cases where other approaches have failed to generate any suitable matches. For the purposes of the DR, the *context* of a source field refers to the path from the root of the data to the source field, expressed in terms of *common* fields. For example, consider the subset of the association knowledgebase depicted in figure 6, in which two associations for the *Minutes* source field are present in the *Alice* owner domain. Each of these associations are really part of a combine association, using *Degrees*, *Minutes*, and *Seconds* in the case of the *Latitude* registered common field, and *Hours*, *Minutes*, and *Seconds* in the case of the *Time* registered common field. In addition, the Lat source field is also associated with the Latitude registered common field.

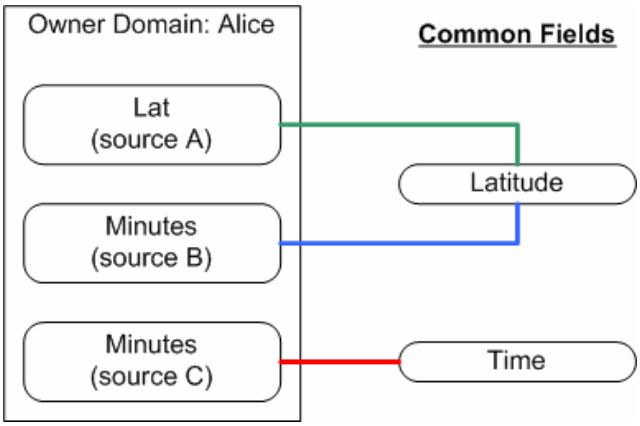

Figure 6: Association knowledgebase with two associations for the *Minutes* source field in the *Alice* user domain.

If the DR is presented with a new hierarchical data source as depicted in figure 7 in which the *Minutes* field appears, the two potential mappings exist in the *Alice* owner domain. However, by examining the hierarchy of the *Minutes* field in the new data source, the DR determines that the *Latitude* common field appears (though the *Lat* to *Latitude* association), and as such the association between *Minutes* and *Latitude* given preference over the association with *Time*.

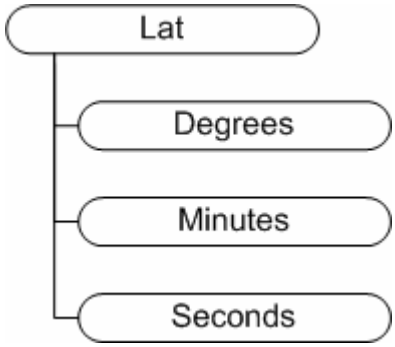

Figure 7: Partial hierarchy of a new data source, in which the DR is attempting to locate an association for the *Minutes* field.

Additional Approaches

The DR employs some additional approaches that allow promising associations to be identified, including content-based approaches in which either the source data values in are compared with known registered common field values in an effort to identify similarities. In addition, the pattern of data values can be used, as it was by SIMON [1, 4, 5, 6, 7, 11, 12, 13] to determine which registered common fields contain data types similar to those found in unrecognized source data fields.

Although this paper has focused on name-based data sources, the DR is also capable of generating potential associations for data sources in which named fields are not present, such as a frame/sub-frame based format. In these cases, positional references, either absolute or relative, are used at the source data field identifiers.

Conclusion

The IDACT Data Registry component allows a data owner to register their dataset with an IDACT instance, after which time the Query Manager and Transformation Manager can use the resulting association knowledgebase to retrieve and transform data for users. During the registration process, the intent of the DR is to leverage past experience to automate much of the effort of associating fields in the data source to registered common fields. Once the automated association process is complete, the data owner is then free to accept or modify the proposed associations. The DR uses this information to not only

provide the QM and TM components with the ability to create custom queries and transformations, but also to improve the data registration process for future new data sources.

Acknowledgements

IDACT, including the Transformation Manager, Datasource Registry, and Query Manager components, is being developed under NASA Advanced Information Systems Technology Program (NASA award AIST-02-0135).

References

[1] Hay, B. and K. Nance. "SIMON: An Intelligent Agent For Heterogeneous Data Mapping. International Conference on Intelligent Systems and Control. Honolulu, Hawaii. August 13-18, 2000.

[2] Lisee, M., K. Nance, and B. Hay. "HTEST: A Configurable Triggered Data System for Simulating Space Systems Data Source Integration. Proceedings of the 23rd Space Simulation Conference. November 2004.

[3] Nance, K. and B. Hay. "Automatic Transformations Between Geoscience Standards Using XML." Computer & Geosciences Journal: Special Issue on XML Applications in Geosciences. (In Press)

[4] Nance, K. "Data System Planning for Formerly Used Defense Sites (FUDS)." Proceedings of the American Society of Business and Behavioral Sciences: Government and Business Problems. February 20-26, 1997.

[5] Nance, K. "Decision Support and Data Mining." Proceedings of the International Simulation Multiconference. April 6 – 10, 1997.

[6 Nance, K. "Synthesis of Heterogeneous Data Sets." Proceedings of the 9th Annual Software Technology Conference. May 6 – 10, 1997

[7 Nance, K. "Applying AI Techniques in Developing Plans for Formerly Used Defense Sites (FUDS) in Alaska." Mathematical Modeling and Scientific Computing, vol. 8, 1997.

[8] Nance, K. and B. Hay. "Automating Conversions Between Metadata Standards Using XML and XSLT." 2004 DAMA International Symposium and Metadata Conference. May 2004.

[9] Nance, K. and B. Hay. "IDACT: Automating Data Discovery and Compilation." Proceedings of the 2004 NASA Earth Science Technology Conference, May 2004.

[10] Nance, K. and B. Hay. "IDACT: Automating Scientific Knowledge Discovery" Proceedings of the IASTED International Conference on Environmental Modelling and Simulation. November 2004.

[11] Nance, K. and J. Wiens. "SynCon: Simulating Remediation Planning." Mathematical Modeling and Scientific Computing, 1998.

[12] Nance, K. J. Wiens and S. George. "The SynCon Project: Arctic Regional Environmental Contamination Assessment" Proceedings of the 38th Annual Western Regional Science Conference. February, 1999.International Conference of the Society for Information Technology & Teacher Education. February, 1999.

[13] Nance, K, J. Wiens and S. George. "The SynCon Project: Phase II Assessing Human Health in the Arctic" Proceedings of the International ICSC Congress on Computational Intelligence: Methods and Applications. June, 1999.