E-Agricultural Services and Business

A Conceptual Framework for Developing a Deep Web Service

Nattapon Harnsamut, Naiyana Sahavechaphan
nattapon.harnsamut@nectec.or.th, naiyana.sahavechaphan@nectec.or.th

Large-scale Simulation Research Laboratory
National Electronics and Computer Technology Center
Pathumthani, Thailand

ABSTRACT

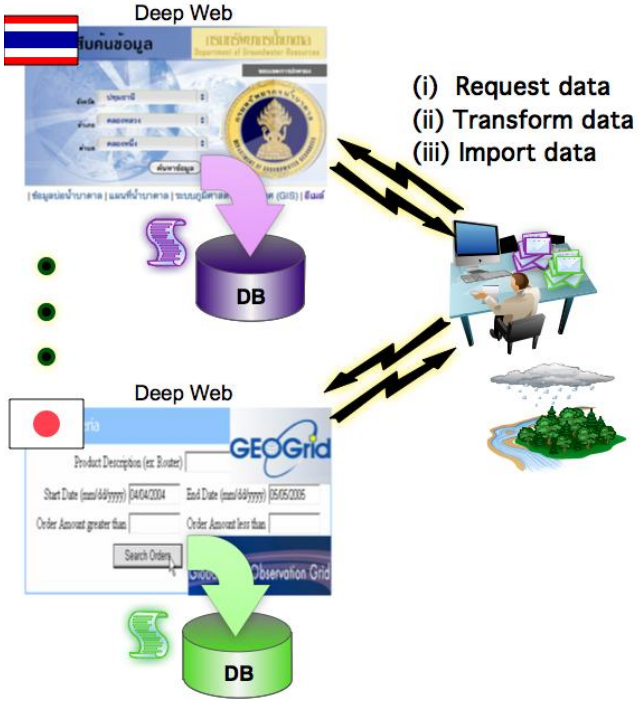
Today, many agricultural organizations have developed Web databases or deep Web' to publish their observation data. However, the availability of deep Web alone limits any agricultural applications to automatically perform. There is thus a need for a deep Web Service or simply *dWS* that facilitates any client applications to automatically access the data available on a specified deep Web. It is likely that several *dWS'* must be developed. In this paper, we thus propose a conceptual framework that facilitates the development of a *dWS* as per a given deep Web. Specifically, it provides an essential mechanism that enables *dWS* developers to simply generate all necessary configurations, and then build the corresponding *dWS*.

Keywords *: Information Retrieval; Deep Web; Deep Web Service; Query Transformation; Table Mining; Data Transformation; Data Filtering*

Introduction

Many agricultural applications have been developed to facilitate agriculturists in efficient decision-making and problem-solving. Essentially, these agricultural applications are driven by observation data that is different in term of observation types, geographical locations, time intervals and freshness, and is measured by self-organization1, other organizations or both.

While an observation data from the self-organization would be simply accessible by agricultural applications, the access to the observation data available by other organizations would rather be complicated. This is due to the fact that Web databases [1] (or deep Web) are commonly used for data publication to any users. Here, to feed applications the data available on a deep Web, users must (i) request for the desirable observation data through the Web query form; (ii) transform the requested data into a specified format; and (iii) import the transformed data into such applications. This process is shown in Figure 1. According to the demand of daily or real-time observation data, the data feeding process should thus be recursively done as per each deep Web. This process is clearly time-consuming and must be manually achieved, prohibiting any agricultural applications that require data from deep Web' to be *automatically* performed.



Recognizing that the data access on a deep Web is challenge, several approaches have thus been proposed in the literature. In particular, wise-integrator [2], Web query interface [3] and query tunneling [4] apply schema matching technique to integrate different domain-specific deep Web' and introduces the corresponding unified Web query form. While the access to data from several deep Web' can simply be achieved through a unified deep Web (a deep Web of deep Web'), the manual feeding process still remains. In addition, it is limited to a single domain of data.

Interestingly, Web Services Technology [5] has been developed to facilitate the interoperability of applications regardless of their underlying platforms and infrastructures. To address the above problems on the availability of a deep Web alone, there is thus a need for a deep Web Service or simply *dWS* that facilitates any client applications to automatically access the data available on a specified deep Web. It is likely that several *dWS'* must be developed. In this paper, we thus propose a conceptual framework that facilitates the development of a *dWS* as per a given deep Web. Specifically, it provides an essential mechanism that

Configurations, and then build the corresponding *dWS*.

The rest of this paper is organized as followed: Section 2 describes the *dWS* framework architecture. Each module of the *dWS* framework is presented in Section 3 - 6. Section 7 concludes the paper.

DWS Framework

Figure 2 illustrates *dWS* framework. Essentially, it composes of four main components: (i) Query Transformation – that transforms the query defined based on a chosen metadata standard (termed as standard query) into the query understood by a deep Web (termed as Web query); (ii) Table Mining – that extracts the Web result represented in the HTML-based table format into a simplied format understood by computer; (iii) Result Transformation -- that transforms the simplified Web result from the Table mining module into a standard result as per the previously used metadata standard; and (iv) Result Filtering – that filters out any potential irrelevant results and returns the client the standard result. Details of these modules are provided in the following sections.

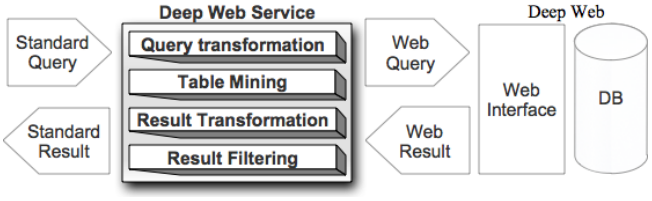


Figure 2. he deep Web Service Framwork.

Here, developers can simply implement the *dWS* framework to generate a particular *dWS* as per a deep Web. With the availability of *dWS*', an application can automatically access observation data provided by several deep Web' via their corresponding *dWS*' at any times as shown in Figure 3. Specifically, an application invokes each *dWS* with standard query. *dWS* next transforms the standard query into the Web query. Then, *dWS* submits the Web query to the deep Web. On the return of Web result, *dWS* transforms it into a standard result and finally returns the standard result to the application.

Query Transformation

It is likely that two or more deep Web' that provide similar observation data have used different metadata to specify Web query form and Web result, termed as Web query and Web result metadata respectively. Here, to accommodate the data access of such deep Web', the metadata standard of choice should be developed. Clearly, there would be the difference between the metadata standard and Web query metadata as well as Web result metadata. For example, in Figure 4, the three elements *"Amphur"*, *"Day"*, and *"Timestamp"* altogether define the metadata standard, while the Web query metadata is specified by *"Province"*, *"Day"*, and *"Time"* elements. Here, these two metadata are structurally different in term of element semantic and format.

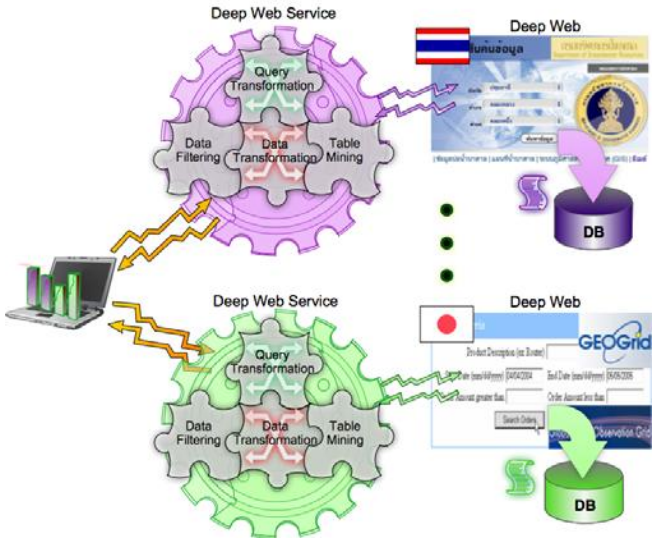


Figure 3. He deep Web Services.

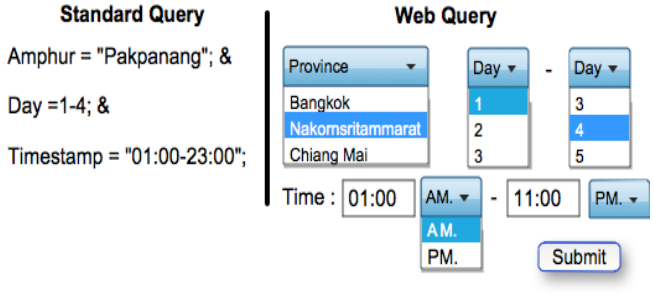


Figure 4. Standard Query versus Web Query.

Consider Figure 4 once again. Here, in addition to the structural mismatch, there is the content mismatch between the two elements *"Amphur"* and *"Province*. To solve the content conflict, the appropriate method must be employed. For example, the geo-code is used to converge the *"Amphur = Pakpanang"* to *"Province = Nakornsitammarat"* since *"Pakpanang"* is an amphur in the *"Nakornsitammarat"* province.

According to the distinction between standard query and Web query, there is thus a need for the transformation of standard query into Web query. This transformation is accomplished by the query transformation module as shown in Figure 5. Specifically, it (i) extracts the metadata from Web query form; (ii) facilitates user to visually create the query mapping configuration (QMC) that captures the transformation knowledge between the standard query and the Web query; and (iii) analyzes the QMC and then automatically generates an appropriate Web query to be submitted to the deep Web. As an example, the standard query *"Amphur = Pakpanang"*, *"Day between 1 to 4"* and *"Timestamp between 01:00 to 23:00"* is transformed to the corresponding Web query *"Province = Nakornsitammarat"*, *"Day between 1 to 4"* and *"Time between 01:00 AM. to 11:00 PM."*

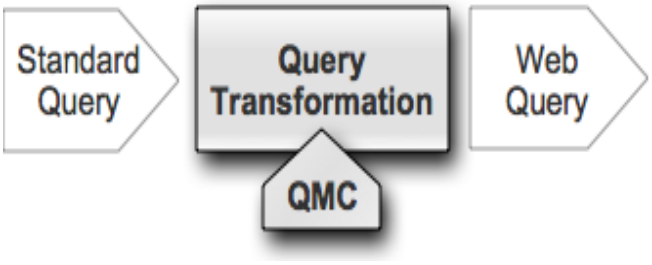
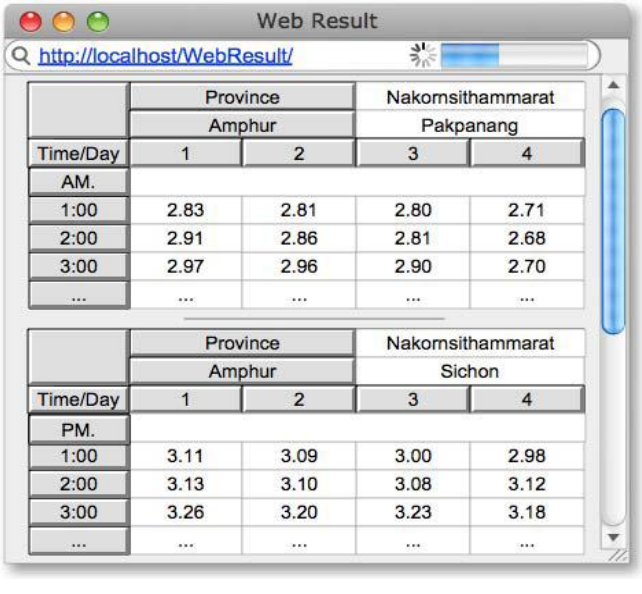


Figure 5. The Query Transformation.

Table Mining

Figure 6 shows the Web result on return from submitting the Web query *"Province = Nakornsitammarat"*, *"Day between 1 to 4"* and *"Time between 01:00 AM. to 11:00 PM.* which is generated by the query transformation module. Here, the Web result is typically in HTML-based table format. While it is simply be parsed by human, this parsing is difficult for computer. It is thus necessary to transform all table formats into a corresponding unified format. A horizontal 1-dimensional table format has been chose as a unified format since it (i) preserves all information contained in an original format; and (ii) supports each particular query ranging from a simple to complex one.



	Province		Nakornsithammarat	
	Amphur		Pakpanang	
Time/Day	1	2	3	4
AM.				
1:00	2.83	2.81	2.80	2.71
2:00	2.91	2.86	2.81	2.68
3:00	2.97	2.96	2.90	2.70
...	...	...	...	...

	Province		Nakornsithammarat	
	Amphur		Sichon	
Time/Day	1	2	3	4
PM.				
1:00	3.11	3.09	3.00	2.98
2:00	3.13	3.10	3.08	3.12
3:00	3.26	3.20	3.23	3.18
...	...	...	...	...

Figure 6. The Web Result.

This transformation is accomplished by the table mining module. Specifically, it composes of four main components as show in Figure 7. Details of this component along with complete examples are given in [6].

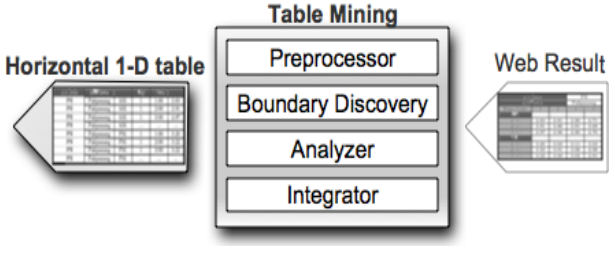


Figure 7.he Table Mining.

- *Preprocessor* – that extracts all potential attributes and values from the Web result represented in the HTML-based table, and simplifies all spanning cells and super rows[7] into their appropriate forms to facilitate the process of the rest components.
- *Boundary Discovery* - that applies the similarity of data types in order to find the boundary of each intrinsic primitive tables residing in the composite table. Consider Figure 7 as an example once again. Here, the Web result contains two primitive tables designed for Amphur Pakpanang and Amphur Sichon.
- *Analyzer* - that utilizes the semantic correspondence and hence analyzes the attribute-value relationship and the value association of cells in a boundary. The formula given in [8] is used for semantic similarity and correspondence of each labels.
- *Integrator* - that integrates different pieces of the analyzed information resulted from the Analyzer component by gradually forming the desirable horizontal 1-D table as shown in Figure 9.

Province	Amphur	Time Period	Time	Day	Height
Nakornsitammarat	Pakpanang	AM.	1:00	1	2.83
Nakornsitammarat	Pakpanang	AM.	2:00	1	2.91
Nakornsitammarat	Pakpanang	AM.	3:00	1	2.97
Nakornsitammarat	...	...	...	...	...
Nakornsitammarat	Sichon	PM.	1:00	2	3.09
Nakornsitammarat	Sichon	PM.	2:00	2	3.10
Nakornsitammarat	Sichon	PM.	3:00	2	3.20
...	...	...	...	...	...

Figure 8. The Horizontal 1-Dimension Table corresponding to Figure 6.

Result Transformation

Recall that there is the metadata conflict between the metadata standard and the Web query metadata. There is also the mismatch among the metadata standard and the Web result metadata. There is thus a need for the transformation of a Web result into a standard result. This is achieved by the result transformation module as shown in Figure 9.

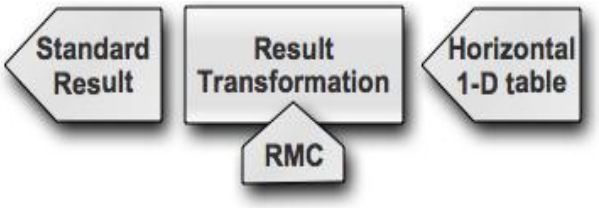


Figure 9. The Result Transformation.

Specifically, the result transformation module (i) extracts the metadata from a horizontal 1-D table on return from the table mining module; (ii) facilitates users to visually create the result mapping configuration (*RMC*) that captures the transformation knowledge between a metadata standard and a Web result metadata (a horizontal 1-D table); and (iii) analyzes the RMC and then automatically generates an appropriate standard result as per a Web result.

Figure 10 shows the standard result corresponding to the Web result in Figure 8. Here, the column *"Province"* is removed. The column *"Time Period"* is adjusted with the new attribute name *"Timestamp"* along with its content into 24-hour format.

Amphur	Timestamp	Day	Height
Pakpanang	1:00	1	2.83
Pakpanang	2:00	1	2.91
Pakpanang	3:00	1	2.97
...	...	...	...
Sichon	13:00	2	3.09
Sichon	14:00	2	3.10
Sichon	15:00	2	3.20
...	...	...	...

Figure 10. The Standard Result.

RESULT FILTERING

Recall that there is the content conflict between the standard query and the Web query (see section 3). This can result in the irrelevant results that should be filtered out, making the result be applicable to the standard query. This is accomplished by the result filtering module as shown in Figure 11.

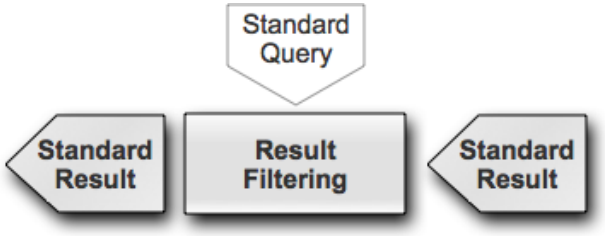


Figure 11. The result filtering.

Consider one again Figure 10. Here, as per the standard query *"Amphur = Pakpanang"*, *"Day between 1 to 4"* and *"Timestamp between 01:00 to 23:00",* this result contains some irrelevant hits of Sichon Amphur. These hits must be filtered out to preserve the standard query as shown in Figure 12.

Amphur	Timestamp	Day	Height
Pakpanang	1:00	1	2.83
Pakpanang	2:00	1	2.91
Pakpanang	3:00	1	2.97
...	...	...	...
~~Sichon~~	~~13:00~~	~~2~~	~~3.09~~
~~Sichon~~	~~14:00~~	~~2~~	~~3.10~~
~~Sichon~~	~~15:00~~	~~2~~	~~3.20~~
...	...	...	...

Figure 12. The final Standsard Result.

Future Work anfd Conclusion

This paper proposes the conceptual framework that facilitates the development of a *dWS* as per a given Web database or deep Web. Specifically, it provides an essential mechanism that enables *dWS* developers to simply generate all necessary configurations, and then build the corresponding *dWS*. With the availability of *dWS*', any client applications can automatically access data from the corresponing deep Web'. Furture work is to develop such framework and make it publicly available. Essentially, it makes contribution to our Web portal [9] as it enables the access of data across different Web databases rather than relational databases alone.

REFERENCES

[1] H. Bin, P. Mitesh, Z. Zhen,and C.K. Chen-Chuan, "Access the deep web: Survey," Commun. ACM journal, pp 94-101, 2007

[2] H. He, W. Meng, and Z. Wu, "Wise-integrator: an Automatic Integrator of Web Search Interfaces for E-commerce," VLDB Conference, pp. 357–368, 2003.

[3] B. He, and C.C. Chang, "Statistical schema matching across Web Query interfaces," Proc. ACM SIGMOD Conf., 2003

[4] T.Kabisch, and M. Neiling, "Wrapping of Web Sources with restricted Query Interface by Query Tunneling," Electron. Notes Theor. Comput. Sci. Journal, pp. 55-70, 2006

[5] Web Services @ W3C. http://www.w3.org/2002/ws

[6] Table Mining. www.informationgrid.org/publications/tableMining/

[7] A. Tengli, Y. Yang, and N. L. Ma, "Learning table extraction from examples," Proc. the 20th international conference on Computational Linguistics, pp. 987, 2004.

[8] N. Tansalarak and K. T. Claypool. Qmatch - using paths to match xml schemas. Data Knowl. Eng., 60(2):260–282, 2007

[9] N. Sahavechaphan, J. Phengsuwan, N. Harnsamut, S. Vannarat, A. Kawtrakul, "The Sustainable Web Portal for Observation Data," AFITA, 2010.