The WWW Based Data Mining Toolbox Architecture

Lukasz Kurgan [1,2], Krzysztof J. Cios [1,2,3,4] and Michael Trombley [1]

[1] Dept. of Computer Science and Engineering, University of Colorado at Denver
[2] Dept. of Computer Science, University of Colorado at Boulder
[3] University of Colorado Health Sciences Center, Denver
[4] 4cData, LLC, Golden, CO, U.S.A.
emails: lkurgan@carbon.cudenver.edu, Krys.Cios@cudenver.edu, mptrombl@ouray.cudenver.edu

Abstract. This paper presents the Data Mining (DM) toolbox architecture based on cutting edge World Wide Web (WWW) technologies. The DM toolbox is used to discover new and useful knowledge by integrating results generated by multiple DM tools. The proposed architecture allows submission of data to the DM toolbox and generation of results that combine knowledge generated by several different DM tools. The DM toolbox dynamically finds DM tools that are relevant to a specific data mining task, submits the data to the tools, receives results of their analysis, and combines the results to generate a final report. The proposed architecture will increase the usability of DM tools, helping achieve a more consistent and better integrated Data Mining and Knowledge Discovery (DMKD) process.

1 Introduction

1.1 The KD Process

Knowledge Discovery (KD) is a nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns from large collections of data [7]. DM is just one of the KD steps concerned with the generation of knowledge from data. In practice, DM, KD, and DMKD are used synonymously to describe the same process.
Several DMKD process models were proposed. They include: the six-step model by [4, 5], the five-step model by [3], and the nine-step model by [8]. For an extensive discussion of the above models see [5].

1.2 The DM Toolboxes, Tools, and Methods

To define an architecture for a DM toolbox, we first define the difference between DM methods and tools. The DM method is an implementation of a DM algorithm. The DM tool is a DM method that can communicate and operate within the DMKD framework. In other words, a DM tool is a highly interoperable DM method that works within the framework.
A DM toolbox consists of several DM tools that are used in tandem to discover new and useful knowledge from data. The DM toolbox user would submit his data to the toolbox, and receive the combined knowledge discovered by all the DM tools that would be used by the toolbox. The most important feature of the proposed architecture is the ability of the DM toolbox to dynamically select and use only those DM tools that are relevant to a specific data analysis task, requested by user. The goal for the proposed architecture is to increase usability, user friendliness, and applicability of the DM tools that will be used. The proposed architecture will also increase consistency, and integration of the DMKD process model.

1.3 XML: Key Ingredient for Building DM Toolboxes

XML (eXtensible Markup Language) is a standard proposed by the WWW Consortium [1]. XML is a markup language for documents that contain structured or semi-structured information that uses custom-defined tags. Structured information consists of content and context. XML is a standard to identify structures in documents by enabling users to define tags (element names) that are used to describe context of the data. As such, XML is one of the best mediums for building data integration systems [12]. XML technology is widely used in industry to transfer and share information in a platform-independent way. Most of the current database management systems already support the XML standard.

From the DM point of view, XML can help in solving multiple problems:

- building standard data and knowledge repositories that would enable sharing of both data and the discovered knowledge between different DM tools that work on different software platforms. XML is able to help with both creation of such repositories and communication between them and the DM tools.
- implementing communication protocols between DM tools. By using standardized communication protocols, different DM tools, say, developed by different companies, could be integrated to generate broader, more understandable and accurate, data models.

To summarize, XML and XML-based technologies provide a wide range of tools that can help in transforming DM methods into tools, and integrating these tools into DM toolboxes.

2 XML-based Technologies

XML, along with other technologies that are built on top of XML constitute core technologies for the proposed DM toolbox architecture.

2.1 Communication Protocols

XML-RPC (XML-Remote Procedure Call) [19] is a very simple XML and HTTP based protocol. It allows for making procedure calls and transmittal of complex data structures between programs running on disparate operating systems, and in different environments. XML-RPC implementations are available for virtually all operating systems, programming languages, dynamic and static environments. SOAP (Simple Object Access Protocol) [14] is another platform-independent and popular XML and HTTP based protocol. SOAP is a superset of XML-RPC, but they are not fully compatible. SOAP is mostly used for accessing services, objects, and servers over the Internet.

SOAP and XML-RPC are two communication protocols that are not only platform-independent and loosely coupled, but also seamless in terms of implementation. They can be used for communication between different DM tools, and between DM tools and knowledge repositories. The proposed DM toolbox can use these protocols to access all XML compatible DM tools over the Internet. Such architectures support ease of development, distribution and customization since both protocols are open source, Internet based, and have strong industry support.

2.2 Predictive Data Models Language

PMML (Predictive Model Markup Language) [13] is the XML-based language for defining predictive data models, which was developed by the Data Mining Group [6]. It allows for vendor-independent defining and sharing of data models between PMML compliant applications. PMML is a unified language for storing knowledge generated by different DM methods, and thus helps in removing incompatibilities caused by using proprietary formats. The PMML currently supports several DM models like decision trees, naive Bayesian models, regression models, sequence and association rules, neural networks, and center- and distribution-based clustering algorithms [6]. For details on PMML see [10]. Products from many major vendors like IBM, Oracle, SPSS, NCR, Magnify, Angoss and others already currently support PMML. The usability of XML for storing, retrieving and using the domain knowledge via the use of PMML is described in [2].

2.3 Service Description and Discovery Tools

WSDL (Web Services Description Language) [18] is the XML based format for describing web based services. WSDL allows the description of services and their messages regardless of what message formats or network protocols are used for communication. Most commonly, WSDL is used in conjunction with SOAP, HTTP, and MIME. WSDL provides an interface definition, abstract semantics and implementation definitions, and end points and network addresses, where the described web service can be invoked.

UDDI (Universal Description Discovery and Integration) [15] is a platform-independent framework based on XML, SOAP, HTTP and Domain Name System (DNS) protocols. UDDI is an industry initiative that serves the purpose of describing, discovering and integrating web-based services. It defines an API for publishing and searching web services, and can be applied to both XML and non-XML services. UDDI has strong industry support; currently over 300 companies support the UDDI initiative.

3 The DM Toolbox Architecture

3.1 Overview

The idea of implementing DM toolboxes arises from a simple observation that no single DM tool performs well on all types of data. There are many DM software implementations of DM methods available. Comparison of 43 existing implementations of DM methods was done in [9]. The proposed DM toolbox architecture tries to solve the problem of integration of available DM tools [9].

The main idea behind the proposed architecture is that a DM toolbox would consist of DM tools that would be published as web services [5]. Currently, several programming platforms already offers tools for publishing software as web service, providing WSDL description for it, and enabling SOAP based communication between the published services.

The DM toolbox architecture will use:

- UDDI to publish, find, and use DM tools that are published as web services. We envision that because of very strong industry support, DM researchers and companies will publish their software online and use UDDI to advertise it
- WSDL to describe formal DM tools and toolboxes
- SOAP as the protocol for interactions within a toolbox
- XML as the universal data format. XML can be used directly for storing data, and by using the PMML standard it can be also used to store discovered knowledge
- Internet as the communication medium

The proposed architecture implements the DM toolbox as a web client that accesses DM tools that are published as web services using a UDDI service, WSDL description mechanism, and SOAP as the communication protocol. The execution model of the proposed architecture consists of the following steps:

- The toolbox accepts the data from the user
- The toolbox checks availability and description of online-enabled DM tools (published as web services) using UDDI
- The toolbox invokes the tools that can provide meaningful results for currently processed data
- The toolbox serves data or portions of user data to the chosen DM tools for processing
- The chosen tools process the data, and return results to the toolbox
- The toolbox analyses and integrates the results and serves them back to the user. The results can be stored in a repository that uses a PMML language model, which will help to integrate knowledge coming from different sources.

The detailed execution model of the proposed architecture is shown in Figure 1 (black arrows denote the runtime communication, gray arrows denote the design-time or dynamic communication).

Using the proposed architecture, the DM toolbox can access and use several DM tools that are published as web services. The toolbox is able to process and integrate results sent back from the tools and store them in the knowledge base. We think that the proposed architecture will be widely accepted because of several factors:

- ease of developing web services, using SOAP protocol and UDDI; see implementation example below

- flexibility and wide applicability of the proposed architecture. The architecture can work cross-platform, uses popular Internet based protocols, supports distribution and is powered by dynamic finding of all DM models applicable to a specific task
- use of open standards that have broad industry support

The DM toolbox architecture is shown in Figure 2.

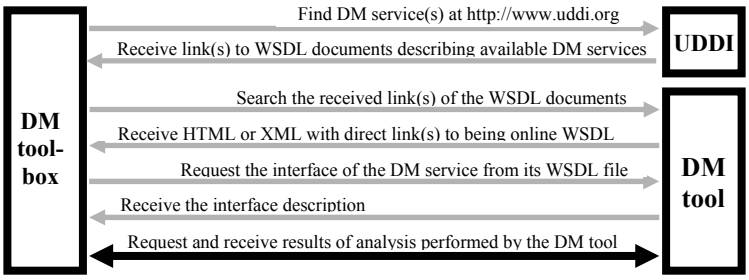

Fig. 1. The detailed execution model of the proposed DM toolbox architecture

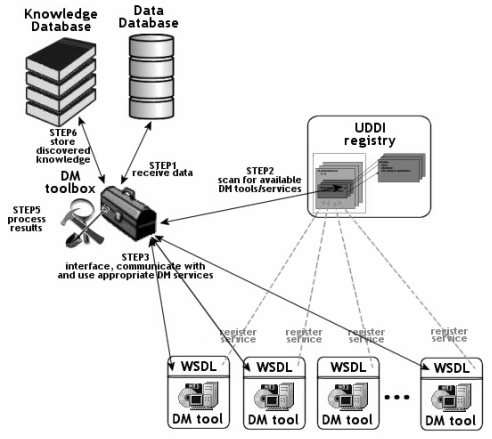

Fig. 2. DM toolbox architecture using Internet via HTTP, XML, SOAP and UDDI

3.2 Implementation

Web services are commonly implemented in any of several different programming languages, on many different platforms, and using a wide variety of application development tools. For the purpose of this demonstration, a sample web service was generated using C# and Visual Studio .Net. The DMServices web service was implemented with four methods that take and return string collections, but actually have no other functionality. They are simply to illustrate the process of finding and accessing web services through UDDI. The service was registered in the UDDI registry at run time using techniques described in [11]. The building and consequent registration of web services is not the topic of this paper.

In order to dynamically locate a web service, UDDI provides a framework for simplifying the discovery process [16]. Two key structures are the tModel and businessService. Often, the tModel for a service is equivalent to the service's WSDL document. The businessService is an XML-based structure used to represent details of the service in UDDI. How and where the service is accessed is provided in a structure known as the bindingTemplate, and is nested within the businessService. An example of the businessService for the DMServices web service is shown in Figure 3.

Using the XML structures defined by UDDI [16], we created a user interface for finding web services based on user-provided search strings for the service name and description. We again used C#, Visual Studio .Net, and the UDDI SDK 1.75 [17]. The development tools used allowed us to quickly develop an interface between the software and the UDDI registry, but are by no means the only way to do so.

```
<BusinessService (…)>
  <name>DMServices
  </name>
  <description (…)>Provides some DM functions.
  </description>
  <bindingTemplates>
    <bindingTemplate (…)>
      (…)
      <accessPoint URLType=
       "http">http://isl2.cudenver.edu/DM.asmx
      </accessPoint>
      (…)
      <tModelInstanceDetails>
        <tModelInstanceInfo tModelKey="…">
          (…)
        </tModelInstanceInfo>
      </tModelInstanceDetails>
    </bindingTemplate>
  </bindingTemplates>
  (…)
</BusinessService>
```

Fig. 3. A portion of the businessService XML description of the DMServices web service.

Using our tools, we simply took the user's input string(s), searched across all business listings, inspecting the service names and descriptions associated with each business for a match to the user's input. As matching services were discovered, we set a boolean flag to display the resulting matches in a tree structure for the user to explore. The primary functionality can be seen in the code snippet of Figure 4.

```
Regex nameRegex = new Regex(".*" + textBox1.Text +
".*",RegexOptions.IgnoreCase);
Regex descRegex = new Regex(".*" + textBox2.Text +
".*",RegexOptions.IgnoreCase);
FindBusiness fb = new FindBusiness();
fb.Name = "%"; // Wildcard to search across all business names
try {
     BusinessList bl = fb.Send();
     treeView1.Nodes.Clear();
     if (bl.BusinessInfos.Count == 0)
       treeView1.Nodes.Add("No services found!");
     foreach (BusinessInfo bi in bl.BusinessInfos) {
       GetBusinessDetail gbd = new GetBusinessDetail();
       gbd.BusinessKeys.Add(bi.BusinessKey);
       BusinessDetail bd = gbd.Send();
       foreach (BusinessEntity be in bd.BusinessEntities) {
         TreeNode businessNode = new TreeNode(be.Name);
         foreach (BusinessService bs in be.BusinessServices) {
           bool nameMatch = false;
           bool descMatch = false;
           if (nameRegex.Match(bs.Name).Success)
             nameMatch = true;
           foreach (Description d in bs.Descriptions) {
             if (descRegex.Match(d.Text).Success)
                                descMatch = true;
...}...
```

Fig. 4. Example of the UDDI search code

Finally, in Figure 5, we show our tool and the results it found in the UDDI using the service description search string "data mining". As one can notice, two results were found. This is because we registered our service under two names, showing the possibility that more than one service may easily be found to perform a desired task.

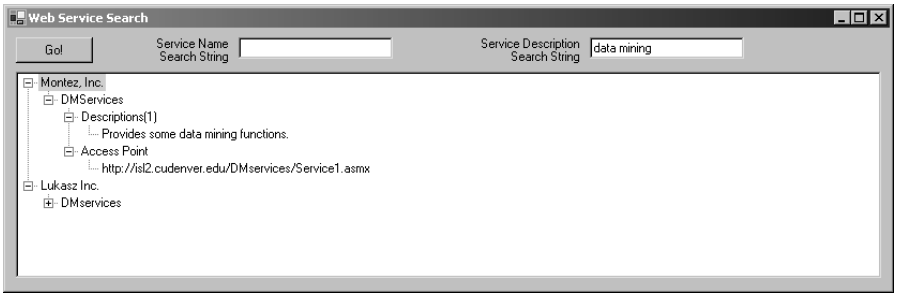

Fig. 5. Sample output for performing a search for data mining web services using UDDI

4 Conclusions

This paper presents a web-based architecture for a DM toolbox, which uses XML based technologies like XMP-RPC, SOAP, PMML, WSDL, and UDDI. The architecture uses

SOAP and XML-RPC for communication between the toolbox and DM tools used by it, UDDI for dynamic searching and accessing of the tools, and PMML as the standard for storing discovered knowledge. The architecture has several advantages, like using open standards that have broad industry support, ease of development, flexibility, wide applicability, and support for distribution. We think that there is a strong need for a tool that aims to integrate DM solutions since, for the last several years, the DMKD community has generated solutions that were not applicable in a broad context.

XML based technologies make it possible to build DM toolboxes that can dynamically integrate multiple DM tools, to build standardized knowledge repositories, and to communicate and interact between different DM tools. Development of such toolboxes will result in wider applicability of DM products.

References

1. Bray, T., Paoli, J., & Maler E., (2000) Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, http://www.w3.org/TR/2000/REC-xml-20001006
2. Büchner, A.G., Baumgarten, M., Mulvenna, M.D., Böhm, R., & Anand, S.S., (2000) Data Mining and XML: Current and Future Issues, *Proc. of the First International Conference on Web Information Systems Engineering (WISE'00)*, 127-131, Hong Kong
3. Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., & Zanasi, A. (1998), *Discovering Data Mining: From Concepts to Implementation*. Perentice Hall
4. Cios, K.J., Teresinska, A., Konieczna, S., Potocka, J., & Sharma, S. (2000) , Diagnosing Myocardial Perfusion from PECT Bull's-eye Maps - A Knowledge Discovery Approach, *IEEE Engineering in Medicine and Biology Magazine*, Special issue on Medical Data Mining and Knowledge Discovery, 19(4), 17-25
5. Cios, K. J., & Kurgan, L. (2002), Trends in Data Mining and Knowledge Discovery, In: Pal N.R., Jain, L.C. and Teoderesku, N. (Eds.), *Knowledge Discovery in Advanced Information Systems*, Springer, in print
6. DMG (2001), The Data Mining Group, http://www.dmg.org/
7. Fayyad, U.M., Piatesky-Shapiro, G., Smyth, P., & Uthurusamy, R. (1996), *Advances in Knowledge Discovery and Data Mining*, AAAi/MIT Press
8. Fayyad, U.M., Piatetsky-Shapiro, G., & Smyth, P. (1996), Knowledge Discovery and Data Mining: towards a unifying framework, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD96)*, Portland, OR
9. Goebel, M., & Gruenwald, L. (1999), A Survey of Data Mining Software Tools, *SIGKDD Explorations*, 1(1), 20-33
10. Grossman, R.L., Bailey, S., Ramu, A., Malhi, B., Hallstrom, P., Pulleyn, I., & Qin, X. (1999), The Management and Mining of Multiple Predictive Models Using the Predictive Modeling Markup Language, *Information and Software Technology*, 41(9), 589-595
11. Januszewski, K. (2001), Web Service Description and Discovery Using UDDI, Part I, MSDN Library, http://msdn.microsoft.com/library/default.asp
12. Kurgan, L., Swiercz, W., & Cios, K.J. (2002), Semantic Mapping of XML Tags using Inductive Machine Learning, *Proceedings of the 2002 International Conference on Machine Learning and Applications* (ICMLA'02), pp.99-109, Las Vegas, NV
13. PMML (2001), *Second Annual Workshop on the Predictive Model Markup Language*, San Francisco, CA
14. SOAP 1.1 (2000), W3C Note, http://www.w3.org/TR/SOAP/
15. UDDI (2001), Universal Description, Discovery, and Integration (UDDI) specification, version 2.0, http://www.uddi.org/
16. UDDI (2001), *Using WSDL in a UDDI Registry 1.06: UDDI Working Draft Best Practices Document*, http://www.uddi.org/bestpractices.html
17. UDDI SDK (2001), Microsoft UDDI Software Development Kit 1.75, http://msdn.microsoft.com/downloads/default.asp?URL=/downloads/sample.asp?url=/msdn-files/027/001/814/msdncompositedoc.xml
18. WSDL (2001), Web Services Description Language (WSDL) 1.1, W3C Note, http://www.w3.org/TR/wsdl
19. XML-RPC (2001), UserLand Software, Inc., http://www.xmlrpc.com/