MEDCollector: Multisource Epidemic Data Collector

João Zamite, Fabrício A. B. Silva, Francisco Couto, Mário J. Silva

LaSIGE, Faculty of Science, University of Lisbon
epiwork@lasige.di.fc.ul.pt

Abstract. This paper analyzes the requirements and presents a novel approach to the development of a system for epidemiological data collection and integration based on the principles of interoperability and modularity. Accurate and timely epidemic models require the integration of large, fresh datasets. Thus, from an e-science perspective, collected data should be shared seamlessly across multiple applications. This is addressed by our approach, MEDCollector, trough workflow design enables the extraction of data from multiple Web sources. The mapping of extracted entities to ontologies will guarantee the consistency within gathered datasets, and therefore enhance epidemic modeling tools.

Key words: Epidemic Surveillance, Data Collection, Information Integration, Workflow Design

1 Introduction

The study of epidemic disease propagation and its control is highly dependent on the availability of reliable epidemic data. Epidemic surveillance systems play an important role in this subject, extracting exhaustive information with the purpose of understanding disease propagation and evaluating its impact in public health through epidemic forecasting tools.

International organizations, such as the World Health Organization (WHO), have epidemic surveillance systems that collect infectious disease cases. However, although official disease statistics and demographics provide the most reliable data, the use of new technologies for epidemic data collection is useful to complement data already obtained from national reporting systems.

In recent years, several projects have researched the use of the Web as a platform for epidemic data collection. The systems developed by these projects gather epidemic data from several types of sources [1], such as query data from search engines [2], internet news services[3] and directly from users [4]. Alternative sources for epidemic data are social networks, e.g. Twitter [5], which are forums where people share information that can be accessed as Web services. These alternative sources of information can be used to identify possible disease cases, or at least provide a glimpse about the propagation of a disease in a community.

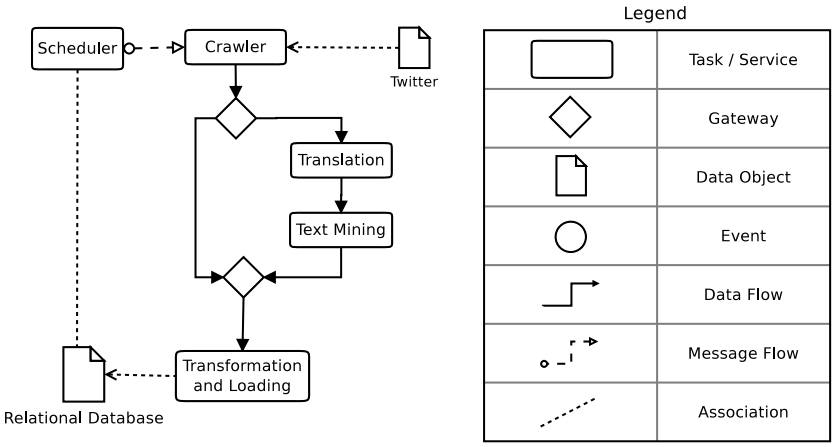


Fig. 1. Example of a workflow, to extract messages from twitter, text-mine them, and insert both the message and extracted information into the database.

The aforementioned systems use different methods for data presentation. Therefore, an integrative effort is needed to consolidate their data so it can be used in e-science data analysis.

This need is highlighted by the EPIWORK project, a multidisciplinary effort to develop an appropriate framework for epidemiological forecasting [6]. This framework is aimed at the creation of a computational platform for epidemic research and data sharing, which will encompass the design and implementation of disease incidence collection tools, the development of large-scale data-driven computational and mathematical models to assess disease spread and control.

An approach to extract and integrate data from multiple sources is the definition of workflows, which enables the composition of collection mechanisms using web services (see Fig. 1). Following this approach, this paper describes the development of MEDCollector, a system for information extraction and integration from multiple heterogeneous epidemiological data sources. MEDCollector is a component of EPIWORK's information platform, the *Epidemic Marketplace* [7]. MEDCollector enables the flexible configuration of epidemic data collection from multiple sources, using interoperable services orchestrated as workflows. Collected data can then be packed into datasets for later use by epidemic modeling tools. Through the use of Web standards for data transmission, the system enables seamless integration of web services to extend its basic functionality. This system gathers and integrates data from multiple and heterogeneous sources, providing epidemiologists a wide array of datasets obtained from the Web using its own data services, in addition to traditional data sources.

The remainder of the paper is organized as follows: Section 2 provides insight into previous related work; Section 3 is an assessment of the system requirements

for an epidemic data collector; Section 4 presents our implementation; Section 5 describes a short example of the system use, showing the creation of a workflow for epidemic data collection from social networks; Section 6 presents the conclusions and perspectives for future work in MEDCollector.

2 Related Work

E-science involves the use of web, computational and information technologies to achieve scientific results [8]. It requires the development of middleware and networking technologies to perform tasks, such as data acquisition and integration, storage, management, mining and visualization. This provision of scientific environments allows global collaboration by enabling universal access to knowledge and resources. Recently, a number of initiatives, such as myGrid and its workflow environment Taverna[9] [10] in bioinformatics, and EGEE and DEISA [11] in multiple domains, have been bridging the gap between the need for computational tools and their seamless integration through the use of standards and interoperable services.

The Web presents a valuable source for collecting epidemic data, but it requires coping with a variety of formats, ranging from free text to XML documents. Disease reporting services, like the ProMED-mail newsletter [12], EuroFlu and reports from the European Center for Disease Prevention and Control (ECDC) are useful sources of epidemiological data. The ProMed-mail newsletter, maintained by the International Society for Infectious Diseases, is a notification service that sends their registered users information about new disease cases via e-mail. EuroFlu.org, a WHO website, and the ECDC's European Influenza Surveillance Network (EISN) [13] publish weekly reports on the activity of Influenza-like diseases.

Internet Monitoring Systems (IMS) can retrieve data using two distinct approaches: passive data collection and active data collection. Systems that use passive data collection mechanisms, such as Gripenet [4] and Google Flu Trends [2], provide interfaces to their users who voluntarily submit their data. On the other hand, active data collection systems, such as Healthmap [1], use crawlers that browse the Web through hyperlinks and existing web services.

The IMS Gripenet depends directly on the active participation of its voluntary users, which receive weekly newsletters about influenza activity and are requested to fill out a form about the presence, or not, of influenza symptoms during the past week. This system was based on Holland's Influenzanet [14] model and is currently implemented on seven other countries: Belgium, Italy, Brazil, Mexico, United Kingdom and Australia and Canada.

Google Flu Trends is a system that performs analysis on user queries to the Google search engine and has been shown to predict influenza activity within two weeks prior to the official sources for the North American Population. This system is currently being extended to cover other countries around the world. Both Google Flu Trends and the previously mentioned IMS collect data directly from their users.

Healthmap, takes a different approach. It is a worldwide epidemic data presentation website that represents disease cases, mostly of contagious diseases, gathered from different sources. These sources can be diverse in nature, ranging from news casting services to official epidemic reports, and have different degrees of reliability. Disease and location information is extracted via a text processing system and presented on a map via the Google Maps API.

3 Epidemic Data Collector Requirements

An epidemiological data collector should follow a set of principles and requirements enabling extensible data collection and the creation of consistent, integrated datasets, while coping with the heterogeneity associated with its sources.

- *Active data collection.* Harvesting Web data automatically using available web services and their application programming interfaces. This enables data collection from sources like Twitter, Google Flu Trends and EISN reports. Depending on the source the harvesting mechanism collects an entire message containing the name of a disease for further processing or harvest epidemiological estimates known to be published at the defined source.
- *Passive data collection.* Receiving data posts from a number of sources, including news feeds and email subscriptions (e.g. ProMED-mail). Data received by passive data collection mechanisms requires structuring before being integrated and loaded to the system.
- *Flexible Scheduling.* To cope with different periods of data update in each source the system must enable the scheduling of the activation of each data collection mechanism at their sources.
- *Local Storage.* Different data sources have variable data availability times, and data may only be available for some time period at certain sources, if any. An approach to solve the problem associated with dealing with volatile data, as well as the temporal disparity of data sources is to locally store all the data retrieved by the system in a local dedicated relational database.
- *Ontology Referencing.* Enables the use of vocabularies when referencing entities in the spatial and health domains. The use of ontologies enables the disambiguation of named entities, the mapping of entities with multiple references across data sources, and the establishment of hierarchical relationships among entities. This hierarchy becomes particularly relevant when using geographic referencing. For instance, with the support of a geospatial ontology, we could relate cities with their respective countries. This enables the aggregation of data defined for specific levels to higher levels, e.g. disease cases identified in London can be used in the United Kingdom domain.
- *Modularity and Configurability.* An epidemic data collector that retrieves data from the Web requires a degree of flexibility in order to cope with changes or additions in its sources. By adopting a SOA architecture

[15], the system has its functionality distributed through discrete units, or services Orchestrations, or workflows, enable the design of data flow sequences between the different services. Configurable workflows enable the reconfiguration and addition of new services whenever necessary by defining how services are interconnected and how information is transmitted between them [16]. By defining services with a set of configurable inputs and outputs based on Web Standards they become highly interoperable which improves the flexibility of workflow creation.

4 Implementation

The MEDCollector implements the above requirements through the dynamic design of service workflows. It is inspired in an initial prototype we developed to collect messages from Twitter containing disease and location names [17].

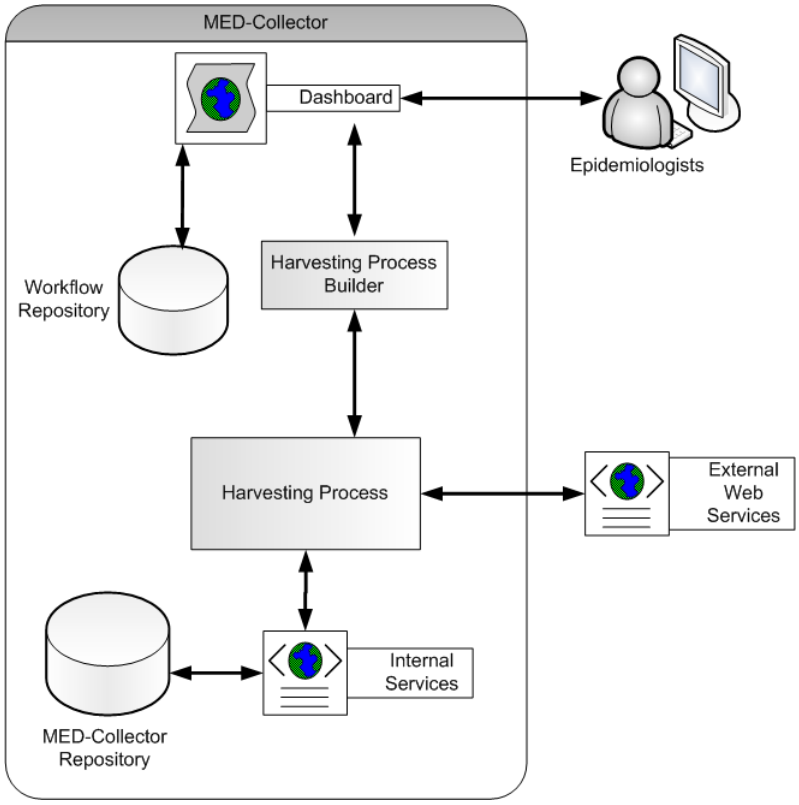


Fig. 2. MEDCollector's basic architecture.

The architecture of the MEDCollector is represented in Fig. 2. Its main system components are:

- *Dashboard*. Provides user-interface capabilities to the system, enabling the user to define harvesting processes and to monitor currently deployed processes.
- *Workflow Repository*. Stores workflows designed in the Dashboard.
- *Harvesting Process Builder*. Converts designed workflows to deployed Harvesting processes.
- *Harvesting Processes*. Processes that orchestrate communications between multiple services, both internal and external, to perform data collection from external sources accordingly to workflow definition.
- *Internal Services*. Provide basic system functionalities and interact with the MEDCollector Repository.
- *MEDCollector Repository*. Stores all the data collected by the system.

The MEDCollector Repository stores both the data collected from the Web and data collection schedules. It is implemented as a MySQL relational database. For clarity in the description of this repository's implementation, we present it as storage for two types of data: a Case Data and a Scheduling Data.

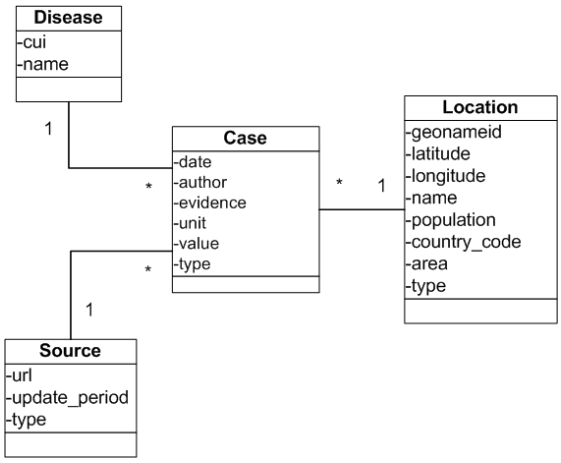
4.1 MEDCollector Repository

Case Data The collected data are organized in the repository under a classic Data Warehouse star schema [18]. The fact table, shown in Fig. 3(a), has the following dimensions describing the cases:

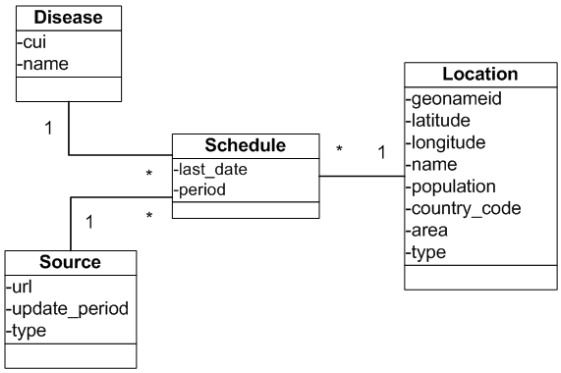
- *Disease*: reference to the disease name and a concept unique identified (cui) that identifies that disease in the Unified Medical Language System (UMLS) [19].
- *Location*: reference to a location monitored by the system, including a geonameid which identifies that location in the GeoNames ontology [20].
- *Source*: reference to the monitored source, its URL and, in some cases, the update interval for that source.

Scheduling Data The schedule of data harvesting operations, represented in Fig. 3(b), has an identical organization with the harvesting events as the fact table and the same dimension tables.

The information in the repository is accessible through a series of services for inserting and selecting. The database currently includes all countries in the world and their capitals as well as a set of 89 infectious diseases.



(a) Case Data



(b) Scheduling Data

Fig. 3. UML class diagram of the MEDCollector Repository.

4.2 Dashboard

The Dashboard provides the user interface to add new data sources or further ways to process information through the addition of new services or by adjustment of service parameters.

The Business Process Execution Language (BPEL) [21] is a workflow design language that uses XML to describe the interaction between services. The BPEL process, corresponding to a designed workflow, is itself a service and can be interpreted and executed by a BPEL engine.

One of the difficulties with the use of BPEL lies on the need of methods for creating process definitions by non-technical users [16], this requires the MEDCollector to have a user interface.

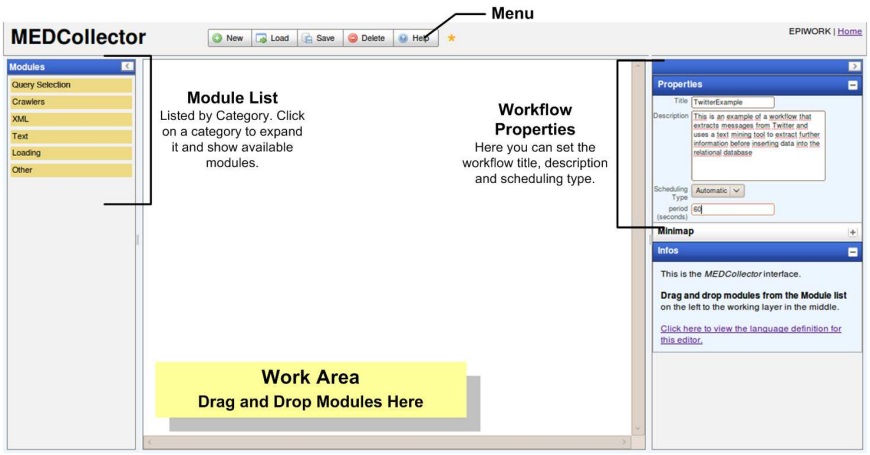


Fig. 4. Global view of the Web Interface implemented using WiringEditor and description of its components.

We considered scientific workflow systems like Taverna, but they require the direct definition of WSDLs and communication via SOAP. In addition, these systems currently do not offer on-browser interfaces, requiring users to go through local installation and configuration processes prior to using the software.

WireIt[22] enables the definition of a "Visual Language" that specifies modules, their inputs and outputs, which represent services in MEDCollector. It is also bundled with a single-page editor that enables the definition of workflows through a wirable interface, see Fig. 4. The wirable interface consists of drag-and-drop elements which can be connected with wires between their inputs and outputs. Workflows designed in this interface are saved to the Workflow Repository.

WireIt is an open source JavaScript library for the creation of web wirable interfaces similar to Yahoo! Pipes [23] and uDesign [24]. WireIt uses Yahoo! User Interface library 2.7.0 for DOM and event manipulation and is compatible with most web browsers.

The Dashboard also enables the user to specify scheduling properties for each workflow, such as the period to wait between workflow runs.

4.3 Harvesting Processes

When the user saves a workflow from the interface, the Harvesting Process Builder receives a JSON representation of the workflow from the interface and creates the files necessary to deploy a Harvesting Process for running in the BPEL engine. Each of these processes orchestrate communications between basic services that perform the data collection accordingly to workflow definitions. A

Harvesting Process consists of a process descriptor, a BPEL process and a WSDL document.

We use Apache ODE (Orchestration Director Engine) [25] to execute our Harvesting Processes. Apache ODE provides several extensions to standard BPEL engines including XPath 2.0 support, for easier variable assignments, and an HTTP binding extension that enables direct connection to RESTful Web Services. This engine also provides an interface that enables monitorization of currently deployed processes.

4.4 Services

Internal services represent the basic operations performed by the system. These can be information collection services, text mining services, transformation services, scheduler services, and others.

We have implemented the following in the current version of the MEDCollector:

1. *Query Selection Services.* These specify when each disease is actively monitored in what locations and sources. There are two types of query selection services, user defined query and a priority based query selection service. In the first the user directly specifies what query to run, and therefore which disease to monitor at what location and source. The latter uses the stored Scheduling Data, selecting the period and last search date values for each disease-location-source triple and outputting the triple with the highest positive priority value according to the formula:

   $$priority = date - last\ search\ date - period$$

   If there are no positive values the service sends a fault message that is caught by the BPEL Process, stopping it and scheduling another run the next day. These triples can be filtered by source, location or disease, in order to create processes with specific scopes, e.g. influenza in Portugal with Twitter as a source.
   The priority is related to the amount of data collected in the past. Each week, a MEDCollector utility re-evaluates the update intervals according to the previous detected case entries:
   - Daily period: every triple with more than 1 entry the previous week.
   - Weekly period: every triple with more than 1 entry the previous two weeks and 1 or less entries the previous week.
   - Fortnightly period: every triple with more than 1 entry the previous month and 1 or less entries the previous two weeks.
   - Monthly period: every triple that does not fit the criteria mentioned above.
2. *Active Data Harvesting Services* retrieve content through APIs or provided URLs. These services structure collected cases to a XML schema compatible with other MEDCollector services. Scheduler services coordinate when these harvesting are actively querying the data sources.

3. *Passive Collection Services* receive data posted by disease reporting services and other sources such as email subscriptions.
4. *Text related services* include regular expression text mining services that search strings for patterns, text mining services and translation services that use the Google Language API [26]. Text Mining services, receive a message from harvesting and passive collection services and extract further information from them. They use a set of rules to mine previously retrieved messages for evidence that specifies number of cases, deaths or estimates.
5. *Database Loading* is done through a service that receives a XML message and accordingly performs an insert in the local relational database.
6. *Data Structuring Services* that provide functionalities such as data structure transformation and access to specific data elements to improve the flexibility of the workflows. This enables the use of external web services in the sequence flow by transforming their data into compatible data types and formats.

Furthermore, two interface modules enable the invocation of external services through REST or SOAP. These modules enable the addition of new functionalities to the system, such as gathering data from new sources or interaction with other applications.

5 Usage Example

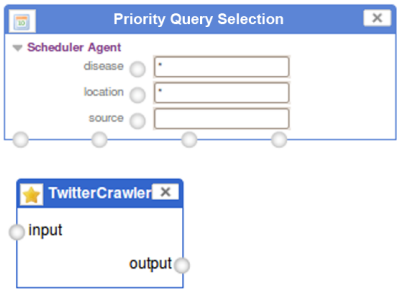
This section illustrates the creation of workflows in MEDCollector, using again the extraction of messages from the Twitter Social Network as an example. Fig. 5 and 6 presents a step-by-step description on how to create a workflow to extract messages from Twitter, translate and mine them for epidemiological data.

To create a workflow the user starts by adding a Query Selection Service to the Interface Work Area (Fig. 5 a). To retrieve messages from Twitter the user connects the Query Selection XML output to the crawler input and specifying the source in the Query Selection input (Fig. 5 b). The Query Selection also enables filtering by disease or location so that users can specify specific entities to be searched for.

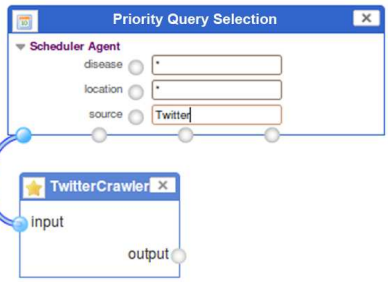
The users can also text-mine the messages extracted. Since the available text mining service is implemented only for the English language, the user uses a translation service (Fig. 5 c). Since the user does not know which language each message is in he/she leaves "input language" blank. The Translation service will identify what language the message is in prior to translating it. The user chooses the desired output language - "en" since he wants the output to be in English - then he/she connects the translation service output to the text mining service (Fig. 5 d).

The user can store both the raw messages as well as the occurrences extracted from text mining by connecting both the output of the crawler and the text mining service to a Merge Gate and then connecting it to a Loading Service (Fig. 6).

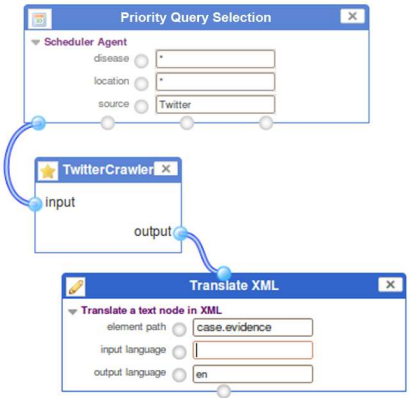
After pressing "Save" on the interface's menu, a JSON message is sent to the BPEL Process Builder, which deploys the process to be run by Apache ODE.



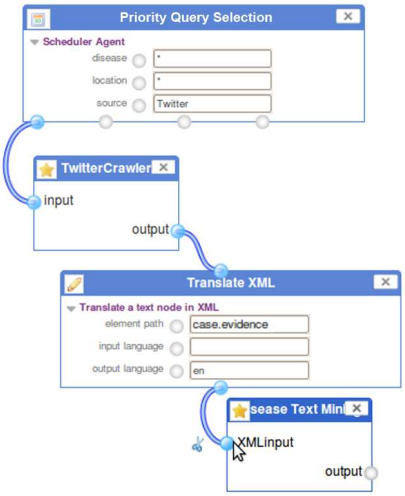
(a) Start by adding a Scheduler and an Harvesting Service to the Work Area.



(b) Connect the XML output of the Scheduler to the Harvesting Service.



(c) To translate the text of the extracted messages use the Translation Service.



(d) After translation a text mining service can be used to extract further information.

Fig. 5. a) through d) - Step-by-Step creation of a workflow that Extracts Messages from Twitter to collect epidemiological data.

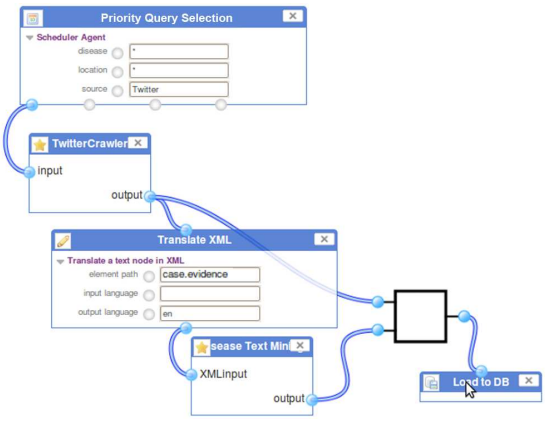


Fig. 6. Final step of the step-by-step workflow creation. To store both raw messages and text-mined cases connect both outputs to a merge game and then connect it to a loading service.

6 Conclusions and Future Directions

The MEDCollector is implemented as a component of the information platform being developed for the EPIWORK project - the Epidemic Marketplace. By enabling the collection and integration of data from multiple web sources, MEDCollector grants epidemiologists with a novel means to gather data for use in epidemic modelling tools.

The Dashboard enables users to dynamically design Web Service workflows through drag-and-drop components, without worrying about technical specifications. This enables users to directly create and modify workflows to customize data collection mechanisms according to their specific needs.

Users can set workflows to extract messages from several sources:

- *Social Network Services*, such as Twitter, where people freely share information. Text messages can be extracted from these sources.
- *Epidemiologic Surveillance Services*, such ProMED-Mail and Google Flu Trends. Each source with different data structures and formats. Users can design workflows to extract messages or disease case estimates depending on the source.
- *New Services*, such as Google News, which report RSS feeds and newsletters containing news relating to specific domains and locations. Text messages related to diseases can be extracted from these sources.

Harvesting processes collect all identified results available at the source. Should a problem occur and MEDCollector processes go offline for a period of time, when it is placed online it will retrieve missing data by continuing with its next scheduled searches.

Through the use of Web Standards for data transmission, MEDCollector enables seamless integration of externally supplied web services, granting extensibility to its basic features.

The next step is the creation a new layer for the interface that accommodates the configuration of dataset creation services. This new layer will be composed mainly of services that select information from the relational database and structure it according to the needs of the users, through XML transformation and selection. This transformation will enable the creation of aggregated and consistent datasets which can be used by other applications.

EPIWORK's information platform includes a dataset repository - the Epidemic Marketplace - where datasets can be stored for later use by epidemic modeling tools. MEDCollector will submit consistent datasets for storage in the Epidemic Marketplace at regular time periods through an upload API method being developed for this repository's mediator.

Another challenge is the development of visualization tools adequate to this data. This will enable epidemiologists to have a preliminary analysis of the data prior to its extraction from the system.

7 Acknowledgements

The authors want to thank the European Commission for the financial support of the EPIWORK project under the Seventh Framework Programme (Grant # 231807), the EPIWORK project partners, CMU-Portugal partnership and FCT (Portuguese research funding agency) for its LaSIGE Multi-annual support.

References

1. J. Brownstein and C. Freifeld, "HealthMap: The development of automated real-time internet surveillance for epidemic intelligence," *Euro Surveill*, vol. 12, no. 11, p. E071129, 2007.
2. J. Ginsberg, M. Mohebbi, R. Patel, L. Brammer, M. Smolinski, and L. Brilliant, "Detecting influenza epidemics using search engine query data," *Nature*, vol. 457, no. 7232, pp. 1012–1014, 2008.
3. A. Mawudeku and M. Blench, "Global Public Health Intelligence Network (GPHIN)," in *7th Conference of the Association for Machine Translation in the Americas*, 2006, pp. 8–12.
4. S. Van Noort, M. Muehlen, A. Rebelo, C. Koppeschaar, L. Lima, and M. Gomes, "Gripenet: an internet-based system to monitor influenza-like illness uniformly across Europe." *Euro Surveill*, vol. 12, no. 7, p. E5, 2007.
5. Twitter. [Online] Available: http://www.twitter.com/. [Accessed December, 2009].
6. EPIWORK. [Online] Available: http://www.epiwork.eu/. [Accessed February, 2009].
7. M. J. Silva, F. A. Silva, L. F. Lopes, and F. M. Couto, "Building a digital library for epidemic modelling," in *Proceedings of ICDL 2010 - The International Conference on Digital Libraries*, vol. 1. New Delhi, India: TERI Press – New Delhi, India, 23–27 February 2010, invited Paper.

8. (2009) e-IRG White Paper 2009. [Online] Available: http://www.e-irg.eu/index.php?option=com_content&task=view&id=40&Itemid=39.
9. P. Li, J. Castrillo, G. Velarde, I. Wassink, S. Soiland-Reyes, S. Owen, D. Withers, T. Oinn, M. Pocock, C. Goble, S. Oliver, and D. Kell, "Performing statistical analyses on quantitative data in taverna workflows: an example using r and maxdbrowse to identify differentially-expressed genes from microarray data," *BMC Bioinformatics*, vol. 9, no. 334, August 2008.
10. A. Gibson, M. Gamble, K. Wolstencroft, T. Oinn, and C. Goble, "The data playground: An intuitive workflow specification environment," in *IEEE International Conference on e-Science and Grid Computing*, 2007, pp. 59–68.
11. M. Riedel, A. Memon, M. Memon, D. Mallmann, A. Streit, F. Wolf, T. Lippert, V. Venturi, P. Andreetto, M. Marzolla, A. Ferraro, A. Ghiselli, F. Hedman, Z. A. Shah, J. Salzemann, A. Da Costa, V. Breton, V. Kasam, M. Hofmann-Apitius, D. Snelling, S. van de Berghe, V. Li, S. Brewer, A. Dunlop, and N. De Silva, "Improving e-Science with Interoperability of the e-Infrastructures EGEE and DEISA," in *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia*, 2008, pp. 225–231.
12. L. Madoff and V. Yu, "ProMED-mail: an early warning system for emerging diseases," *Clinical infectious diseases*, vol. 39, no. 2, pp. 227–232, 2004.
13. European Influenza Surveillance Network (EISN). [Online] Available: http://www.ecdc.europa.eu/en/activities/surveillance/EISN/. [Accessed December, 2009].
14. R. Marquet, A. Bartelds, S. van Noort, C. Koppeschaar, J. Paget, F. Schellevis, and J. van der Zee, "Internet-based monitoring of influenza-like illness(ILI) in the general population of the Netherlands during the 2003 – 2004 influenza season," *BMC Public Health*, vol. 6, no. 1, p. 242, 2006.
15. S. Durvasula, M. Guttmann, A. Kumar, J. Lamb, T. Mitchell, B. Oral, Y. Pai, T. Sedlack, H. Sharma, and S. Sundaresan, "SOA Practitioners' Guide, Part 2, SOA Reference Architecture," 2006.
16. D. Garlan, "Using service-oriented architectures for socio-cultural analysis." [Online]. Available: http://acme.able.cs.cmu.edu/pubs/show.php?id=290
17. L. F. Lopes, J. Zamite, B. Tavares, F. Couto, F. Silva, and M. J. Silva, "Automated social network epidemic data collector," in *INForum - Simpósio de Informática*, September 2009.
18. C. Utley, "Designing the Star Schema Database," *Data Warehousing Resources*, 2002.
19. O. Bodenreider, "The unified medical language system (umls): integrating biomedical terminology," *Nucl. Acids Res.*, vol. 32, no. suppl_1, pp. D267–270, January 2004. [Online]. Available: http://dx.doi.org/10.1093/nar/gkh061
20. GeoNames. [Online] Available: http://www.geonames.org/. [Accessed December, 2009].
21. A. Alves, A. Arkin, S. Askary, B. Bloch, F. Curbera, Y. Goland, N. Kartha, Sterling, D. König, V. Mehta, S. Thatte, D. van der Rijn, P. Yendluri, and A. Yiu, "Web services business process execution language version 2.0," OASIS Committee Draft, May 2006.
22. E. Aboauf. WireIt - a Javascript Wiring Library. [Online] Available: http://javascript.neyric.com/wireit/. [Accessed January, 2010].
23. Yahoo Pipes. [Online] Available: http://pipes.yahoo.com/pipes. [Accessed October, 2009].

24. J. Sousa, B. Schmerl, V. Poladian, and A. Brodsky, "uDesign: End-User Design Applied to Monitoring and Control Applications for Smart Spaces," in *Proceedings of the 2008 Working IFIP/IEEE Conference on Software Architecture*, 2008.
25. T. A. S. Foundation. Apache Orchestration Director Engine. [Online] http://ode.apache.org/. [Accessed January, 2010].
26. Google AJAX Language API. [Online] Available: http://code.google.com/apis/ajaxlanguage/. [Accessed January, 2010].