

Efstathiou Nectarios

Psaroudakis Chrysostomos

Skitsas Michael

Koutras Nikolaos

ADITESS Advanced Integrated Technology Solutions & Services, Nicosia, Cyprus

WARSYS: An efficient data warehouse to support decision support components of health services in major crises

Keywords

critical infrastructure, data warehouse, major incidents

Abstract

Critical infrastructures (CI) are organizations and facilities of major importance and form the backbone of a national economy, security and health. Due to their significance, they may become target of an attack such as terrorist resulting in crisis event. Decision making systems to support and facilitate the operation of emergency health services in major incidents and crises, could play a significant role. In order to support those systems, efficient databases that could store and manage data from different sources, multiple levels of user privileges within a secure environment become a necessity. In this work, we propose and develop the WARSYS component. The main advantage of WARSYS database, is the ability to collect and store information from different sources providing appropriate interfaces. The WARSYS database structure will be developed within IMPRESS framework, with a view of extracting in real time, medical and logistics information from available repositories.

1. Introduction

Critical infrastructures (CI) are organizations and facilities of major importance and form the backbone of a national economy, security and health. Due to their significance, they may become target of an attack such as terrorist resulting in crisis event. In such cases, the role of incident management operators become very crucial. Several challenges arise regarding the ways that such events are handled. The disproportion between the available human/material resources in the response capacity as well as the inherent time constraints of an emergency are two main among other challenges.

To overcome those challenges, IMPRESS project [10] will improve the efficiency of decision making in emergency health operations, which have a direct impact on the quality of services provided to the citizens. This project will facilitate the communication between Health Services at all levels of response and the crisis cycle with the necessary

health care system support, supervision and management of participating organizations. In order to support those systems, Decision Making Systems an efficient database (data warehouse) that could store and manage data from different sources, multiple levels of user privileges within a secure environment become a necessity. In this work, we propose and develop WARSYS system.

WARSYS is responsible for the importing and management of structured data in IMPRESS. It provides an interface for importing data from medical and logistics repositories, such as hospital information systems. The data will be made available to the Data Harmonization Component and then to the other IMPRESS components through the IMPRESS Messaging Bus (AMQ); responsible for the communication between most of the IMPRESS components. Also, WARSYS provides a mechanism to return data to the AMQ. Beyond the communication part, the main responsibility of WARSYS into the IMPRESS solution is to

incorporate an internal database structure that will be used to store and retrieve the extracted data and information generated inside the IMPRESS system. According to the above, WARSYS scope of work will cover two major axes: (a) Data importing and storage from various IMPRESS external resources, and (b) Provision of data to other IMPRESS components as well as the retention of data processing results from IMPRESS DSS components.

The rest of this paper is organized as follows: Section 2 presents the WARSYS Conceptual Architecture. Section 3 describes the data importing process of WARSYS including both data sources outside IMPRESS solution and within IMPRESS. Section 4 presents the WARSYS access mechanisms while Section 5 presents the WARSYS database. Finally, section 6 deals with security issues.

2. WARSYS Conceptual Architecture

WARSYS conceptual architecture is presented in *Figure 1*. The main component of WARSYS architecture is the WARSYS DB Server where all the data are stored. This server is directly connected with a webserver responsible for the communication and providing web services to access the database. As we mentioned earlier, WARSYS is accessible from internal and external IMPRESS users.

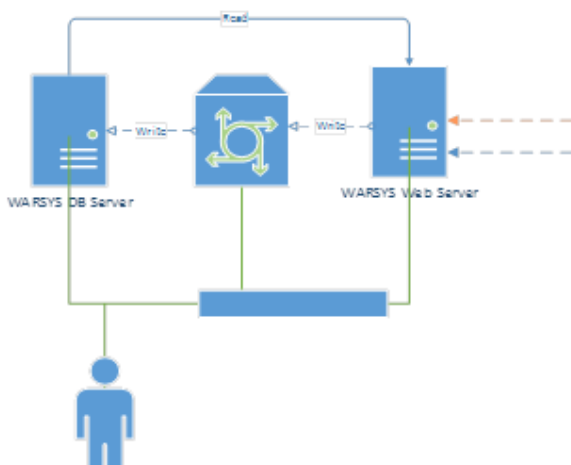


Figure 1. WARSYS Conceptual Architecture

As presented in *Figure 2*, the whole system is protected by secure connections and all the incoming and outgoing traffic goes through a firewall. Internal IMPRESS users are able to store and read data from WARSYS, while external users only store data. Further details about the WARSYS architecture and implementation will be described in the next sections of this document.

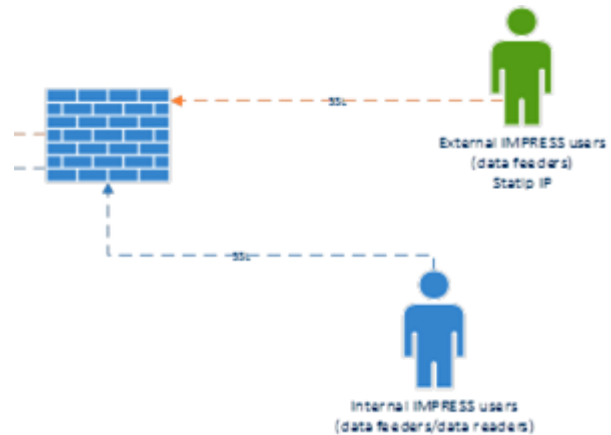


Figure 2. WARSYS Users

3. Data Importing Process

Within this section we will be discussing all aspects involved in the data importing process, including aspects concerning data importing using SQL Server Integration Services (SSIS) from sources lying outside in the IMPRESS system, the different storage areas of WARSYS, as well as the relevant SSIS packages. Additionally, alternative ways of importing data are also analysed. Aspects concerning the implemented WARSYS interfaces, solutions mechanisms, intended users and testing environment are also explained.

3.1. Sources outside IMPRESS using SSIS

Microsoft Integration Services is a platform for building enterprise-level data integration and data transformation solutions. Integration Services are used for solving complex business problems by activating events (copying or downloading files, sending e-mail messages) in response to events, updating data warehouses, cleaning and mining data, as well as managing SQL Server objects and data. The packages can work as standalone tools or in collaboration with other packages to address imposed needs. Integration Services can extract and transform data from a wide variety of sources such as XML data files, flat files, and relational data sources, the discussed services can then manage data loading into one or more destinations.

SSIS is a platform for data integration and workflow applications. The primary usage of SSIS is data warehousing, and serves as a product featuring fast and flexible data extraction, transformation, and loading (ETL). SSIS may also operate as an add-in to Microsoft Visual Studio [13], providing a rich set of built-in tasks and transformations; tools for constructing packages as well as the integration services for running and managing packages.

Developers can create solutions through Integration Services either by making use of the graphical interface (with no need of writing source code) or by programmatically generating custom tasks and package objects, capable of fulfilling the developer's needs.

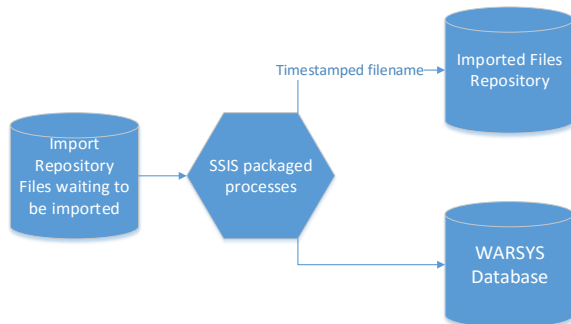


Figure 3. Import Process from External Sources

The diagram in Figure 3 depicts WARSYS procedures as these are followed when data importing from external sources occurs. In particular, WARSYS incorporates three different storage areas: (a) Import repository that is used for files that are waiting to be imported, (b) the WARSYS Database and (c) Imported files repository where files are timestamped with date and time of import as well as accompanied with a log file.

3.1.1. Storage Area Reference 1

Storage Area Reference 1 is the Import Repository where the files are waiting to be imported. This area is a file system storage. Regarding the security and ethics the following are considered: EFS/NTFS with security policy on file/folder level, accessible via SFTP or FTPS or WebDAV only from static IP addresses.

3.1.2. Storage Area Reference 2

The second storage area is the WARSYS database. Security and ethics related parameters are applied upon the database user.

3.1.3. Storage Area Reference 3

The last storage area is the Imported Files Repository. Files within this repository are timestamped with date/time of import among with the log file of import (txt or log with same filename). Similar with storage area 1, this area is also a file system storage and the same security and ethics parameters are applied.

3.2. SSIS Packages

Files to be imported are physically stored on the file system of a WARSYS frontend application server (*Storage Area Reference 1*). Every X^1 amount of time, SSIS packaged processes take place and read from the *File Repository 1* and update the relevant tables in the WARSYS Database (*Storage Area Reference 2*). The following messages constitute an example of logging activity and the produced messages.

- Your {filename} posted on {date-time} was {status} imported.
- There were {number of records} successfully imported.
- {Number of records} failed to import due to errors in your file, please contact your system administration.

The integrity of files to be imported is validated with SSIS and further produced data are appended in the database with the timestamp at the time of submission is also retained.

Whenever a field is updated, the action is logged and the log in *Storage Area Reference 3* is updated. Once the original file (from *Storage Area Reference 1*) is processed, with the SSIS package, it is then moved to *Storage Area Reference 3*, where its log resides. The moved file is then renamed to match its log's name. This is done so that a per-file track record of imports is maintained in the WARSYS database. This is done so that imported files are made available for debugging and logging purposes.

Due to the data criticality, the complexity of different types of data and sources, along with the potential of data corruption, as this might occur either from origin or during transmission, DBA personnel monitors the WARSYS database events, and when required external users are contacted in order to be assisted in the correction of file submissions. This task is trivial for the personnel and should involve DBA administration.

3.3. Alternatives Ways to Import Data

Beyond the usage of SSIS packages, an alternative way to import data is using python [12] scripts. These scripts accept in the form of input the XML Schema (i.e. XSD file) of interest and may generate python data structures (python class definitions) which represent the elements of the input schema. The explained operation may be realized with the use of *GeneratedDS*; an existing python module.

¹ The value of this figure will be determined at a later stage in the project and will be tuned, according to the results as these obtained from the use of the mock solution.

GenerateDS takes as input an XSD schema and generates a data structure that can then be used for handling data complying with the already processed XML schema. The generated structure is extendable and can be modified based on the requirements of the considered project.

3.4. WARSYS Secure Web Interface

One of the potential users of IMPRESS solution and therefore WARSYS mechanisms is hospital personnel. To facilitate the procedure and provide a simple operational framework to such users, where the knowledge and the ability on working with database systems is limited to basics, WARSYS provides a secure web interface. This interface is accessible only from a specific static IP and the authentication will be done by providing credentials (at the minimum username and password) accompanied with a certificate. Once the user is granted access to the WARSYS Database they are offered the ability to:

1. Upload a file. Various file types are allowed to be imported. Such formats can be EDXL xml, csv, excel, etc.
2. Provide the data through a web form. Information will be imported through a number of predefined fields including bed availability, doctor/nurse workforce, facility types etc.

The data from the above described process will be stored in the WARSYS database and posted to the IMPRESS Messaging Bus in EDXL format.

3.5. WARSYS IMPRESS Interface

Further to the WARSYS communication with the external users and sources through the secure web interface, WARSYS provides another interface responsible for the communication between IMPRESS components. In particular, the WARSYS IMPRESS interface is a WARSYS subcomponent which handles, incoming and outgoing information as this enters and exits the IMPRESS Message Bus. WARSYS listens to the Message Bus, through a specific queue or topic, and posts back through the same queue/topic.

The WARSYS IMPRESS interface handles two communication channels. One for the IMPRESS subcomponents internal communication/interaction and one for the Data Harmonization Component (DHC). The DHC implements the data harmonization procedure that is required in order to harmonize the multidisciplinary and heterogeneous datasets of the IMPRESS Platform and provide a semantically homogenized view of the data.

Through the first channel, two main operations from WARSYS are provided. (a) *Read from WARSYS*: Servicing Data requests from other IMPRESS components through the Message Bus by listening and posting (either through OpenWire or STOMP protocols). Messages can be of the following structures: JSON, EDXL or other specified format. Messages are then submitted to the Message Bus. (b) *Write to WARSYS*: data processing results of IMPRESS DSS components are transmitted through the IMPRESS Messaging bus to WARSYS for storage. Data available from the rest of IMPRESS components can be obtained through the same way.

WARSYS acts both as a consumer and producer in terms of communication (Messaging Bus) depending on the type of request. For instance, when a dataset is requested from other components to WARSYS, it acts as a producer and services the request by providing the data. On the other hand, when other components write data on WARSYS, it acts as a consumer. The following example based on the medical triage process presents a case where WARSYS could act both as a consumer and producer.

Medical Triage is the prioritization of patient care during disasters; following mass casualty incidents. During this process patients are color-tagged based on their condition in three categories: (i) green, patients with minor injuries, (ii) yellow, patients that treatment could be delayed and (iii) red, patients that require rapid medical attention; on a side note, black label could be used in cases of death or minimal chance for surviving. When medical staff arrives at the point of disaster and triage is deployed, a list of patients with their respective colours is created, data are transmitted to IMPRESS from INCIMAG via the Messaging Bus. INCIMAG is the Incident Management Tool of the IMPRESS solution. Under this scenario, WARSYS is a consumer and listens to the triage results. Then, WARSYS will update all the relevant components and supply the triage results having the role of the producer; by posting back to the AMQ. Further details on the producer and consumer role of WARSYS will be presented in later paragraphs.

The WARSYS IMPRESS interface incorporates a python compiled daemon that reads data from the WARSYS database (using SQL queries) and then posts back to the Message Bus using the queue WARSYS; and/or other topic that is to be defined later in the project. On the other hand, a similar script, python daemon, listens to the Messaging Bus on the queue WARSYS (or any other queue/topic that might be defined later in the project) and by using an ODBC connector sends data to WARSYS,

either directly via a MSSQL connection or through an intermediating SSIS package. Due to the flexibility and a modular based design of WARSYS system, the usage of other messaging systems as well as the usage of API calls is feasible.

Regarding the second communication channel handled by WARSYS, the Interface with the DHC, a connection from the DHC to WARSYS will be implemented through a TCP/IP connection using the JDBC protocol.

4. WARSYS Access Mechanisms

4.1. External Write Mechanisms

External write mechanisms to WARSYS system will be triggered by external IMPRESS user actions. In this case, data files uploaded by external users are firstly stored in their “personal” repository (*Storage Area Reference 1*), then processed with WARSYS SSIS and imported afterwards. The mechanisms consisting WARSYS intelligent Import module are the following:

- SSIS Package for HL7-RIM [8], [9]
- SSIS Package for CAP [3]
- SSIS Package for EDXL-RM [5]
- SSIS Package for EDXL-DE [5]
- SSIS Package for EDXL-TEP [7]
- SSIS Package for EDXL-TEC [6]
- SSIS Package for EDXL-HAVE [4]
- SSIS Package for CSV [2]
- SSIS Package for XML [14]

Each file type to be imported in WARSYS, triggers a specific mechanism. The next steps are a high level description of the followed procedure for a specific mechanism.

1. Checks the file against integrity and files specifications
2. Processes the file by opening and reading its content
3. Imports the data in the database; while keeping the original file having the timestamp as part of the filename and then records all monitored processes in a log file.

4.2. Internal Read Mechanisms

Internal IMPRESS users/subcomponents can read from WARSYS using the following procedure.

1. Submit a WARSYS tagged request to the Message Bus; providing their data request using the predefined description. This information goes in the messaging bus in textual format and has as its destination a queue/topic.

2. When WARSYS gets the request, from a particular user/subcomponent: Checks whether the component has the privilege to request the specified dataset and in the case where access is granted, the system matches the description of the request to a specific procedure

WARSYS runs the procedure and posts its outcome to the messaging bus. For this reason, a specific table in WARSYS database is implemented and is responsible to hold this “request description” against database stored procedures.

4.3. Internal Write Mechanisms

IMPRESS components can post processing results or other data to WARSYS, through the Messaging Bus. WARSYS listens on a messaging bus queue name or topic (i.e. WARSYS), and proceeds with data write processes as previously discussed.

5. WARSYS Database

The WARSYS database is the core of WARSYS as it provides data storage utilities that make stored data available to the WARSYS subcomponents and IMPRESS components.

5.1. Data from Medical and Logistics Systems

It is already noted that WARSYS manages the import of structured data in the IMPRESS architecture. More specifically, it provides an interface for importing data from medical and logistics repositories and also allows the extraction of data and subsequent – real time – provision through the Data Harmonization Component and the IMPRESS Messaging Bus (AMQ).

WARSYS incorporates an internal database structure that is used for the storage and retrieval of extracted data and generated information. This information shall be made available to authenticated components/systems, within the IMPRESS architecture, while WARSYS ensures, both internally and externally, security and data protection. Moreover, WARSYS also allows dynamic access to patient information and medical care facility information; the level of detail in retrieved information is dependent on the level of user privilege.

The external repositories from where WARSYS gets the medical and logistics information are structured data files. The development phase of data sources is at the evaluation stage and testing.

5.2. Data Sources

Several data sources of different types can be incorporated to the WARSYS database. The flexible design of the architecture allows us to design and implement new tables and logic to include any type of sources.

Sample of data sources already included (or can be included) to the WARSYS database are Population data, Chemical plants and other SEVESO infrastructures, Hospital Warehouse Management Systems, Ambulance service, etc. WARSYS architecture is also linked with weather related sources. Currently, weather information as well as up to 5-days forecast is available from the WARSYS system. Weather data are retrieved from the OpenWeather Map service.

5.3. Functional View

For the organization of data within WARSYS database, we derive a functional view that is applied for all the data sources. Every table has a three letter prefix, stating the type of data in the table. Due to the large number of tables, a naming convention, based on their category, has been adopted so as to avoid confusion between tables. This is done by introducing a three letter prefix; the DBA then immediately knows the use of the *Table 1* presents the table prefixes that are currently used. Other prefixes will be created during the development of the project.

Table 1. Table Usage Prefixes

Prefix	Usage of table
GEN	Generic tables
GEO	Tables with Geo-Referenced information or geography oriented
HIS	Hospital Information system outputs
RES	Resources tables
SSIS	SSIS packages tables
SYS	System tables, such as users, user types, priorities etc.
TAX	Taxonomy tables
INC	Incident related tables

5.3.1. Data Protocols

The ongoing goal of the EDXL protocol is to facilitate emergency information sharing and data exchange across the local, state, tribal, national and non-governmental organizations of different professions that provide emergency response and management services. EDXL accomplishes this goal by focusing on the standardization of specific messages (messaging

interfaces) to facilitate emergency communication and coordination particularly when more than one profession or governmental jurisdiction is involved.

CAP: Common Alerting Protocol provides an open, non-proprietary digital message format for all types of alerts and notifications.

HL7-RIM: The Reference Information Model (RIM) is the combined consensus view of information from the perspective of the HL7 working group and affiliates. The RIM is the ultimate source from which all HL7 Version 3 standards draw their information-related content.

EDXL-RM/EDXL-DE: The primary purpose of the Emergency Data Exchange Language Resource Messaging (EDXL-RM) Specification is to provide a set of standard formats for XML emergency response messages. These Resource Messages are specifically designed as payloads of Emergency Data Exchange Language Distribution Element-routed messages (EDXL-DE. Together EDXL-DE and EDXL-RM are intended to expedite all activities associated with resources needed to respond and adapt to emergency incidents. The Distribution Element may be thought of as a container. It provides the information to route payload message sets (such as Alerts or Resource Messages), by including key routing information such as distribution type, geography, incident, and sender/recipient IDs.

EDXL-TEP/TEC: The primary purpose of EDXL-TEP is an XML messaging standard for exchange of emergency patient and tracking information during patient admission or release. TEP supports patient tracking across the Emergency Medical Services (EMS) care continuum, as well as hospital evacuations and patient transfers, providing real-time information to responders, Emergency Management, coordinating organizations and care facilities in the chain of care and transport. The EDXL-TEC Client Registry Exchange standard is the outgrowth of preliminary work of the EDXL Practitioner Steering Group and TEC Steering Committee, which included a Project Initiation Document [EDXL-TEC PID] and Requirements Statement in addition to a Draft Messaging Specification [EDXL-TEC RQMTS] for the Tracking of Emergency Clients Messaging Standard.

EDXL-HAVE: EDXL-HAVE is an XML messaging standard primarily for the exchange of information related to health facilities within the context of emergency management. HAVE supports sharing information about facility services, bed counts, operations, capacities, and resource needs of first responders, emergency managers, coordinating

organizations, hospitals, care facilities, enabling the health community to provide involved parties with a coherent view of the health system.

CSV: comma-separated values, file stores tabular data (numbers and text) in plain-text form where the file is constituted of a sequence of characters, with no binary data. Such files are used for storing records where each record consists of fields, separated by a comma.

XML: Extensible Mark-up language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. It is defined by the W3C's XML 1.0 Specification and by several other related specifications; all of which are open standards.

6. Security

WARSYS incorporates an internal database structure that is used for the storage and retrieval of extracted data and generated information. This information shall be made available to authenticated components/systems, within the IMPRESS architecture, while WARSYS ensures, both internally and externally, security and data protection. Moreover, WARSYS also allows dynamic access to patient information and medical care facility information; the level of detail in retrieved information is dependent on the level of user privilege (gold/silver/bronze).

From the data protection perspective, the following processes will be taking place.

- Servers and Databases are automatically backed up daily
- All databases feature Full Recovery model
- Change Tracking is enabled with a retention period of two days (with an auto cleanup)
- Servers are VMs where snapshots before major changes can be taken, manually by the administrator. Snapshots on Database servers is not a good practice, therefore on the fly backups will be kept before schema changes.
- RAID 6 storage is incorporated.

Beyond the security and data protection that is taken into consideration in the WARSYS component, it includes several backups and system recovery techniques, secure IPs and user credentials for the case of sharing and accessing information within IMPRESS. It also considers the Information Security and Access Protocol. This protocol defines a set of standards, procedures and rules governing the disclosure and use of data.

Finally, to support the continuity of provided services setup, a second (on line-mirrored) infrastructure is provided from different network

premises. Thanks to this feature, all services will be available in the case of lost network. Changes in one system are automatically updated in all other failover premises.

7. Conclusions

The current work provides details of the functional implementation of the WARSYS component in the IMPRESS architecture. The WARSYS is responsible for managing the importing of structured data in IMPRESS and subsequent actions for making available the data to the messaging bus for use by the IMPRESS components.

Within this paper, our development focus is on several key tasks related to the data and characteristics that WARSYS has to consider. In particular, we focused on the evaluation of data sources and have discussed the ways in which data might be imported in WARSYS. Then, we investigated the generation of data sets, available to other IMPRESS components, based on how/when/other parameters are defined during the implementation of the other IMPRESS components. Furthermore, the role of the Messaging Bus has also been discussed as it serves as a communication channel for importing and exporting.

Acknowledgments

This work has been partially funded by the EC in the 7th Framework Programme, (SEC-2013.4.1-4: Development of decision support tools for improving preparedness and response of Health Services involved in emergency situations) under grant number FP7-SEC-2013-608078 - Improving Preparedness and Response of HEalth Services in major crises (IMPRESS).

References

- [1] ALTOVA XML Spy, [available at: <http://www.altova.com/whatsnew.html>].
- [2] CSV Available, [available at: http://en.wikipedia.org/wiki/Comma-separated_values].
- [3] Desourdis, R. I. (2014). *Achieving interoperability in critical IT and communication systems*. Artech House.
- [4] EDXL-HAVE, [available at: <http://docs.oasis-open.org/emergency/edxl-have/v2.0/edxl-have-v2.0.pdf>].
- [5] EDXL-RM/EDXL-DE, [available at: <http://docs.oasis-open.org/emergency/edxl-rm/v1.0/errata/EDXL-RM-v1.0-OS-errata-os.pdf>].

- [6] EDXL-TEC, [available at: <http://docs.oasis-open.org/emergency/edxl-tec-registry/v1.0/edxl-tec-registry-v1.0.pdf>].
- [7] EDXL-TEP, [available at: <http://docs.oasis-open.org/emergency/edxl-tep/v1.0/edxl-tep-v1.0.pdf>].
- [8] HL7-RIM, [available at: <http://docs.oasis-open.org/emergency/edxl-rm/v1.0/EDXL-RM-SPEC-V1.0.pdf>].
- [9] HL7-RIM, [available at: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=77].
- [10] IMPRESS: *IMproving Preparedness and Response of HEalth Services in major crises*. FP7-SEC-2013–608078, [available at: <http://www.fp7-impress.eu>].
- [11] Microsoft SQL Server Migration Assistant for MySQL, [available at: [https://msdn.microsoft.com/en-us/library/hh313109\(v=sql.110\).aspx](https://msdn.microsoft.com/en-us/library/hh313109(v=sql.110).aspx)].
- [12] Python, [available at: <https://www.python.org/>].
- [13] Visual Studio Shell, [available at: <https://msdn.microsoft.com/en-us/library/bb129445.aspx>].
- [14] XML, [available at: <http://en.wikipedia.org/wiki/XML>].