# Evaluation With the VIRTUOSO Platform

## An Open Source Platform for Information Extraction and Retrieval Evaluation.

Gérard Dupont*
CASSIDIAN, Elancourt, France

Gaël de Chalendar†
CEA, Fontenay-aux-Roses, France

Khaled Khelif*
CASSIDIAN, Val de Reuil, France

Dmitri Voitsekhovitch†
CEA, Fontenay-aux-Roses, France

Géraud Canet†
CEA, Fontenay-aux-Roses, France

Stéphan Brunessaux*
CASSIDIAN, Val de Reuil, France

## ABSTRACT

This paper describes a software architecture designed to enable the evaluation of information processing and retrieval systems. The overall objective of our project is to provide an open technical framework for the integration of tools for collection, processing, analysis and communication of open source information[1]. However, enabling the integration of heterogeneous components does not make sense without a proper way to compare the capabilities of multiple tools.

Thus, as part of the architecture the VIRTUOSO platform offers an evaluation framework which allows one to deploy and run evaluation kits for different use-cases.

## Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*; D.2.11 [**Software Architectures**]: Domain-specific architectures; H.4 [**Information Systems Applications**]: Miscellaneous

## General Terms

Design, Standardization

## Keywords

Integration infrastructure, Service Oriented Architecture, Multimedia Information Processing Platform.

*CASSIDIAN, IPCC - name.surname@cassidian.com

†CEA, LIST - name.surname@cea.fr

[1]Open source information is the target of Open Source INTelligence (OSINT) which concerns any information publicly and legally available.


*DESIRE'11,* October 28, 2011, Glasgow, Scotland, UK.


## 1. INTRODUCTION

Today, searching for information on the Internet is nowadays a critical step in an increasing number of tasks. Simply searching for information to assess facts in media monitoring activities or trying to draw a holistic view on potential emerging technologies and organisations in competitive or strategic intelligence, the problem is always how to browse among documents from various providers. In the latter, we will refer to this activity through the generic acronym "OSINT" which stands for Open Source Intelligence.

Thanks to the recent advances in information retrieval, natural language analysis and other research fields, a large panel of tools to enable more effective searching for information exists. However, these software parts are all targeting specific areas within the information retrieval activity and in most cases the task of composing a complete intelligence report will need to make use of a variety of these tools. This leads to a selection process for each of the sub-tasks tackled by these tools in order to have the best tool for each function. The difficulty to use and compare these software tools and their multiple combinations, as well as the lack of information about their actual capabilities, is a real problem that many intelligence organisations are currently facing.

Therefore standardized assessments of information processing and retrieval tools have become a necessity. Taking the benefits from the information retrieval research area and its strong 50-years-old tradition of producing evaluations [2], multiple recognised experimentation protocols as well as metrics and datasets that enable these evaluations are available. Information retrieval is one of the most established evaluation fields, with the well-known TREC conference now in its 19th edition (see [7, 4]).

Recent initiatives are emerging in this area for a more complete and consistent integration of evaluation data and protocols. One can cite the Open Relevant Project[2] launched as a new Apache Lucene sub-project or the PROMISE project[3]. Both try to overcome the problem of dataset (i.e. corpus, queries and relevance judgments) accessibility, PROMISE being more advanced as it provides a real information management system dedicated to this topic, including the man-

[2]http://lucene.apache.org/openrelevance/
[3]http://promise-noe.eu

agement of evaluation results with the DIRECT infrastructure[4].

However the lack of a technical infrastructure to deploy and test tools that cover the full spectrum of an information retrieval system, from crawling to retrieval, is still a constraint. This limitation does not allow a clear understanding of the use and benefit of the multiple tools in the market or beyond-state-of-the-art approaches from laboratories.

In an effort to overcome this problem, the VIRTUOSO platform will provide a technical framework for the integration of tools for collection, processing, analysis and communication of open source information. In other words, it tackles each step of a complete and coherent software infrastructure that address OSINT needs. This framework is thus designed to: enable a smooth combination of multiple tools and provide a common platform that will ease academic evaluation as well as independent technical and functional benchmarking of each tool. In that sense, it is close to the SOIRE framework, proposed in [3], which proposes a technical architecture dedicated to the evaluation of information retrieval systems.

While the primary objective of the VIRTUOSO project is to provide an open source integration platform for OSINT application (see [1]), the problem of evaluating each individual tool that composes a complete system is one of the most predominant problems that the platform should address.

In the following, we will present the VIRTUOSO approach to the problem of integration of heterogeneous software components through a brief description of the software architecture as well as two of its key features: generic service interfaces and standardised processing chain. In the subsequent section, we focus on the implementation of the evaluation framework and the concept of the evaluation kit that enables the assessment of specific functions through recognised datasets and metrics. Finally we will describe the next steps foreseen in the design and implementation of the platform.

# 2. VIRTUOSO APPROACH

The VIRTUOSO framework relies on the WebLab provided by CASSIDIAN as open source software (see [5]). The key concept is "Service Oriented Architecture" (SOA) that serves as the core paradigm for the design and integration of components (i.e. tools). Each component that could be integrated in the platform shall implement one or several functionalities that are described by generic service interfaces. The function work-flow needed to provide user applications will be done by putting together services and calling them in the right order through a standardised processing chain. Each component, implementing one or several service interfaces does not have any knowledge of the other services and their capabilities. They will provide to the others one or several processing capabilities (i.e. services) which will be driven by the orchestrator to define business processes.

Coupled with an "Enterprise Service Bus" (ESB) which allows one to abstract the location and physical implementation of services, this architecture is capable of scaling up for processing large amounts of data. In particular, this abstraction of services to virtual "endpoints" enables dynamic duplication of services (which could be hosted on different physical servers) to address computing bottlenecks in processing chains.

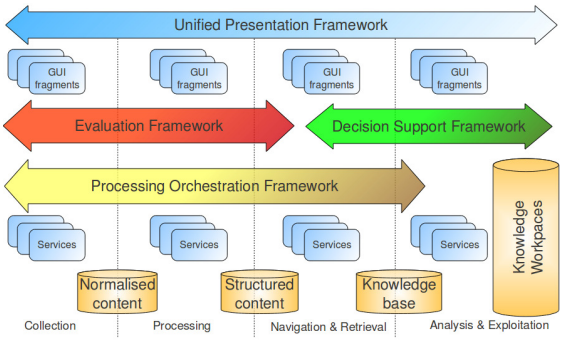
[4] http://direct.dei.unipd.it

As a consequence, the service definition and conception is a key step in the platform. The granularity of the services should be one of the main concerns during the design and development of a component.

## 2.1 Architecture

In the scope of the VIRTUOSO project the platform was segmented in four different and mainly autonomous frameworks in order to isolate the main aspects of a large OSINT system. The various components are thus organised in four frameworks responsible for different functions as presented in figure 1:

**Figure 1: Overview of VIRTUOSO architecture.**




- processing (dedicated to collection & processing) ;
- decision support (which ease semi automatic analysis & exploitation of processing results) ;
- unified presentation (to unify user interface) ;
- and the evaluation (described hereafter).

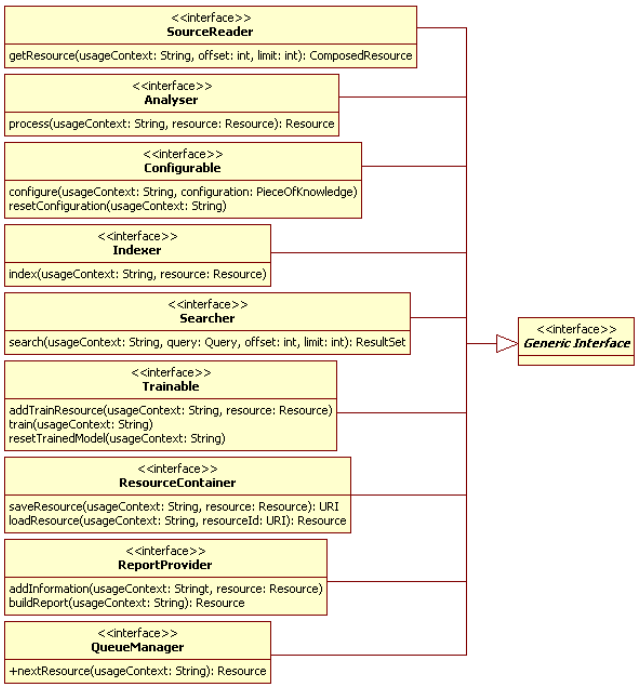
Besides the frameworks, there are repositories which store the raw (i.e. original version) and normalised (i.e. converted to common format) content and the structured content (i.e. analysed version with annotation to locate information extracted). It includes a knowledge base used to store all the information extracted from documents through the processing chain and eventually consolidated by users.

## 2.2 Generic service interfaces

All processing components developed in the context of the VIRTUOSO project will be integrated as services using the WebLab platform which is the core of the processing framework. This relies on many standards, including semantic Web standards and Web service (SOAP) communication protocol. In addition to these standards, WebLab proposes a data exchange model allowing heterogeneous components to communicate with each other. This model defines the common grammar which from a technical point of view will be expressed through XML schema. It describes the structure and parts of content of any data exchanged such as the multiple document types processed in the platform. The semantic Web standards (RDF[6], RDFS/OWL) have been used in order to ensure the sustainability of the model and the compatibility to exploit existing annotation models or domain ontologies built with information extraction tools.

Based on this common data exchange model, the platform also specifies information processing service interfaces. These specifications aim at normalising data exchange and easing component integration. These interfaces are described in UML then transformed into WSDL. The XML-Scheme of the generic interfaces can be included in all services. And thus, the Web Service technology enables one to generate APIs that will enable handling of exchange model objects in the chosen programming languages. A reduced set of generic service interfaces enables the covering of coarse grain functions that appear in an information collection, processing and retrieval application.

**Figure 2: Overview of the WebLab generic service interfaces used in the processing framework.**




Currently in version 1.2, the WebLab core defines 9 generic services interfaces presented in figure 2 and described hereafter:

1. **Analyser:** It contains a method to enable analysis and enrichment of a resource (i.e. document in most of cases), given a usage context.

2. **Configurable:** The configurable interface will be used to define services of which behaviour can be adapted based on usage context.

3. **Indexer:** The indexer interface will be used to define resources indexing services.

4. **Queue Manager:** The QueueManager interface will be used to define services able to iterate over a set of resources and return them one by one. It allows one to model any kind of source of document (i.e. input of the system).

5. **Report Provider:** The report provider interface will be used to define services that can produce reports based on a set of resources consumed. These services will be, in most cases, configurable to enable the reporting to be dependent of the usage context.

6. **Resource Container:** The resource container interface will be used to define services that can manage the resource persistence and thus take into account any kind of storage component.

7. **Searcher:** The searcher interface will be used to define services that allow for the searching of resources.

8. **SourceReader:** The sourceReader interface will be used as entry point for any analysis of source content.

9. **Trainable:** The trainable interface will be used to define services of which behaviour changes dynamically through machine learning. Based on resources consumed, the service will then be able to learn new behaviour models.

Within a complete OSINT system, the interfaces QueueManager, Analyser, Indexer and Searcher will be the most important. These correspond respectively to the connector to the input data, the information analysis and extraction component and finally the search engine.

Thanks to this definition of generic interfaces, the design of a processing chain is simplified. Service orchestration and multiple call management is described in BPEL[5]. In a simple case of document processing, it will start with a QueueManager to get resources, then followed by a sequence of Analyser for information processing and extraction and finally an Indexer. On the retrieval part, chains could involve Searcher as well as Analyser depending on the expected content.

# 3. EVALUATION FRAMEWORK

## 3.1 Overview

The aim of the evaluation framework is to define a standardised business process to evaluate the algorithmic quality of the various tools issued from the processing framework. Evaluations will internally use datasets and evaluation software proposed by international evaluation campaigns, for example, TREC for information retrieval for example. These tools and datasets will be wrapped inside Web services implementing standard WebLab generic service interfaces. This way, the evaluation process can be implemented as a Virtuoso processing chain that can be integrated in a Virtuoso application. Figure 3 shows the envisioned architecture of an evaluation setup using this evaluation framework.

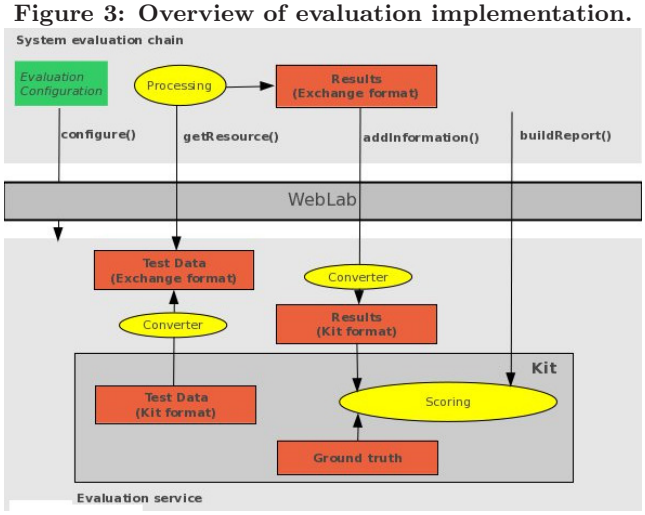
An evaluation service must implement three WebLab generic interfaces:

- Configurable to be able to set global information related to an evaluation, such as the run name;

- SourceReader to enable the service being evaluated to retrieve test data in WebLab data exchange format;

[5] http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html

**Figure 3: Overview of evaluation implementation.**




- ReportProvider to allow the service being evaluated to submit its data in WebLab data exchange format and to get the evaluation result as a WebLab document, results being expressed in XML/RDF.

Thus the creation of a Virtuoso evaluation service consists in writing programmes able to:

- convert the test data format into the WebLab exchange model format;
- convert the WebLab exchange model format to the submission format used in the evaluation campaign;
- express the evaluation result as XML/RDF and generate a result report in the WebLab exchange model.

For each targeted evaluation campaign, it will be necessary to define an ontology enabling the expression of the evaluation data such as scores. We will try to use as much as possible a core evaluation ontology for standard elements such as precision and recall and specialised ontologies for the kind of data specific to a campaign.

The work on the evaluation framework has just started. Currently, the list of evaluation kits necessary to evaluate all the kinds of processing components that will be developed during the VIRTUOSO project is not complete. Some of the kits considered are:

- Named Entities Extraction with ESTER-NE or ACE ;
- Topic Detection, using TDT 5 data;
- Documents Retrieval, using CLEF data;
- Objects Classification using PASCAL VOC data;
- Similar Images Retrieval, using ImageCLEF data;

## 3.2 Information retrieval evaluation kit

Figure 4 shows the interface that any evaluation service has to respect. It implements three generic interfaces. First of all, the Configurable interface allows for the registration of the evaluated service with the evaluation one. Then, the SourceReader interface permits to gather test data documents and finally, ReportProvider is used to submit results (analysis or search for example) and then to ask for the computation of the results.

**Figure 5: Evaluation service kit components.**

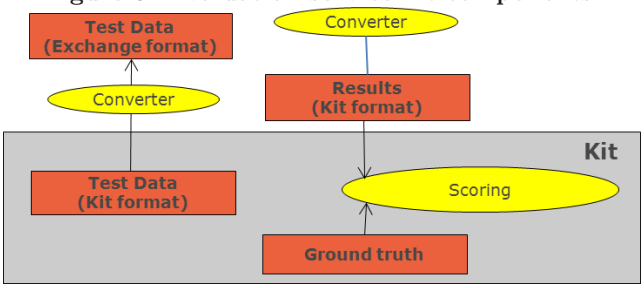
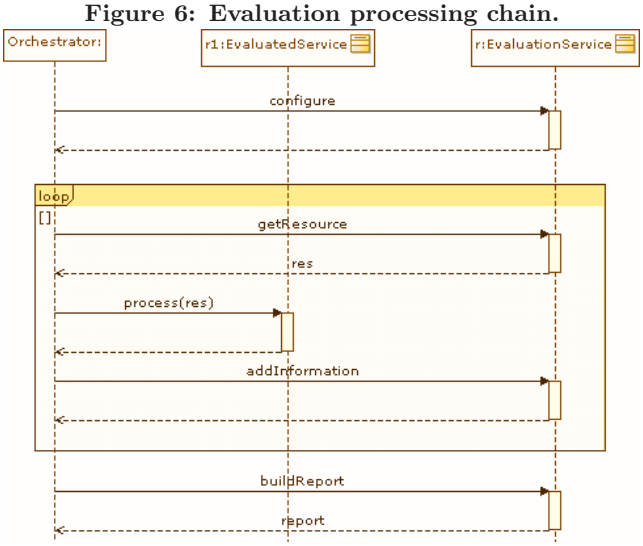


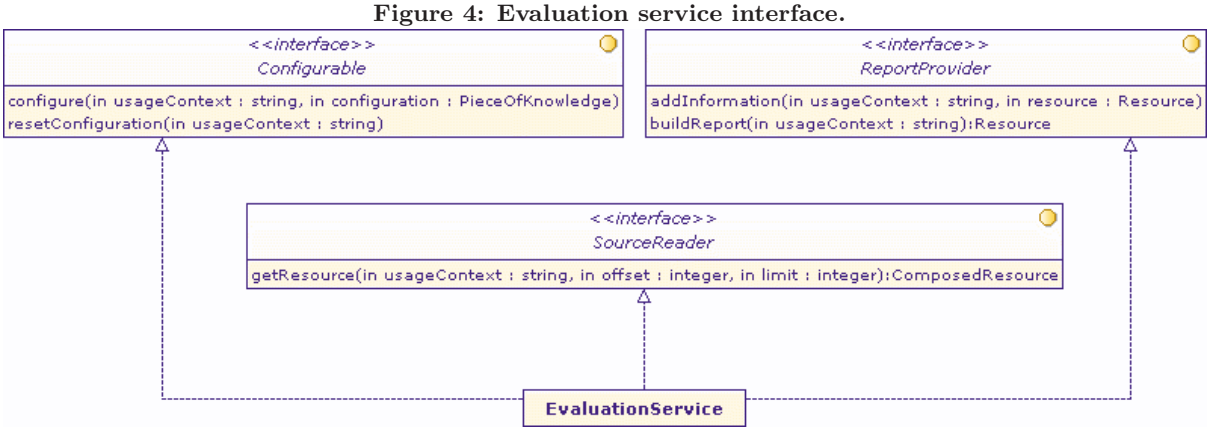

Figure 5 depicts how to build a service respecting the above interface using evaluation campaign material material. An evaluation campaign typically offers a test corpus that is sent to the participants, a *ground truth* that contains the results that participants should find in the test set and a scoring tool allowing one to compute the evaluation measures from results sent by participants and the *ground truth*. To implement the evaluation one will need two converter programs. The first one is to convert the test data in the kit format into WebLab documents. These documents will then be retrieved using the getResource() method from the SourceReader interface. The second converter must be able to convert the WebLab documents transmitted to the evaluation service which contain the evaluated service results on the test data into results in the kit format. Finally, the implementation of the buildReport() method will call the kit scoring tool and express its output into RDF data integrated into the WebLab Resource which is its return value.

**Figure 6: Evaluation processing chain.**




The evaluation service and the evaluated one will be glued together using a processing chain containing a sequence like

Figure 4: Evaluation service interface.




the one depicted in Figure 6. The processing chain contains the following steps. First, the orchestrator configures the evaluation service with a given usage context identifying the service instance to be evaluated. Then, in a loop, each document to be evaluated is retrieved and sent to the evaluated service. The evaluated service result is sent back to the evaluation service. When all the documents are analysed, the orchestrator asks the evaluation service to build a report based on the submitted analysed documents. This one sends back the evaluation results.

## 4. CONCLUSION & NEXT STEPS

The evaluation process presented above is adapted to the standard comparative evaluation campaigns paradigm. This one does not take into account the fact that, in the information retrieval user experience, the way to present results to the user and to interact with them is as important as the pure quality of the algorithms. In the remaining timz of the project, we will study how this framework can evolve to take into account the user interaction through graphical user interfaces. For this sake the next version of the VIRTUOSO architecture will include detailed specifications of user interaction and its use in interactive information retrieval experimentations.

## 5. ACKNOWLEDGMENTS

The work described in this paper has been partially funded by the European Union in the framework of the FP7 Security Virtuoso project[6] (contract FP7-SEC-2009-1 242352).

## 6. REFERENCES

[1] G. Canet, G. de Chalendar, L. Dubost, G. Dupont, A. Dyevre, and K. Khelif. Toward a versatile information toolkit for end-users oriented open-sources exploitation : Virtuoso. In *"Sources Ouvertes et Services" - SOS'2011, En association avec EGC'2011*, 2011.

[2] C. W. Cleverdon. Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems. Technical report, Cranfield Coll. of Aeronautics, Cranfield, England, 1962.

[3] M. Dittenbach, B. Pflugfelder, A. Pesenhofer, G. Roda, and H. Berger. Soire: a service-oriented ir evaluation architecture. In *Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 2101–2102, New York, NY, USA, 2009. ACM.

[4] S. T. Dumais and N. Belkin. *TREC: Experiment and Evaluation in Information Retrieval*, chapter The Interactive TREC Track: Putting the user into search., pages 123–153. E. Voorhees and D. Harman (Eds.), MIT Press, 2005.

[5] P. Giroux, S. Brunessaux, S. Brunessaux, J. Doucy, G. Dupont, B. Grilheres, Y. Mombrun, and A. Saval. Weblab: An integration infrastructure to ease the development of multimedia processing applications. *ICSSEA*, 2008.

[6] R. W. Group. Resource description framework (rdf): Concepts and abstract syntax. Technical report, W3C, 2004.

[7] P. Thompson, H. Turtle, B. Yang, and J. Flood. Trec-3 ad-hoc retrieval and routing experiments using the win system. In Gaithersburg, editor, *Proc. of the Text REtrieval Conference*, pages 211–218, USA, 1995.

[6] http://www.virtuoso.eu/