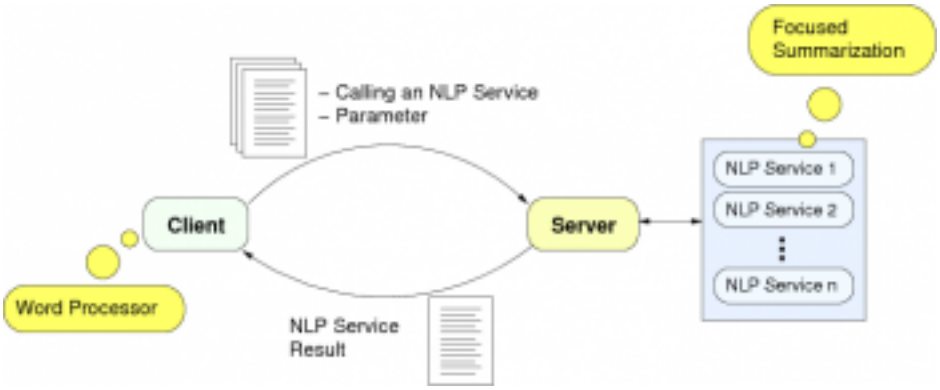
Semantic Assistants Architecture

- Semantic Assistants
- Semantic Desktop

toc_collapse=0; Table of Contents

1. Overview

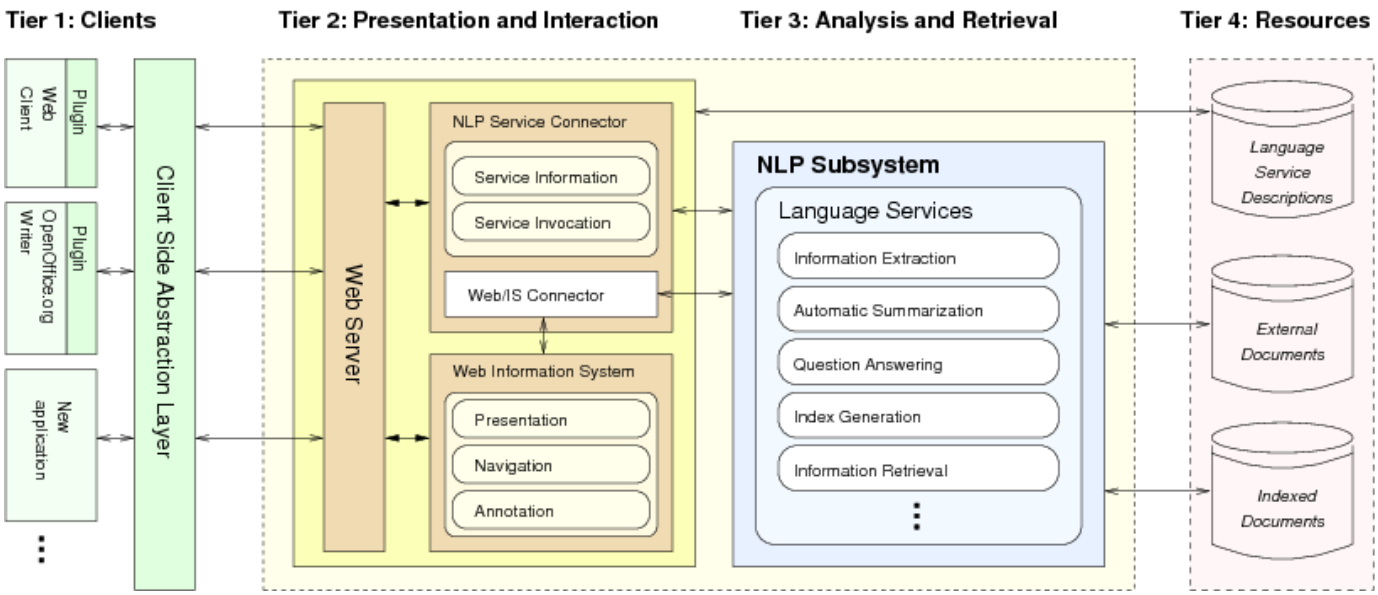


The Semantic Assistants project aims to bring natural language processing (NLP) techniques directly to end users by integrating them with common desktop applications (word processors, email clients, Web browsers, ...), web information systems (wikis, portals) and mobile applications (based on Android). To facilitate this integration, a service-oriented architecture has been developed that allows to integrate clients with NLP services implemented in the GATE framework. NLP services are described with an ontology-based (OWL) semantic description that captures users, their languages, tasks, and various artifacts.

For the general motivation, design, and background on the Semantic Assistants project, please read our ASWC 2008 publication [1]. If you use our Semantic Assistants architecture, resources, or client-plug ins for your own work, please include a reference to this paper.

2. Architecture: Web Services for NLP

Semantic Assistants use a four-tier information system architecture:



Semantic Assistants Architecture

Tier 1
of the architecture consists of client applications and a *Client-Side Abstraction Layer (CSAL)*. Currently, there are a number of example clients distributed with the system: a simple command-line client for testing purposes, a plug-in for the OpenOffice.org/LibreOffice *Writer* word processor and a plug-in for the Eclipse framework. In addition to desktop clients, we offer two novel integrations for web information systems, in particular for wikis and web portals. The client-side abstraction layer consists partly of hand-written Java classes that provide common client-side functionality, partly of automatically generated Java classes. The communication between client and server is implemented by means of W3C Web services.

Tier 2
of the architecture consists of a *Web server* and the *NLP Service Connector*. The NLP Service Connector currently integrates the GATE framework for NLP. It is responsible for a number of tasks, including communication with the client, reading and querying the language service descriptions, running requested language services, and generating response messages.

Tier 3
is the NLP subsystem. At present, only the GATE framework is supported. It makes use of the GATE API in order to assemble language services, store them in a permanent way, and invoke them when they are requested by a client.

Tier 4
is the resource tier. Here we have the language service descriptions, which are authored in the Web Ontology Language (OWL). Tier 4 further contains external documents, which the NLP subsystem must be able to access.

3. System Components

The current implementation of the Semantic Assistants architecture comes with the following open source components:

Server
The server is the core of the architecture. It communicates with the clients through the CSAL on one hand and to the NLP framework through the NLP Service Connector on the other. At present, the architecture only contains a connector for GATE. However, it was explicitly designed to allow an easy integration of other frameworks (for example, UIMA). For describing available services, we use ontology-based (OWL) service descriptions. As a service-oriented architecture (SOA), every service is automatically available to all clients connected to the architecture, using standard Web Services Description Language (WSDL) interface descriptions.

Client-Side Abstraction Layer (CSAL)
Our top goal was to make it as easy as possible for client (plug-in) developers to integrate NLP functionality. As clients

should be able to connect to the architecture entirely by "local" means, we provide an *abstraction layer*, named CSAL, which is located on the client side and performs the actual communication with the server.
Apart from the communication functionality, the CSAL also provides common client-side functionality, i.e., useful data types and methods that are frequently required when integrating NLP into desktop clients.

Clients

A number of clients come with the architecture: For the desktop, a command-line client, a plug-in for the OpenOffice.org/LibreOffice *Writer* word processor [2], and a plug-in for the Eclipse framework [3]. For information systems, we provide an integration with wiki systems, in particular MediaWiki [6]. An integration with the Liferay web portal framework is also available [4]. Our latest addition is an Android library [5] that allows developers of mobile apps to easily integrate NLP services to offer 'intelligent' support to end users.

Example Resources

NLP functionality is provided to clients through Web services. To match clients with suitable services (depending on language, formats, etc.), each NLP service comes with a semantic service description in OWL format. Three example service descriptions are included in the current distribution: an information extraction (IE) service that detects persons and locations (using GATE's ANNIE pipeline), an IR service (using the Yahoo PR) and a compound service, which combines the IR and the IE service. These should help you in defining your own NLP services that you deliver to end users (e.g., summarization, question-answering, domain-specific NLP services).

4. Software and Documentation

Please visit the Semantic Assistants page for more documentation and download of our open source software.

References

1. Witte, R., and T. Gitzinger, "Semantic Assistants – User-Centric Natural Language Processing Services for Desktop Clients", *3rd Asian Semantic Web Conference (ASWC 2008)*, vol. 5367, Bangkok, Thailand : Springer, pp. 360–374, Feb. 2–5, 2009, 2008.
2. Gitzinger, T., and R. Witte, "Enhancing the OpenOffice.org Word Processor with Natural Language Processing Capabilities", *Natural Language Processing resources, algorithms and tools for authoring aids*, Marrakech, Morocco, June 1, 2008.
3. Butz, C., and P. Lingras (Eds.), Witte, R., B. Sateli, N. Khamis, and J. Rilling, "Intelligent Software Development Environments: Integrating Natural Language Processing with the Eclipse Platform", *24th Canadian Conference on Artificial Intelligence (Canadian AI 2011)*, vol. 6657, St. John's, Newfoundland and Labrador, Canada : Springer-Verlag, pp. 408–419, May 25–27, 2011.
4. Löffler, F., B. Sateli, B. König-Ries, and R. Witte, "Semantic Content Processing in Web Portals", *4th Canadian Semantic Web Symposium (CSWS 2013)*, vol. 1054, Montréal, QC, Canada : CEUR-WS.org, pp. 50–51, 07/2013.
5. Daniel, F., G. A. Papadopoulos, and P. Thiran (Eds.), Sateli, B., G. Cook, and R. Witte, "Smarter Mobile Apps through Integrated Natural Language Processing Services", *The 10th International Conference on Mobile Web Information Systems (MobiWIS 2013)*, vol. 8093, Paphos, Cyprus : Springer Berlin Heidelberg, pp. 187–202, 08/2013.
6. Sateli, B., and R. Witte, "Natural Language Processing for MediaWiki: The Semantic Assistants Approach", *The 8th International Symposium on Wikis and Open Collaboration (WikiSym 2012)*, Linz, Austria : ACM, 08/2012.

Except where otherwise noted, all original content on this site is copyright by its author and licensed under a Creative Commons Attribution-Share Alike 2.5 Canada License.

Source URL (retrieved on *2025-12-04 05:11*): https://www.semanticsoftware.info/semantic-assistants-architecture