

## INTELLECTUALIZATION OF PROJECT MANAGEMENT WEB SERVICES BASED ON INTEGRATION WITH NATURAL LANGUAGE PROCESSING MODULES

*Yegoshyna G.A., Voronoy S.M.*

*O. S. Popov Odesa national academy of telecommunications,  
1 Kuznechna St., Odessa, 65029, Ukraine.  
yegoshyna@onat.edu.ua*

## ІНТЕЛЕКТУАЛІЗАЦІЯ ВЕБ-СЕРВІСІВ УПРАВЛІННЯ ПРОЕКТАМИ НА ОСНОВІ ІНТЕГРАЦІЇ З МОДУЛЯМИ ПРИРОДНО-МОВНОЇ ОБРОБКИ

*Єгошина Г.А., Вороной С.М.*

*Одеська національна академія зв'язку ім. О.С. Попова,  
65029, Україна, м. Одеса, вул. Кузнечная, 1.  
yegoshyna@onat.edu.ua*

## ИНТЕЛЛЕКТУАЛИЗАЦИЯ ВЕБ-СЕРВИСОВ УПРАВЛЕНИЯ ПРОЕКТАМИ НА ОСНОВЕ ИНТЕГРАЦИИ С МОДУЛЯМИ ЕСТЕСТВЕННО-ЯЗЫКОВОЙ ОБРАБОТКИ

*Егошина А.А., Вороной С.М.*

*Одесская национальная академия связи им. А.С. Попова,  
65029, Украина, г. Одесса, ул. Кузнечная, 1.  
yegoshyna@onat.edu.ua*

**Abstract.** The article explores a possibility of project management web services intellectualization and perfection with the use of natural language processing techniques. It is shown, that the support of high reliability, extensibility, flexibility and density of the interface of such systems is a necessary condition for the providing of effective tools for developers of software systems of various profiles. The directions of development of modern means and services of natural language processing are analyzed. These include libraries and toolkits such as Google Cloud Natural Language API, Amazon Comprehend, IBM Watson's, CVLizer, Wit.ai, DialogFlow, TextRazor, InData Labs API, Paralleldots API, NLP фреймворк BitText, Ambiverse API, Datumbox API and Indico.io. The requirements for the project management system are formulated in terms of reliability, extensibility, flexibility, interface density and availability. The article shows the advantages of using natural language processing tools to improve the efficiency and availability of project management web services. Since most of natural language processing public web services provide similar functionality with similar performance, it is reasonable to create a multiservice integral system architecture that would be able to access several such services at concurrently (based on load or quality criteria), while falling back to the locally deployed natural language processing system when it comes to processing secure or confidential data or in case of external services failures.

**Key words:** web service, project management, Integrated Development Environment, module, interface, natural language processing, parser, lemmatization, machine learning.

**Анотація.** У статті розглянуто можливість інтелектуалізації веб-сервісів управління проектами шляхом інтеграції з модулями обробки природної мови. Показано, що підтримання високої надійності, розширюваності, гнучкості та повноти інтерфейсу таких систем є необхідною умовою надання ефективного інструментарію для розробників програмних систем різного профілю. Встановлено, що поширеною тенденцією є підвищення портативності програмного забезпечення якісного показника, що відповідає за можливість використання одного і того ж додатку в різних середовищах виконання програмного забезпечення. Аналізуються напрями розвитку сучасних засобів та сервісів обробки природної мови. Розглянуто такі бібліотеки й інструментарії, як Google Cloud Natural Language API, Amazon Comprehend, IBM Watson's Natural Language Understanding, CVLizer, Wit.ai, DialogFlow, TextRazor, InData Labs API, Paralleldots API, LUIS language understanding, NLP фреймворк BitText, Ambiverse API, Datumbox API і Indico.io. Сформульовано вимоги до системи управління проектами з

надійності, розширюваності, гнучкості, повноти інтерфейсу і доступності. Сформовані переваги використання засобів обробки природної мови для підвищення ефективності та доступності веб-сервісів керування проектами. Показано доцільність розробки мультисервісної системи управління програмними проектами, що забезпечує динамічну генерацію користувальницького інтерфейсу з інтегрованими модулями, які реалізують функціональність розглянутих веб-сервісів обробки природної мови.

**Ключові слова:** веб-сервіс, управління проектами, середовище виконання, модуль, інтерфейс, обробка природної мови, парсер, лематизація, машинне навчання.

**Аннотація.** В статье рассмотрена возможность интеллектуализации веб-сервисов управления проектами путем интеграции с модулями обработки естественного языка. Показано, что поддержание высокой надежности, расширяемости, гибкости и полноты интерфейса таких систем является необходимым условием предоставления эффективного инструментария для разработчиков программных систем различного профиля. Установлено, что распространенной тенденцией является повышение портативности программного обеспечения с целью обеспечения возможности использования одного и того же приложения в различных средах исполнения. Анализируются направления развития современных средств и сервисов обработки естественного языка. Рассмотрены такие библиотеки и инструментальные средства, как Google Cloud Natural Language API, Amazon Comprehend, IBM Watson's, CVLizer, Wit.ai, DialogFlow, TextRazor, InData Labs API, Paralleldots API, NLP фреймворк BitText, Ambiverse API, Datumbox API и Indico.io. Сформулированы требования к системе управления проектами по надежности, расширяемости, гибкости, полноте интерфейса и доступности. Выделены преимущества использования средств естественно-языковой обработки для повышения эффективности и доступности веб-сервисов управления проектами. Показана целесообразность разработки мультисервисной системы управления программными проектами, обеспечивающей динамическую генерацию пользовательского интерфейса с интегрированными модулями, реализующими функциональность рассмотренных веб-сервисов обработки естественного языка.

**Ключевые слова:** веб-сервис, управление проектами, среда исполнения, модуль, интерфейс, обработка естественного языка, парсер, лемматизация, машинное обучение.

The ability to make and implement management decisions in reaction to the external and internal environment changes of the enterprise is, without a doubt, one of the crucial success factors in the modern business world. For several decades computers, informational networks and specialized software, designed for different purposes, whether it be the enterprise resource management, client relations organizing, planning or internal communications support have been used for such tasks.

One of the main software packages among such products are project management systems - specialized software designed to organize processes of planning, distribution, control and implementation of project tasks either in scope of one workflow or several projects. Maintenance of high reliability, expandability, flexibility, interface comprehension and continuous improvement of such systems is a necessary step for provision of useful and efficient support toolset for employees [1–3].

This article covers the problem of project management solutions development and intellectualization due to the implementation of natural language processing capabilities aimed for the use of small and micro enterprises, development teams and educational groups in conditions of limited resources. The proposed solution must be deployable to low-end shared hosting services with low implementation costs and limited processing power and improve an existing project management web services without sacrificing any of their features.

**The goal of the article** is to investigate a possibility of project management web services intellectualization and perfection with the use of natural language processing techniques.

Natural human language is a discrete symbolic categorical lossy signaling system, used for information encoding for communication between human beings. It uses sound, gestures, images and symbols. Unlike machine codes and strict programming languages (formal languages), natural human language is highly ambiguous, dependent on common sense, contextual knowledge and user life experience. Straight parsing technologies used to analyze highly formalized data, like programs written in programming languages or other structured data, are ineffective for natural language processing. This makes the task of conversion of natural language data into structured information

available for use by computers, and vice versa, quite hard and requires exhaustive knowledge from the processing software about various language aspects. In most cases: syntax (word structural relationships), morphology (word components and their composition), lexical and compositional semantics (study of meaning), but sometimes: phonetics and phonology, discourse conventions and pragmatics. Today, natural language processing is implemented in a wide variety of “everyday use” software: search engine query processors and indexers, office suit grammar check systems, virtual keyboards, question-answering bots and personal assistants [1–3].

A review of the most popular and available libraries, services and datasets that can be used for the development and intellectualization of project management web application is needed. Natural language processing web services are mostly software-as-service solutions that may lose opportunities for specialization (it is quite hard to perform specific features on third party services that might not be supported by service provider by default), costs and security (you must ensure the security of transaction of the confidential natural language data through the Internet and trust service provider for high information security provision). But it surely gains in the absence of needing to build server architectures for language processing tasks aside of your main application and its maintenance. Also, despite most of service providers supporting application training with your datasets, all the basic functionality of the system, algorithms it uses and their efficiency is carried out and upgraded by the service provider, allowing the developer to focus on the actual language processing based application functionality. Google Cloud Natural Language API suite provides syntax analysis, NER and entities extraction, sentiment analysis, and content tagging in atleast9 languages include English and Chinese (Simplified and Traditional). Amazon Comprehend is NLP and ML suite from Amazon, covering most common tasks like NER, tagging, and sentiment analysis. IBM Watson's provides public services of Natural Language Understanding, Natural Language Classifier and Machine Translation.

CVLizer is a multilingual resume parser that supports merging of information sources to not just by resume itself, but accompanying sources and other documents. “Wit.ai” and “DialogFlow” (ex-“API.ai” by Google) are web services that provide extensive frameworks for creating natural language interfaces for applications and devices. TextRazor offers commercial API for entity extraction, disambiguation and linking, key phrase extraction, automatic topic tagging and classification. InData Labs API provides machine learning API to analyze text data from social media and unlock rich audience insights, including sentiment analysis, language and location detection, interests analysis, gender and age identification based on text data from user profiles. Paralleldots API provides multilingual sentiment analysis, keyword extraction, text classification on taxonomies, entity extraction, sentiment and intent analysis as well as provides own implementation of abusive content classifier. BitText API is an NLP Framework supporting lemmatization, POS-tagging, phrase extraction and entities detection, sentiment analysis and provides a machine learning middleware for conversational agents (slot generation and sentence rewriting functions). Ambiverse API is specialized on entities extraction from incoming natural language text and matches them to the articles from public knowledge bases. Datumbox API supports functions of sentiment analysis, twitter sentiment analysis, subjectivity analysis, topic classification, spam detection, adult content detection, readability assessment, language detection, commercial detection, educational detection, gender detection, keyword extraction, text extraction and document similarity. Indico.io API provides functions of sentiment analysis, text tagging, keywords extraction, language identification, entities recognition, sentiment, emotion analysis, unusual author personality detection and article political analysis [2–8].

Unfortunately, unlike most of general-purpose natural language processing web services, specialized services for business purposes (including such features as document/resume parsing, employee profile assemblers, etc.) are all priced by “contract price” and cannot be evaluated by non-commercial, non-legal entities. That is a factor that may interfere with the choice of such product for the developed project management system integration in order to avoid unnecessary risks.

Most of project management software solutions on the market mostly provide the same set of functionalities but vary in the pricing strategy, limitations of functionality for different categories

of customers, flexibility in customization options, expandability in the scope of different customers (developers and non-developers), interface comprehensibility, availability for desktop and mobile users using different operation systems and some distinguishing features.

In [3-7] we have reviewed most common SaaS and deployable project management web applications and services, including: Bitrix24, Wrike, Zoho Projects, Easy Redmine, Freedcamp, Taiga, Quire, Microsoft Project, Redmine, Collabtive, Feng Office. For now, language processing capabilities got partially implemented in Bitrix24 (conversation agent, automatic hyperlinking), Microsoft Project (office suit grammar correction). The previously not-described team messaging system Slack has got a task assistant named “Talla” which is still a beta version platform expanded to Microsoft teams internal communication system. In its current version, the bot just serves as a natural language interface to SaaS-based corporate knowledge base. However, the commercial use of natural language processing software being a trend of recent years, the process of interpenetration with business applications has just began. Natural language processing systems are either deeply spread in the fields they have been used for a long time (big data processing, document parsing, etc), or not yet fully integrated into the everyday common use spheres (including most business applications).

Most common concepts of natural language services application in the scope of business software are:

- automatic referring and hyperlinking scripts: automatic highlighting of project- or company-related named entities and noun groups that are automatically linked to the correspondent entity representation page (user names to user profiles, department names to department information boards, dates to calendar views and automatic task creation forms, etc.);

- language processing interfaces: question-answering systems and conversational agents for internal and external use (from conversation-based poll forms and step-by-step document execution interfaces to multitask informational systems capable of managing the whole web service functionality along with serving as first line of technical support and customer service);

- trend identification and sentiment& emotion analysis systems – first used mostly for marketing analysis, while second might be useful not just as review evaluation tool, but internal and external communication data gathering tool (for example, it can be used to analyze quality of custom support service or detect a ripening conflict within the collective);

- natural language processing-based information extraction support tools – resume parsers, inner correspondence document classifiers, customized information search indexers, etc.;

- natural language-based spam filtration for elimination of malicious correspondence traffic. Depending on the current task, the depth of the NLP system is defined as:

- for simple text parsers, named entities extractors and search applications only first representation segment (words/lemmas) is required.
- for more complex spell checking, machine translation systems, second representation segment (sentences) is required.
- finally, most complex information extraction applications, conversation agents, advanced document summarization systems the last, context-aware level is used. In simplified form it can be explained as a sequence of processing levels: raw text – words parsing / lemmas extraction – sentence analysis – sentence groups relations and text meaning analysis.

We are focusing on a small team project management systems aimed to be used by software project development teams, education groups and employees of small and micro enterprises. Such teams would be limited in funding and hardware resources (possibly a small, full-grade web site on limited-plan virtual private servers, or basic invocation-based website built with one of the popular content management systems on shared hosting). That means, we have only available low processing power capabilities and limited RAM (usually up to 1-2Gb). They might or might not use the developed project management web service, or any other deployable or cheap software as service distributed solution. And more likely (with the exception of the development team), in such teams there are no professional developers interested in maintaining high complexity systems, so

the developed service has to be easily configurable and deployable.

The most generic configuration of shared hosting is built on Apache HTTP Server 2.4 and Nginx 1.12-1.13 HTTP static cache server, PHP Hypertext Preprocessor versions 5.6 – 7.1 and MySQL v5.7 relational database management systems.

Currently there are no natural language integrated solutions within such configuration at all. The top-most issue in it is that despite actual implementations of natural language processing algorithms in PHP, most of them either have very basic tasks support (stemming, tokenization, clustering and very basic classification), either are wrappers around Java or Python libraries which can only work if the whole project is deployed to a virtual or dedicated server. On the other side, most of modern approaches on machine learning involve deep learning or use of long short-term memory neural networks, which require Java or Python support. The fallback to classical non-ML based method might cause the whole system failures because script memory limit violation errors (non-compressed in progress ruleset based on a sufficiently large corporation might take up to 3-6Gb RAM).

The classic application programming interface-centric web application architecture is given below (see Fig. 1). The ultimate advantages of such approach are the ability to redistribute server load due to separation of heavy load page calls, lightweight API requests and the possibility to separate web application view layer development and maintenance from its business logic and data storage (so called “backend”). In order to implement the solution proposed it might be extended (see Fig. 2). As we can see, now it relies on the API stack instead of a single client-server API, which is required to establish communication between all the different components of the system. Communication between 3rd party natural language processing services can be performed either between web server NLP controller and services or directly between services and client applications.

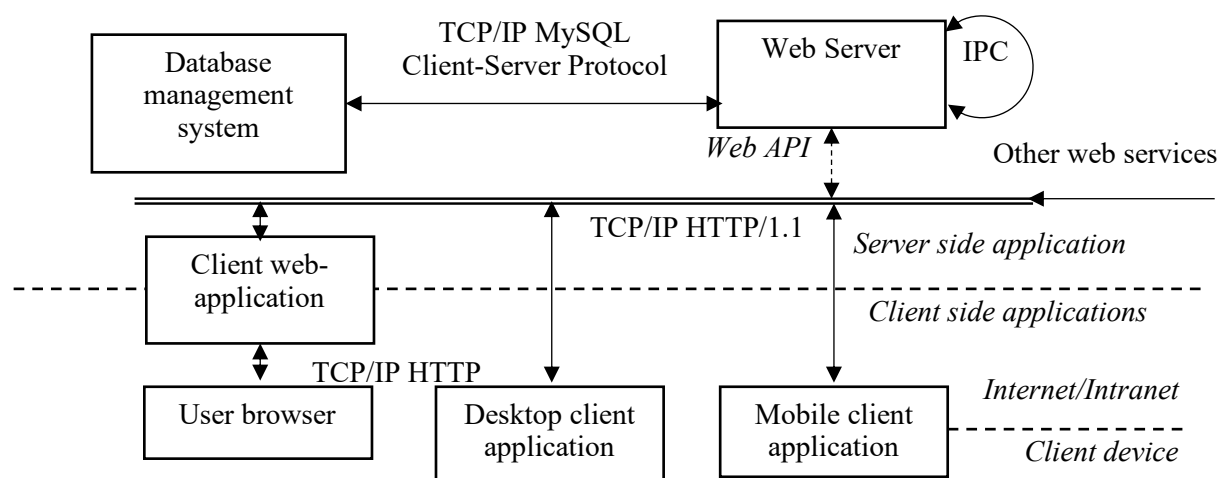


Figure 1 – Classic “API-first” web-application architecture simplified scheme

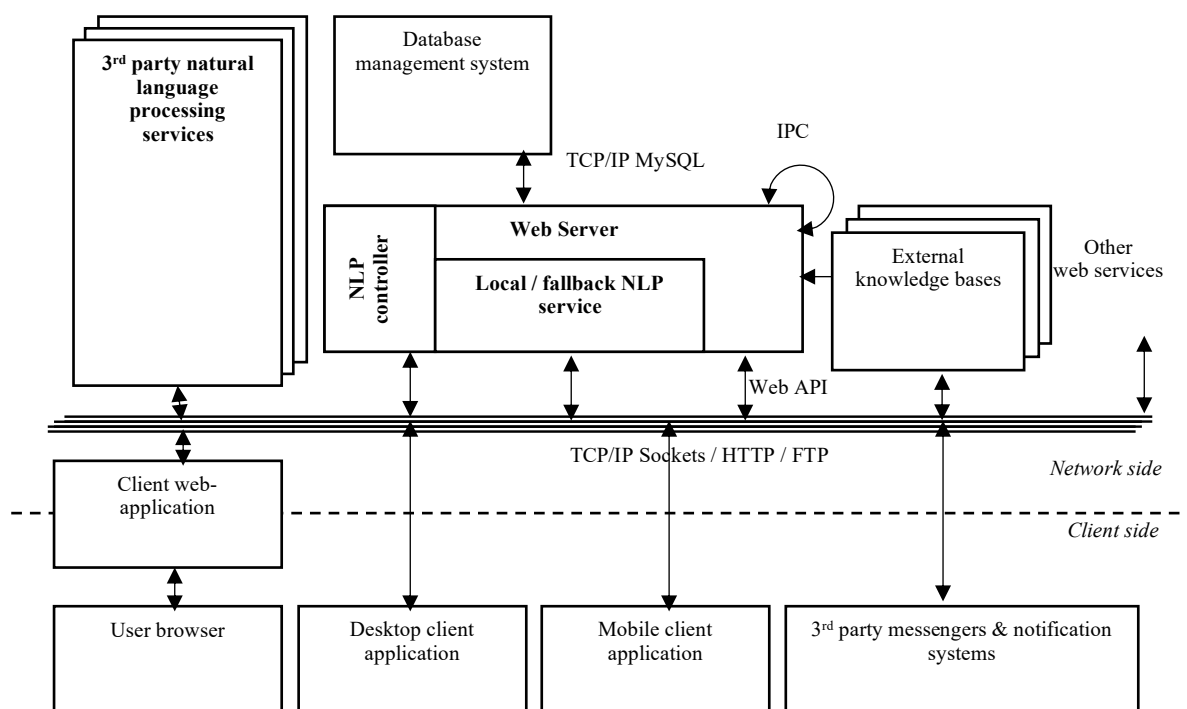


Figure 2 – Extended architecture with reliable NLP subservices support

In this way, all “heavy” computations and file transfers can be separated from the main application, thus reducing the load since NLP services and web server NLP controller would always communicate with text data even if optical character recognition or speech processing is performed. Tasks that cannot be performed via external services will be picked up either by local/fallback NLP service or web server core application itself. When the external service fails to process a request, it will be either bounced to NLP controller or detected as a “no response” event by the web server core application, so that it can be either redirected to other external or internal NLP service or fed to core application for non-NLP based processing. Since even conversation-based natural language processing services do not recognize non-NLP payload (such as files attached to the message), additional message payload detection should also be supervised by the core application NLP or core controllers. In such cases, the web server application acts as a long-term memory and context-aware conversation, correcting the whole interaction process as necessary.

Despite the fact the natural language processing field is not new and has origins back to the time of the birth of computer systems, most widespread public available solutions of it, as well as advancements, are dated within last decade. It is a rapidly developing industry that manifests itself not just in specialized analytic software, but everyday things and programs: search engines, business software, mobile apps and social networks. And yet, it is still has not fully penetrated all areas where its advantages can be used. One of such areas is project management web services. The tendency to a minimal type of system configuration lightweight project management system cannot be improved solely with just a set of natural language processing extensions without sacrificing reliability, expandability, flexibility, interface comprehensibility, availability or price with no principal changes in either application architecture or environment configuration and modifications that might satisfy all such criteria needed.

REFERENCES:

1. Saif Addin Ellafi, 'Comparing production-grade NLP libraries: Accuracy, performance, and scalability' <<https://www.oreilly.com/ideas/comparing-production-grade-nlp-libraries-accuracy-performance-and-scalability>> accessed Oct 23 2018.
2. Shanglun Wang, 'How to Build a Natural Language Processing App' <<https://www.toptal.com/algorithms/how-to-build-a-natural-language-processing-app>> accessed Oct 23 2018.
3. Asma Adala, Nabil Tabbane, Sami Tabbane, 'A Framework for Automatic Web Service Discovery Based on Semantics and NLP Techniques' <<https://www.hindawi.com/journals/am/2011/238683/>> accessed Oct 27 2018.
4. Huang, L. and Sagae, 'Dynamic programming for linear-time incremental parsing' <<http://www.aclweb.org/anthology/P10-1110>> accessed Oct 23 2018.
5. Lapata, M. and Keller, F. 'The web as a baseline: Evaluating the performance of unsupervised webbased models for a range of NLP tasks' <<http://www.aclweb.org/anthology/N04-1016>> accessed Oct 23 2018.
6. David Talby, 'Introducing the Natural Language Processing Library for Apache Spark' <<https://databricks.com/2017/10/19/introducing-natural-language-processing-library-apache-spark.html>> accessed Oct 20 2018.

ЛІТЕРАТУРА:

1. Saif Addin Ellafi. Comparing production-grade NLP libraries: Accuracy, performance, and scalability: веб-сайт. URL: <https://www.oreilly.com/ideas/comparing-production-grade-nlp-libraries-accuracy-performance-and-scalability> (дата звернення 23.10.2018).
2. Shanglun Wang. How to Build a Natural Language Processing App: веб-сайт. URL: <https://www.toptal.com/algorithms/how-to-build-a-natural-language-processing-app> (дата звернення 23.10.2018).
3. Asma Adala, Nabil Tabbane, Sami Tabbane. A Framework for Automatic Web Service Discovery Based on Semantics and NLP Techniques: веб-сайт. URL: <https://www.hindawi.com/journals/am/2011/238683/> (дата звернення 27.10.2018).
4. Huang, L. and Sagae, K. Dynamic programming for linear-time incremental parsing: веб-сайт. URL: <http://www.aclweb.org/anthology/P10-1110> (дата звернення Oct 23.10.2018)
5. Lapata, M. and Keller, F. The web as a baseline: Evaluating the performance of unsupervised webbased models for a range of NLP tasks: веб-сайт. URL: <http://www.aclweb.org/anthology/N04-1016> (дата звернення 23.10.2018)
6. David Talby. Introducing the Natural Language Processing Library for Apache Spark: веб-сайт. URL: <https://databricks.com/2017/10/19/introducing-natural-language-processing-library-apache-spark.html> (дата звернення 20.10.2018).

DOI 10.33243/2518-7139-2018-1-2-94-100