Brazilian Journal of Biological Sciences, 2018, v. 5, No. 9, p. 3-12.
ISSN 2358-2731
https://doi.org/10.21472/bjbs.050901

H-KaaS: A Knowledge-as-a-Service architecture for E-health

Renan G. Barreto[1], Lucas Aversari[1], Cecília Neta A. P. Gomes[2] and Natasha C. Q. Lino[1]

[1]Universidade Federal da Paraíba. Centro de Ciências Exatas e da Natureza. Programa de Pós-Graduação em Informática. *Campus* I. João Pessoa-PB. Brazil. (CEP 58051-900).
[2]Programa de Pós-Graduação em Modelos de Decisão e Saúde. Centro de Ciências Exatas e da Natureza. Universidade Federal da Paraíba. *Campus* I. João Pessoa-PB. Brazil. (CEP 58051-900).

Abstract. Due to the need to improve access to knowledge and the establishment of means for sharing and organizing data in the health area, this research proposes an architecture based on the paradigm of Knowledge-as-a-Service (KaaS). This can be used in the medical field and can offer centralized access to ontologies and other means of knowledge representation. In this paper, a detailed description of each part of the architecture and its implementation was made, highlighting its main features and interfaces. In addition, a communication protocol was specified and used between the knowledge consumer and the knowledge service provider. Thus, the development of this research contributed to the creation of a new architecture, called H-KaaS, which established itself as a platform capable of managing multiple data sources and knowledge models, centralizing access through an easily adaptable API.

Keywords: Knowledge-as-a-service architecture; Health informatics; Knowledge representation.

Received
January 8, 2018

Accepted
April 30, 2018

Released
April 30, 2018

Full Text Article

ORCID

0000-0002-0919-6957
Renan G. Barreto

0000-0003-4152-5842
Lucas Aversari

0000-0002-8131-0566
Cecília Neta A. P. Gomes

0000-0003-2395-4846
Natasha C. Q. Lino

Introduction

With the advance of processing power and speed of data collection on the internet, many organizations focus on the development of tools, modeling techniques and the creation of structures dedicated to knowledge sharing. Knowledge representation, a subarea of Artificial Intelligence, aims to find ways to automatically represent, store and manipulate knowledge using reasoning algorithms (Brachman and Levesque, 2004).

For this reason, the amount of data collected in the health domain increases periodically, resulting in the emergence of diagnostic methods, chemical principles, and advances in molecular biology and genetics, among other medical advances (Wechsler et al., 2003).

Knowledge management and sharing is a promising area, but it is still inefficient in the field of health (Sabbatini, 1998). The knowledge generated through the experiences of the professionals of the area is not usually passed on satisfactorily, thus retained in their own minds (Silva et al., 2005).

In this context, the Knowledge-as-a-Service (KaaS) paradigm aims to provide centralized knowledge that is normally extracted from various data sources and can be maintained by different organizations. In it, a knowledge server responds to requests made by one or more knowledge consumers (Xu and Zhang, 2005).

The aim of this work is to propose an architecture based on the paradigm of Knowledge-as-a-Service, in order to create a common knowledge base that can be used in the medical field or any other health area, and to facilitate the diagnosis of patients, besides the possibility of centrally providing access to ontologies and other means of representing and processing knowledge.

In this article, the following contributions are made: an architecture based on the Knowledge-as-a-Service paradigm in the health area is modeled; a communication API that will be used to transmit knowledge between the knowledge provider and the consumer applications is specified; and an existing prototype in the nephrology domain is adapted to an implementation of the proposed architecture in order to allow its execution and to validate our approach.

Background

Health informatics

According to Hoyt et al. (2008), Health Informatics can be defined as the field of science that deals with formal resources, equipment, and methods to optimize storage, reading and management of medical information in problem-solving and decision making. Thus, Health Informatics aims to improve the quality of health services, reducing costs and allowing the exchange of medical information (Hoyt et al., 2008).

Nephrology and chronic kidney disease

Nephrology is an area of medicine that has as its objective the diagnosis and clinical treatment of diseases of the urinary system, mainly related to the kidney (SBN, 2016).

Chronic Kidney Disease (CKD) is defined as damage to the renal parenchyma (with normal renal function) and/or renal functional impairment present for a period of three months or more. In its last stage, the kidneys are no longer able to maintain the normality of the patient’s internal environment. Thus, early diagnosis and disease prevention have become increasingly important in order to take preventive measures that may delay or halt the progression of CKD (Bastos and Kirsztajn, 2011).

Knowledge representation, ontologies and reasoning

Knowledge Representation is a subarea of Artificial Intelligence (AI) that is concerned with the way that knowledge can be represented symbolically and manipulated automatically by reasoning algorithms (Brachman and Levesque, 2004). Given a structure of knowledge representation and a process of reasoning, it is possible to draw conclusions from previously modeled knowledge. These conclusions can be used to assist in decision making (Ladeira, 1997).

The term ontology, when used in the area of computer science, in the context of knowledge representation systems, refers to a general structure of concepts represented by a logical vocabulary (Russell et al., 1995). To infer in ontologies, we need an inference mechanism called reasoning algorithms. These algorithms allow the comparison of the syntax, possibly normalized structure, and concepts expressed in the ontology (Baader, 2003).

Service Oriented Architectures

Service-oriented architectures (SOA) arose from the need for business integration and automation over the internet (Papazoglou, 2003). In SOA, resources are packaged as well-defined "services" that produce a standardized output independently of the state or context of other parts of the application (Fremantle et al., 2002).

The Software-as-a-Service (SaaS) paradigm describes applications and software delivered as a service over the internet. This architecture has already become an important model for selling and delivering software in various industry sectors, providing several benefits to both service providers and their users (Armbrust et al., 2010), such as cost reduction, elasticity, automatic updates, easy implementation, and a new way of selling functionality to customers that competes with traditional business models (Benlian et al., 2012).

From the users' point of view, the use of the SaaS architecture presents numerous advantages, such as: cost reduction, elasticity, automatic updates and easy implementation. For software developer companies, this architecture offers a new way of selling functionality to their customers and competes with traditional business models (Benlian et al., 2012).

In the context of knowledge sharing and distribution, the Knowledge-as-a-Service (KaaS) paradigm aims to centrally provide knowledge that is normally extracted from various data sources and can be maintained by different organizations. In it, a knowledge server responds to requests made by one or more knowledge consumers (Xu and Zhang, 2005).

According to Xu and Zhang (2005), in an implementation of the KaaS architecture, we can find three main components: Data Owners, which are are responsible for collecting data from their daily transactions and for filtering and protecting the collected information; Knowledge Service Provider, which aims to centralize and provide knowledge through its knowledge server, where data is extracted using an extractor algorithm; and Knowledge Consumers, which are applications that use the provided knowledge in their decision-making process, communicating with the server by using a previously established protocol.

Related work

In order to understand and compare similar researches, this section presents and analyses several proposals of architectures and frameworks applied to different domains.

It was proposed by Grolinger et al. (2013) the Disaster-CDM, a KaaS framework, aiming to be able to store a large amount of disaster-related data from several sources, facilitating its search and indexing in addition to providing support and interoperability tools. Disaster-CDM has as its main form of data storage, the use of Relational Databases (RDB) and NoSQL databases. It communicates with consumer applications in three ways: ontologies, APIs, and services. These three forms of data access allow Disaster-CDM to respond to requests in an integrated manner without depending on how the data is saved internally.

In the Health domain, a relevant article from Yoo et al. (2014), describes a collaborative service-oriented architecture designed to facilitate sharing and cooperation between health care providers, while reducing costs for the patient. The architecture described has three main components: centralizer of medical collaborations; consumer applications and healthcare services providers.

Another research focused on the health domain was done by Lai et al. (2012), which identifies the main characteristics of a collaborative medical services network and, in addition, explores ways in which KaaS could be used to create a private network of medical knowledge in China.

Thus, when analyzing the related work, we can see that service-based architectures can help in decision support in several domains. We can also see that most architectures use a textual format for data serialization and communication with consumers. On the other hand, we can see that the sources of knowledge vary according to the domain and availability of data holding services. Thus, it is necessary to adapt each architecture to its domain, in order to allow a greater use and sharing of available domain knowledge.

Proposed KaaS architecture for E-health

Based on the study and analyses of the architecture proposed by Xu and Zhang (2005) for the Knowledge-as-a-Service paradigm, it was investigated in this work a generic KaaS architecture aimed at the Health domain. For that, a similar, but extended and detailed platform, called H-KaaS, was designed.

H-KaaS aims to centralize access to knowledge generated from several distinct data sources, allowing efficient knowledge sharing. An overview of the H-KaaS architecture can be seen in Figure 1.

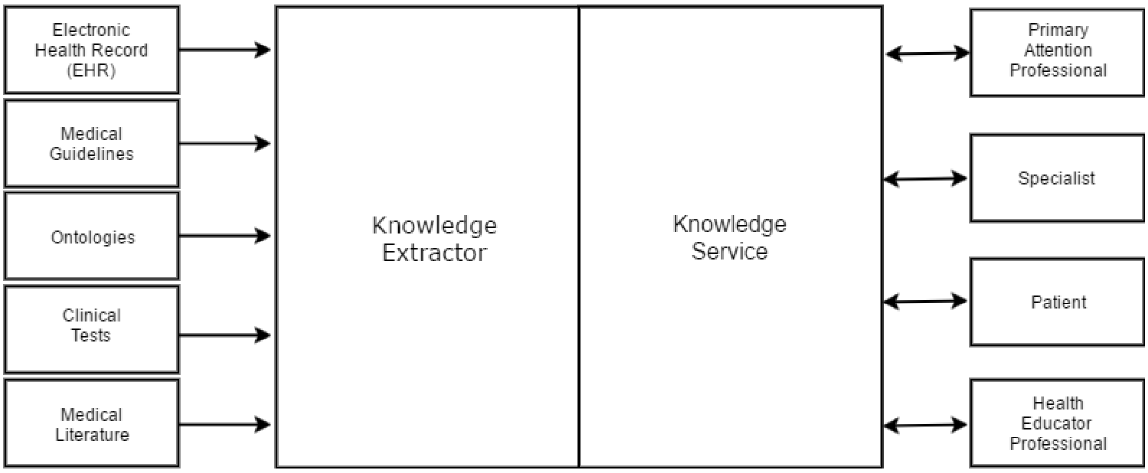


Figure 1. An overview of the H-KaaS conceptual architecture, based on the paradigm of Knowledge-as-a-Service adapted to the Health domain. Source: Own Authorship.

In the medical field, data sources could be considered, for instance: results of clinical tests, domain ontologies, books, periodicals, guidelines for diseases treatment, among others. Each data source has its own characteristics and must be treated independently, in order to facilitate the inclusion and modification of new extraction rules.

For consumer applications, it is possible to create various solutions for each interested part that exists in the domain. For example, applications for clinical decision making processes can be created to improve the service provided by specialists and primary care professionals.

The architecture H-KaaS has been divided into modules in order to be easier to maintain, understand and implement. The main modules that compose the architecture are the (1) knowledge service provider, (2) knowledge consumer, (3) communication API, and (4) data sources. In the next topics, we will talk more about each one of them.

Knowledge service provider

Within the KaaS paradigm, the knowledge service provider module aims to access and process data from the data source, manage knowledge models and serve queries made by knowledge consumers (Xu and Zhang, 2005). In the context of H-KaaS, the knowledge service provider consists of two main parts: the knowledge extractor and the server-side implementation of the communication API.

Extraction of knowledge is made as queries are performed by consumer applications, through inferences in ontologies, queries for documents or other sources of data and knowledge. Each data source has its corresponding extraction rules within the knowledge extractor. Therefore, for each new data source added, extraction and access rules need to be written so that it can be integrated with the existing system.

In addition, within the knowledge extractor, auxiliary algorithms are responsible for several common functions for reading and manipulating information like reading text files, loading and converting images, among others. It is also possible that the knowledge extractor has its own data storage mechanism in order to index and improve access speed for specific queries.

In order to respond to consumer queries, the server implements the communication API, providing standardized responses that can be easily understood by consumer applications.

Knowledge consumer

The H-KaaS architecture, similarly to the KaaS paradigm, provides the possibility of being accessed by different knowledge consuming applications. These applications use the communication API to make queries to the central knowledge base. Possible consumer applications in the Health domain, are clinical decision support websites, mobile applications or embedded systems, electronic medical records, educational systems, and others.

In the security field, each application must have a unique private key that will be used during communication with the service provider in order to authenticate, limit and/or identify the queries made by it. A failure in providing the security key will make it impossible to access the knowledge service provider, which is responsible for creating, storing and sharing the keys.

Communication API

The communication between the knowledge service provider and the consumer applications is made through the communication Application Programming Interface (API), the H-KaaS API. According to Masse (2011), an API is a web service, based on well-defined programming interfaces, which enables communication between applications. In other words, an API can be considered a medium in which communication between two computer programs is allowed.

For the specification of the API, the principles mentioned by Bloch (2006) were used, being the main ones: an API should be small and simple; errors should be reported immediately; a small number of parameters should be kept; and all functions must be well documented. These help to create a good communication API.

Recently, an architecture style for specifying APIs was proposed by Richardson and Ruby (2008), the Representational State Transfer (REST). APIs that follow this paradigm are called RESTful APIs or REST APIs. This architecture uses standards similar to a protocol that is widely used on the Internet, the Hypertext Transfer Protocol (HTTP). In REST APIs there are four main types of commands: GET, POST, PUT and DELETE.

Each command is responsible for an operation on the data. The POST command, for example, is responsible for updating or appending information. Similarly, the DELETE command is able to remove information from the server. The GET and PUT commands respectively are responsible for reading and adding entities on the server (IETF, 2014).

Another important feature of REST APIs is their data output format, where are used semi-structured textual object serialization formats, such as JavaScript Object Notation (JSON) and XML (Richardson and Ruby 2008).

The H-KaaS API is an API based on the REST architecture and uses the JSON data format, which is a light, text-based, language-independent format for information transfer. This format defines some formatting rules that allow the representation of data in a structured way (Crockford, 2006). When designing the H-KaaS API, it has been needed to specify two commands:

- **/service**: Command that receives requests of type GET and lists the services currently implemented in the platform. Each service has its descriptions, identification code and, mainly, a list of the methods available, that can be executed at any time by the consumer application, allowing access to the available knowledge.
- **/service/id/method**: Command responsible for executing a method of a specific service. It receives requests in the POST format and therefore, in order to execute it, the application must provide serialized information based on the input form present in the input key. The response of this command is a JSON serialized object containing some information about the query performed and the result of its execution.

Regarding the error handling and formatting, the H-KaaS API has a special object of class Error that is instantiated and returned when some critical problem occurs during query execution. As for security, for the execution of each API call, the consumer application must send a mandatory security key. This key ensures that access to knowledge is limited only to authorized applications.

Data sources

The H-KaaS provides the possibility of multiple independent data sources, allowing greater flexibility in obtaining information and/or knowledge provided by external services. The data access system can vary according to its source and, therefore, the knowledge extractor is responsible for implementing the necessary means to read and extract the relevant information. Access to data can take place in two ways: local, in which data is organized in the same file system where the service is running; and remote, in which data is transferred from remote sources, occasionally unavailable.

Because remote access is less reliable, it is recommended to use indexing and caching mechanisms in order to improve overall system performance in periods of instability. For some remote data sources, information may be mass-collected for the purpose of local batch processing. Thus, even if the data source stops working after obtaining the data, H-KaaS would still be able to execute the extraction and creation of the knowledge models from the previously obtained data.

Case Study: Providing clinical decision support to CKD

The objective of this case study is to adapt an existing prototype in the Health domain for an implementation of the proposed architecture. Two prototypes were studied in order to choose one to be adapted to the new architecture, both researches have, as knowledge representation system, unique ontologies, created with the help of specialists in the nephrology field.

The OntoDecideDRC, a platform based on web services, aims to support clinical decision to nephrology specialists and those working in primary health care (Tavares et al., 2016). Another similar research was done by Campos et al. (2016) where EducaDRC was created, a semantic repository of learning objects about CKD.

The main difference in the approaches is that EducaDRC, as a semantic repository, needs an additional database for storing triples and metadata, used to describe external resources. Therefore, due to database independence, the OntoDecideDRC was chosen as the ontology and base prototype to be adapted to the H-KaaS architecture.

The knowledge service provider was implemented in the PHP language and, in addition to containing the commands required for the extraction and inference of knowledge, also implements the commands described in the API specification.

For the adaptation of the ontology present in the OntoDecideDRC to the knowledge extractor module, it was necessary to write a command line application, able to execute the reasoner HermiT and infer in an OWL-type ontology, using queries based on Description Logic. This application was implemented in the Java programming language and is part of the knowledge extractor module.

In the knowledge extractor, some auxiliary functions were also created to execute ontology queries. One of these is the implementation of the Glomerular Filtration Rate (GFR) calculation. This calculation uses the simplified MDRD formula specified by Levey et al. (2000) and, through patient information, a numeric value is reached that will be used during inferences in the knowledge model.

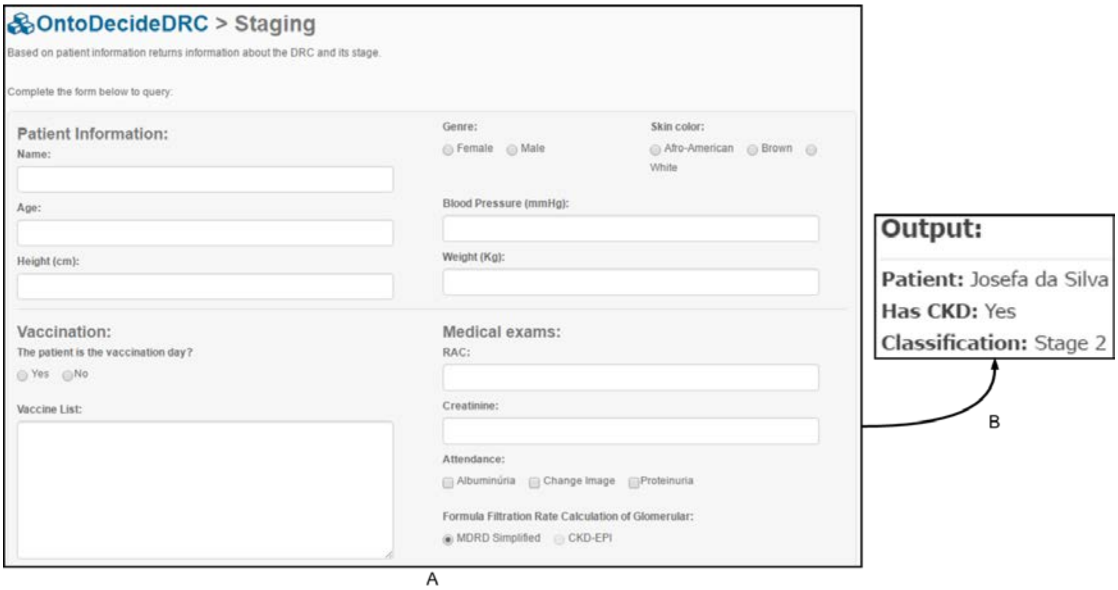
The knowledge consumer module, a website called NefroService, had the objective to test the API and implement the graphical interface of the chosen prototype. The programming language used for the implementation of NefroService was PHP and, in conjunction with the Wordpress framework, made possible the implementation of several pages for login, registration, password recovery, services list, method execution, contact, among others.

When adapting OntoDecideDRC, it was realized that the source code of its graphical interface was not available and, therefore, it was necessary to create a new form for data entry based on the original specification of the prototype. The data provided served as the basis for the staging method of the OntoDecideDRC service on the NefroService. All form fields have been rewritten and organized in a similar way to their original version.

When accessing the NefroService, the user faces a form for inserting its credentials. These credentials are the responsibility of the consumer application, which enables the creation of different levels of access to the application, as well as better control of how information is shared.

Each service has a dedicated page, which can assist the user in choosing an appropriate method. After choosing the method, the user will face the data entry form and a brief description of how to fill it. Figure 2.A shows the staging method form, similar to that implemented in the original prototype of the OntoDecideDRC platform.

This form is generated dynamically based on the data received from the communication API. When submitted, a request is sent to the API and, from it, the HTML code is created with the response that will be shown to the user (Figure 2.B). If during the processing of sending of the request, an error occurs, an error message and a possible solution will be shown to the user.



OntoDecideDRC > Staging

Based on patient information returns information about the DRC and its stage.

Complete the form below to query:

Patient Information:

Name:

Age:

Height (cm):

Genre:
○ Female ○ Male

Skin color:
○ Afro-American ○ Brown ○ White

Blood Pressure (mmHg):

Weight (Kg):

Vaccination:

The patient is the vaccination day?
○ Yes ○ No

Vaccine List:

Medical exams:

RAC:

Creatinine:

Attendance:
□ Albuminúria □ Change Image □ Proteinuria

Formula Filtration Rate Calculation of Glomerular:
● MDRD Simplified ○ CKD-EPI

Output:

Patient: Josefa da Silva

Has CKD: Yes

Classification: Stage 2

Figure 2. CKD staging method form (A) and an example of its response (B).

The implementation of the knowledge consumer website enabled the platform user to perform queries using forms similar to those originally provided by the OntoDecide-DRC platform, the prototype chosen to serve as a source of knowledge and basis for the creation of the graphical interface forms.

Conclusion

In this article, a new Knowledge-as-a-Service based architecture was presented in the health domain, with a detailed description of each of its modules. To validate this new architecture proposed, a case study was made, where an existing prototype in the domain of nephrology was adapted, which allowed the inference in the modeled knowledge base and the creation of a clinical decision support mechanism, similar to the original prototype.

The KaaS paradigm, although relatively new, is promising in terms of distribution and access to knowledge, being possible to use it better in domains where, even there is a large amount of data being collected daily, there are still no efficient ways to share knowledge satisfactorily.

This paper opens the way for new sources of data and knowledge so that they can be developed and made available more effectively. In addition, new applications that use this data in interesting ways can be developed through the use of the communication API.

In conclusion, the development of this research contributed to the emergence of a new architecture, called H-KaaS, which established itself as a knowledge based system capable of managing multiple sources of data and knowledge, centralizing access through an adaptable API.

Conflicts of interest

Authors declare that they have no conflict of interests.

References

Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A. D.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; Rabkin, A.; Stoica, I.; Zaharia, M. A view of cloud computing. **Communications of the ACM**, v. 53, No. 4, p. 50-58, 2010. https://doi.org/10.1145/1721654.1721672

Baader, F. **The description logic handbook**: Theory, implementation and applications. Cambridge: Cambridge University Press, 2003.

Bastos, M. G.; Kirsztajn, G. M. Doença renal crônica: importância do diagnóstico precoce, encaminhamento imediato e abordagem interdisciplinar estruturada para melhora do desfecho em pacientes ainda não submetidos à diálise. **Jornal Brasileiro de Nefrologia**, v. 33, No. 1, p. 93-108, 2011. https://doi.org/10.1590/S0101-28002011000100013

Benlian, A.; Koufaris, M.; Hess, T. Service quality in Software-as-a-Service: Developing the SaaS-Qual measure and examining its role in usage continuance. **Journal of Management Information Systems**, v. 28, No. 3, p. 85-126, 2012. https://doi.org/10.2753/MIS0742-1222280303

Bloch, J. How to design a good api and why it matters. Proceeds of the 21st ACM SIGPLAN Symposium on Object-Oriented Programming Systems, Languages, and Applications, p. 506-507, 2006.

Brachman, R.; Levesque, H. **Knowledge Representation and Reasoning**. San Francisco: Morgan Kaufmann Publishers, 2004.

Campos, S. P. R.; Pegado, C. N. A.; Lino, N. C. Q. Referatório semântico de objetos de aprendizagem sobre doença renal crônica. **Journal of Health Informatics**, v. 8, p. 417-427, 2016. Available from: <http://www.sbis.org.br/biblioteca_virtual/cbis/Anais_CBIS_2016_Artigos_Completos.pdf#page=417>. Accessed on: Nov. 23, 2017.

Crockford, D. **The application/json media type for javascript object notation (json)**. Reston: The Internet Society, 2006. Available from: <https://tools.ietf.org/pdf/rfc4627.pdf>. Accessed on: Nov. 23, 2017.

Fremantle, P.; Weerawarana, S.; Khalaf, R. Enterprise services. **Communications of the ACM**, v. 45, No. 10, p. 77-82, 2002. https://doi.org/10.1145/570907.570935

Grolinger, K.; Capretz, M. A. M.; Mezghani, E.; Exposito, E. Knowledge as a service framework for disaster data management. Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE, p. 313-318, 2013.

Hoyt, R. E.; Sutton, M.; Yoshihashi, A. **Medical informatics**: practical guide for the healthcare professional. 3. ed. Morrisville, North Carolina: Lulu, 2008.

IETF. Hypertext transfer protocol (http/1.1): Message syntax and routing. 2014. Available from: <https://tools.ietf.org/html/rfc7230>. Accessed on: Nov. 30, 2017.

Ladeira, M. **Representação de conhecimento e redes de decisão**. Porto Alegre: Universidade Federal do Rio Grande do Sul, 1997. (PhD thesis).

Lai, I. K.; Tam, S. K.; Chan, M. F. Knowledge cloud system for network collaboration: a case study in medical service industry in china. **Expert Systems with Applications**, v. 39, No. 15, p. 12205-12212 2012. https://doi.org/10.1016/j.eswa.2012.04.057

Levey, A.; Bosch, J.; Lewis, J., Rogers, N.; Roth, D. A simplified equation to predict glomerular filtration rate from serum creatinine. **Journal of the American Society of Nephrology**, v. 11, suppl. 2, p. 155A, 2000.

Masse, M. **REST API design rulebook**. Boston: O'Reilly Media Inc, 2011.

Papazoglou, M. P. Service-oriented computing: concepts, characteristics and directions. Proceedings of the Fourth International Conference on Web Information Systems Engineering - WISE 2003, p. 3-12, 2003.

Richardson, L.; Ruby, S. **RESTful web services**. Boston: O'Reilly Media, 2008.

Russell, S.; Norvig, P.; **Artificial intelligence**: a modern approach. Edinburgh: Prentice-Hall, Englewood Cliffs, 1995. v. 25. p. 27.

Sabbatini, R. Internet e educação médica. **Informática Médica**, v. 1, No. 3, 1998. Available from: <http://www.informaticamedica.org.br/informaticamedica/n0103/editorial.htm>. Accessed on: Nov. 27, 2017.

SBN - Sociedade Brasileira de Nefrologia. O que é nefrologia? 2016. Available from: <http://sbn.org.br/publico/institucional/o-que-e-nefrologia>. Accessed on: Nov. 25, 2017.

Silva, R. P. **Modelo de apoio ao diagnóstico no domínio médico**: aplicando raciocínio baseado em casos. Brasília: Universidade Católica de Brasília, 2005. (Master dissertation).

Tavares, E. A.; Pegado, C. N. A.; Lino, N. C. Q. Suporte à decisão clínica no domínio da doença renal crônica. **Journal of Health Informatics**, v. 8, p. 839-847, 2016. Available from: <http://www.sbis.org.br/biblioteca_virtual/cbis/Anais_CBIS_2016_Artigos_Completos.pdf#page=839>. Accessed on: Nov. 23, 2017.

Wechsler, R.; Anção, M. S.; Campos, C. J. R.; Sigulem, D. A informática no consultório médico. **Jornal de Pediatria**, v. 79, suppl. 1, p. 3-12, 2003. https://doi.org/10.1590/S0021-75572003000700002

Xu, S.; Zhang, W. Knowledge as a service and knowledge breaching. Proceedings of 2005 IEEE International Conference on Services Computing, v. 1, p. 87-94, 2005.

Yoo, J. J.-W.; Gnanasekaran, K.; Cheng, C.-Y. A collaborative healthcare service framework and performance evaluation. **American Journal of Industrial and Business Management**, v. 4, No. 6, p. 274-283, 2014. https://doi.org/10.4236/ajibm.2014.46035