A RESTful Approach for Developing Medical Decision Support Systems

Tobias Weller[1(✉)], Maria Maleshkova[1], Keno März[2], and Lena Maier-Hein[2]

[1] AIFB, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
{tobias.weller,maria.maleshkova}@kit.edu
[2] German Cancer Research Center, Heidelberg, Germany
{k.maerz,l.maier-hein}@dkfz.de

Abstract. Current developments in the medical sector are witnessing the growing digitalization of data in terms of patient tests, records and trials, use of sensors for monitoring and recording procedures, and employing digital imagery. Besides the increasing number of published guidelines and studies, it has been shown that clinicians are often unable to observe these guidelines correctly during the actual care process. [1] The increasing number of guidelines and studies, and also the fact that physicians are often unable to observe these guidelines correctly provide the foundation for this paper. We will tackle these problems by developing a medical assistance system which processes the gathered and integrated data from different sources, and assists the physicians in making decisions, preparing treatment plans, and even guide surgeons during invasive procedures. In this paper we demonstrate how a RESTful architecture combined with applying Linked Data principles for data storage and exchange can effectively be used for developing medical decision support systems. We propose different autonomous subsystems that automatically process data relevant to their purpose. These so-called "Cognitive Apps" provide RESTful interfaces and perform tasks such as converting and uploading data and deducing medical knowledge by using inference rules. The result is an adaptive decision support system, based on distributed decoupled Cognitive Apps, which can preprocess data in advance but also support real-time scenarios. We demonstrate the practical applicability of our approach by providing an implementation of a system for processing patients with liver tumors. Finally, we evaluate the system in terms of knowledge deduction and performance.

1 Introduction

The growing use of sensors in the medical domain, designated devices for recording patient data, and the digitalization of medical knowledge in terms of recording trials or medical guidelines result in large data volumes, which are hard to process and manage by individual physicians. Nowadays, most of the patient data is stored in semi-structured document formats such as spreadsheets, while the results of clinical trials are published directly as text in papers. At the same

F. Gandon et al. (Eds.): ESWC 2015, LNCS 9341, pp. 376–384, 2015.
DOI: 10.1007/978-3-319-25639-9_50

time more and more sensors are being used to observe patients, resulting in large data volumes. As a consequence, not only is it difficult to benefit from all available data in order to solve a particular medical case, it also becomes unfeasible for a physician to mentally process all the patient data according to current clinical studies and keep track of new studies at the same time. To alleviate this situation, we have presented a concept to support clinical decision making using holistic data analysis [3]. This paper shows the realization of this vision using a RESTful architecture for medical decision support systems, which supports physicians at a decision. In particular, we advocate a solution based on formally modeling patient data with RDF and applying Linked Data principles to publish and interlink individual records [3]. Furthermore, we incorporate studies by describing them as formalized rules in RDF.

These rules are interpreted and executed by multiple Cognitive Apps, which are accessible via a RESTful interface and consume and produce Linked Data. This provides flexible and adaptive composition of the system. In summary, this paper makes the following contributions:

1. We describe a rule-based decision system built up from individual Cognitive Apps.
2. We introduce an exemplary Cognitive App for processing medical guidelines, with a RESTful interface and described in Linked Data.
3. We provide a specific implementation for a use case scenario and demonstrate the added value in terms of automatically deduced additional patient knowledge.
4. We show the suitability of the rule-based decision support system for real-time scenarios, while dealing with large data volumes.

This paper is structured as follows: the following section introduces our medical scenario. Section 3 describes our approach toward designing a decoupled REST-based decision support system, while Sect. 4 provides the specific implementation details. We demonstrate the practical applicability of our solution by realizing a specific medical use case and evaluating the added value to the decision support. We evaluate the system in terms of its suitability of supporting real-time scenarios, in cases where physicians need data usable within intraoperative situations (Sect. 5). In addition, we will show how many new facts were generated by applying the inference rules to the patient data. Finally, related work is described in Sect. 6, and we summarize our contributions and provide some conclusions in Sect. 7.

2 Motivation Scenario

Despite the abundance of medical data, currently, the choice of treatment is usually not obvious, as it depends on a wide range of factors. In a previous publication, we defined three concepts [3]:

(i) Patient data represents all data that can be acquired for a patient for whom the treatment plan is prepared. This information can be extracted from images, laboratory reports or other sources of information (e.g. clinical reports,

hospital databases etc.). It can be related to the disease, the organ anatomy and function or general information (e.g. age, habits etc.). *(ii) Factual knowledge* is written down in quotable sources (e.g. clinical guidelines, studies). This allows the physician to make predictions about the morbidity and mortality of the disease and the possible interventions. Guidelines give more specific directions on treatment options for a specific patient. However, they typically merely give rough directions, taking into account only a fraction of the patient individual parameters (e.g. size and number of tumors), while detailed treatment decisions remain to the surgeon *(iii) Practical knowledge* results from experience. It comprises case knowledge that encompasses the ability to interpret patient data, form a prognosis and deduce implications for the treatment, as well as expert knowledge about treatment options and their respective strengths and weaknesses.

The challenges of diagnosing and providing patient individual treatment plans, can be summarized as follows: (i) collect and integrate patient data so that it can be processed and interpreted in a unified manner; (ii) capture factual knowledge given in the form of medical guidelines by formalizing it in terms of rules; (iii) compensate for the lack of observing the factual knowledge. We address all challenges, focusing especially on the last one, by developing an approach for automatically deducing further patient knowledge in the form of mortality probabilities, procedure level of suitability, etc. by applying the rules from (ii) on the patient knowledge base (i), thus compensating for the lack of observing factual knowledge (iii).

3 Developing a Decision Support System

In the following, we describe in detail our design for realizing a decision support system capable of supporting this scenario, as well as being flexible enough for enabling further medical scenarios.

Figure 1 shows a high-level overview of a decision support system for deducing additional patient knowledge. We adopt a classical three-tier architecture including – Data tier, Business-logic tier, and Client tier.

Data tier – semantic knowledge base consisting of distributed interlinked repositories: (1) a central file storage (**XNAT**), which is used to store data generated by users and other systems, and make this data accessible within the knowledge base. (2) a semantic Wiki (**Surgipedia**), which is the data hub in the system and allows modelling metadata and linking it to all knowledge base relevant data instances. For example, Surgipedia contains links to files stored in XNAT or other external data sources. Furthermore, it provides support for annotating the medical guidelines with metadata. (3) a repository for storing **Patient data** where all patient test results are saved, based on a formally specified patient model. (4) a repository for storing factual knowledge in the form of medical guidelines and studies (**Rules**). Rules are formally defined in N3 format, so that they can directly be applied to the patient data. (5) a repository for commonly used data models (**Onto 1**). Data in the knowledge base is published using the Linked Data principles. In this way, the interoperability between the

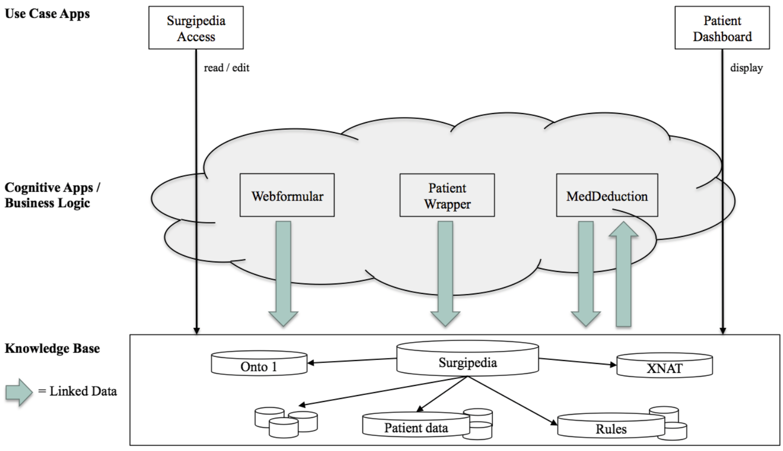

Fig. 1. High-Level overview of the decision support system for deducing patient knowledge

components of the system, the integration of new data into the knowledge base and the ability to interpret this data is realized and guaranteed.

Business-logic tier – implemented via distributed reusable RESTful processing blocks in the form of Cognitive Apps. The communication and interaction between the knowledge base and the Cognitive Apps is based on RESTful interfaces. Cognitive Apps are Web APIs, which also have a semantic description based on the Linked Data principles. The RESTful architecture and the semantic description allows interpreters like Data-Fu [4] to automatically execute Cognitive Apps. This particular scenario includes the following Cognitive Apps – the **PatientWrapper** automatically converts the patient test results into a shared patient model and stores them in the knowledge base. The **Webformular** provides a semi-automated support for converting the medical guidelines into formalised rules in RDF. It takes as input the published studies and the manually composed guidelines via an input GUI and generates corresponding rules. Finally, **MedDeduction** uses the patient data and the rules in order to automatically generate new patient knowledge via deduction and thus provide more data for supporting a better decision.

Client tier – implemented via individual Use Case Apps supporting specific physician decision tasks. For this particular scenario, we have implemented a Use Case App for inputting medical guidelines, in order to assist the process of extracting formalized rules. Another Use Case App provides a user interface for displaying the deduced patient knowledge, including predictions for the morbidity and mortality of a certain intervention (Hepatectomy).

4 Patient Knowledge Deductive System

In this section we describe the implementation of the decision support system, which consists of four Cognitive Apps (PatientWrapper, MedDeduction, PatientGenerator and WebFormular) and the Knowledge Base. These components of the system are implemented in Java. The results of the Cognitive Apps are fed back into the knowledge base. The Apps run on an Apache Tomcat 7 Web Server. The processing steps are executed by performing a HTTP POST method and submitting the corresponding parameters as parameter queries. Only the WebFormular is not RESTful due to the higher number of parameters that are transferred during processing of the inference rules. An overview of the steps of each component in the business tier is given in Fig. 2.

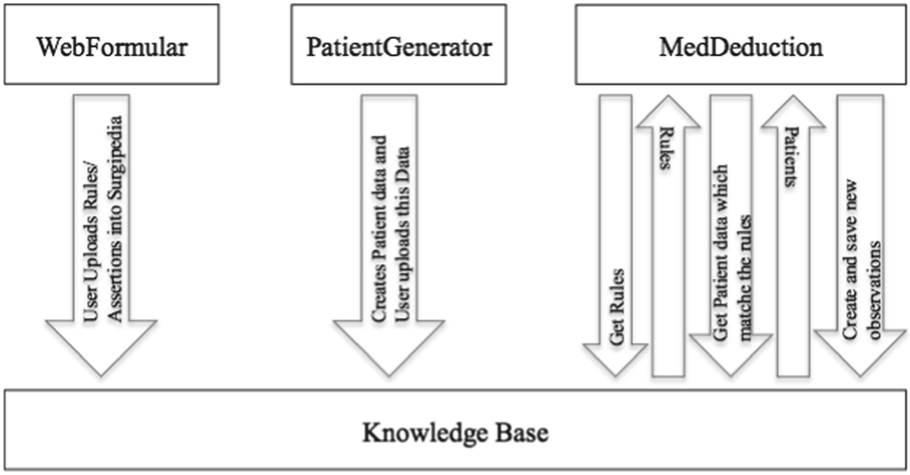

Fig. 2. Overview of the steps of each component in the business tier

The **WebFormular**[1] is a HTML page for entering the medical guidelines in the form of rules. It is divided in two parts – the condition part of the inference rule and the conclusions. An example rule is $\{\texttt{hasHepatectomy} \wedge \texttt{hasHCC} \wedge \texttt{Albumin} < 40\} \rightarrow \{\texttt{Death due to progression after1 Year is 0.1336}\}$ (**Rule ID: 423**). The preprocessing of the entered conditions and implications will be done within a .jsp file. A servlet converts the preprocessed data into the inference rule in N3 format and provides it for downloading. Thereby the user receives the inference rule in RDF and proceeds to upload it into Surgipedia and link it to a wiki page. In addition, the user can enter metadata about the rule. In this way medical guidelines from publications can be stored and described in the knowledge base.

Due to privacy constraints real patient data cannot be used for testing our system implementation. Evaluation has been performed in [3] with a smaller amount of real patient data. This paper validates the technical concept on a large scale by using a large number of patients. Therefore, the **PatientGenerator**[2], which is a Cognitive App that generates patient data for testing the correctness

[1] https://github.com/TobiasWeller/Webformular.
[2] https://github.com/TobiasWeller/Patientgenerator.

and performance of the implementation, as well as for evaluating the knowledge deduction process helps by generating a large number of patients. For example, we generated the Patient 4.899960.7 with an Albumin value of 32.1483 g/dL. The parameters for this patient fit one of the rules. So later on, it will be deduced that this patient has Hepatectomy and HCC.

The **PatientWrapper**[3] performs a HTTP POST request and takes as input the patient data (in a spreadsheet format) and the predefined patient model. The patient data is then transformed into RDF and uploaded to the knowledge base.

Once the rules and the patient data are in the system, we can deduce additional medical knowledge. In order to do this, the **MedDeduction**[4] retrieves the rules from the knowledge base, checks if patients matches the conditions of the rules, and inserts the corresponding deduced new triples. In total it provides four functions. The first function is for deducing all inference rules for all patients. The second deduces one inference rule for all patients. The third tries to deduce all inference rules for one patient. The fourth is for deducing one inference rule for one patient. The Cognitive App can be executed by performing a HTTP POST request and transmitting the corresponding parameters. The HTTP POST request for executing the deducing process is the following:

```
curl POST http://aifb-ls3-vm2.aifb.kit.edu:8080/MedDeduction/
Executer/AllRuleAllPatient
```

Our approach towards implementing the decision support system provides a very flexible distributed solution, where new processing components, i.e. Cognitive Apps, can be integrated on demand, while alternatives providing the same functionality can be used to optimize the results.

5 Evaluation

We evaluate the implemented decision support system based on two criteria – knowledge deduction and performance evaluation. For the knowledge deduction we show that there was new knowledge deduced by applying the formalized rules. For the performance evaluation we show that the system supports real-time scenarios and we compare the results to a local execution of the system. In order to conduct the experiments, we generated 1,000 patients with the help of the Patient Generator. The values for the corresponding factors were randomly generated according to a given range. We used 60 rules, which were taken from studies , and converted with the help of the Webformular. **Knowledge Deduction** In total there were 18,444 new facts generated or rather 129,108 new triples (one fact consists of 7 triples). There were no new patients generated. There was no new knowledge deduced for 4 out of the 1,000 patients, because no rule fit to these patients. The rule that generated most new facts was the following: $\{\texttt{MilanCriteriaFulfilled} = \texttt{True}\} \rightarrow \{\texttt{hasHepatectomy} = \texttt{True}\}$

[3] https://github.com/TobiasWeller/PatientWrapper.
[4] https://github.com/TobiasWeller/MedDeducter.

In total this rule took 9.141 s on the server and was performed for 861 Patients. The rule that took the shortest time was: $\{\texttt{ColorectalCarcinoma} = \texttt{True} \wedge \texttt{hasHepatectomy} = \texttt{True}\} \rightarrow$ $\{\texttt{Probability of 2 YearDisease Free Survival} = \texttt{0.37}\}$
In total, this rule took 0.09 s on the server and did not result in new facts or triples. We looked up among others observations created by these two rules to comprehend the results. The deduction was performed correctly and did not lead to flawed observations.

Performance Evaluation. We compared the performance between execution on a local and on a remote system. The first experiment runs the deduction process locally on a workstation and communicates with the knowledge base over the Web (**Local**). The second experiment runs the deduction process on a remote virtual machine (**Remote**). However, the virtual machine is on the same system as the knowledge base. The local machine has 4 Cores at 2.93 Ghz with 8 GB memory. The virtual machine has 4 Cores at 2.6 Ghz with 16 GB memory.

We compared for each rule the number of new facts that were generated. Both, the local and the remote experiment used the same inference rules and patient data, and produced the same number of facts. All results were valid and the number of facts, generated by both experiments, were the same. Naturally, the runtimes were different. Table 1 contains the measured results. We measured the total runtime for deducing all 60 rules, the shortest time for deducing a rule and the longest time, as well as the average time for deducing one rule.

Table 1. Evaluation local vs. remote deployment

Experiment	Total runtime	Min time p. rule	Max time p. rule	Avg. time p. rule
Local	20.575 min	0.337 sec (0 Facts)	80.31 sec (855 Facts)	0.067 sec
Remote	3.843 min	0.09 sec (0 Facts)	13.066 sec (0 Facts)	0.013 sec

It can be clearly seen that the remote system has and advantage against the local solution. However, this is based on the fact that the deductive system runs on the same machine as the knowledge base. Therefore, no long transmission time is needed to transfer the data. In three cases, the experiment on the remote server took longer than the local experiment. However, these rules had not generated new observations.

In total 18,444 new observations were created during a total runtime of 3.843 min for the remote experiment. This leads to 0.013 facts and rule per second. Since in average the deduction of a fact for one rule takes 0.013 s, we can assume that for 1,000 rules the deduction for one patient sums up to 13 s. This makes the decision support system suitable for real-time use cases, where the physician can immediately take the additional knowledge into consideration for making a decision and planning an appropriate treatment.

6 Related Work

IBM Research developed in collaboration with the Cleveland Clinic Lerner College of Medicine of Case Western Reserve University a medical domain expert system. This cognitive computing technology, called WatsonPaths, supports clinical reasoning by exploring a complex scenario and drawing conclusions. It pulls its knowledge from reference materials, clinical guidelines and medical journals in real-time. On the base of this knowledge, it disproves a set of hypotheses to generate new factors in order to support diagnosis and treatment options.[5]

Another development is the FM-Ultranet [5,6]. This is a decision support system using case-based reasoning in the ultrasonography sector. This decision support system exploits image analysis and pattern recognition techniques to improve and train ultrasound scans interpretation and diagnosis of foetus malformations and abnormalities. Thereby it uses past similar ultrasound scans, stored in the database, for interpreting and diagnosing the actual scans.

CARE-PARTNER is a computerised medical knowledge-support assistance that offers its functionality on the web [2]. It proposes case-based and rules-based reasoning and information retrieval methods to provide useful knowledge to physicians. The system is implemented on the concept of evidence-based medical practice.

7 Conclusions

The increasing digitalization of medical data calls for new solutions that support physicians in planning a treatment strategy. To this end, we introduced a rule-based medical decision support system based on a decoupled distributed RESTful architecture. We combined REST with Linked Data to present a novel approach that has not been previously tested in the medical domain. The system consists of multiple Cognitive Apps that process the medical data. The so implemented system successfully derives additional knowledge about patients, thus assisting the physicians in making decisions. As shown in the valuation section, the system is also suitable for real-time scenarios.

The future work includes the integration of further information sources in order to enlarge the number of inference rules. A significant contribution, therefore, would be the automated extraction of inference rules from studies and guidelines.

Acknowledgments. This work was carried out with the support of the German Research Foundation (DFG) as part of project A02, I01, and S01, SFB/TRR 125 Cognition-Guided Surgery. We would particularly like to thank Patrick Philipp, Mohammadreza Hafezi, Arianeb Mehrabi and Marco Nolden. All of the authors state no conflict of interests. All studies have been approved and performed in accordance with ethical standards. Patient data were gathered and evaluated under informed consent only.

[5] http://www.research.ibm.com/cognitive-computing/watson/watsonpaths.shtml.

References

1. De Clercq, P., Kaiser, K., Hasman, A.: Computer-interpretable guideline formalisms. Stud. Health Technol. Inform. **139**, 22–43 (2008)
2. Bichindaritz, I., Kansu, E., Sullivan, K.M.: Case-based reasoning in CARE-PARTNER: gathering evidence for evidence-based medical practice. In: Smyth, B., Cunningham, P. (eds.) EWCBR 1998. LNCS (LNAI), vol. 1488, p. 334. Springer, Heidelberg (2006)
3. März, K., Hafezi, M., Weller, T., Saffari, A., Nolden, M., Fard, N., Majlesara, A., Zelzer, S., Maleshkova, M., Volovyk, M., Gharabaghi, N., Wagner, M., Emami, G., Engelhardt, S., Fetzer, A., Kenngott, H., Rezai, N., Rettinger, A., Studer, R., Mehrabi, A., Maier-Hein, L.: Toward knowledge-based liver surgery: holistic information processing for surgical decision support. Int. J. Comput. Assist. Radiol. Surg. **10**, 749–759 (2015)
4. Stadtmüller, S., Speiser, S., Harth, A., Studer, R.: Data-Fu: a language and an interpreter for interaction with read/write linked data. In: International World Wide Web Conferences Steering Committee, pp. 1225–1236 (2013)
5. Balaa, Z.E., Strauss, A., Maximini, K.: Fm-ultranet: a decision support system using case-based reasoning, applied to ultrasonography. In: Workshop on CBR in the Health Sciences, p. 3 (2003)
6. Balaa, Z.E., Traphoener, R.: Case-based decision support and experience management for ultrasonography. In: German Workshop on Experience Management, GWEM 2003, pp. 277–278 (2003)