Technical Report

A Generic Approach for Data Management and End-User Development of Clinical Decision Support Systems

Chunli Yan · Helena Lindgren

Abstract The main purposes of clinical decision-support systems (CDSS) are disseminating evidence-based medical knowledge (EBM), supporting a continued medical education, and improving clinical decision making and care. These purposes are traditionally achieved by using solutions that are relatively transparent and explainable to the end user. However, the development and maintenance of such solutions is resource demanding. Currently, there are four challenges existing in CDSSs when adapting to new circumstances. That is, when facing new knowledge, new diseases, different organizations and users with different skills, usually one needs to update the existing CDSS or develop a new CDSS, which requires lots of time and efforts. Hence, this paper aims for reusing an existing CDSS code by virtue of inputs from authorized medical domain expert users, and with minimal requirement of knowledge and software engineers. To facilitate knowledge elicitation and end-user development, an ACKTUS-based architecture for CDSS development and management is presented that contains: I) A knowledge base and a content management system built on Semantic Web technology to achieve modularity, reusability, customisation, and the possibility to allow medical experts to model the medical knowledge and to structure the information that builds up the design of the user interface; II) A user interface and an graphical user interface generator that automatically generates the user interface whenever the user logs in, so that the interface is synchronised with updates of the knowledge base; III) An inference engine that utilizes patient-specific data and applies various rules in the knowledge base to conduct the reasoning and decision making. These modules can be reused when adapting to new situations. A CDSS for dementia diagnosis is developed and used as an example in the presentation

C. Yan
Department of Computing Science, Umeå University, SE-901 87 Umeå, Sweden
E-mail: yan.chunli@umu.se

H. Lindgren
Department of Computing Science, Umeå University, SE-901 87 Umeå, Sweden
E-mail: helena.lindgren@umu.se

of the generic architecture. A pilot study of the CDSS is presented involving four medical professionals with different levels of expertise. The results show how the generic approach allows for easy knowledge representation and management of EBM, supports a continued medical education and may improve clinical decision making and care provision.

Keywords CDSS · Semantic Web · Ontology · Data Management · Knowledge Acquisition

1 Introduction

A clinical decision support system (CDSS) is a system that can effectively manage healthcare data and offer assistance to physicians and other health professionals in cognitive tasks such as clinical decision-making [13]. It can help physicians to make decisions more quickly and improve the quality of decision making. It was found that CDSSs significantly improved clinical practice in 68% of trials [20]. Studies have shown that a CDSS increases the effectiveness of prescribing medication without increasing cost [39]. The objective of a CDSS is also to disseminate new evidence-based knowledge to the clinical physician at the point of care [5]. As such, it can function as a tool for a continuing medical education embedded in a daily clinical practice, provided it applies artificial intelligence methods that are transparent and that can explain its inferences. Consequently, CDSSs are used for the educational purpose and to disseminate consensus guidelines for care, developed by the medical community.

Despite the general consensus that CDSSs are thought to have the potential to improve the healthcare, there are some challenges preventing its broad use. One of these challenges is about its implementation under flexible circumstances. Listed below are four typical situations: I) the *knowledge acquisition* bottleneck in knowledge engineering of medical information [21,15]; II) the *knowledge management* bottleneck and code reusability issue [9]; III) *customisation* to the routines at different care providing organisations, e.g., following different versions of national treatment protocols [34], and IV) flexible support for the development of skills in an individual user (e.g., adaptation to different levels of expertise) [37,38]. In this paper, the term "user" is a physician or other health professional using the CDSS. As can be seen later, the users include ordinary users and medical domain expert users. The latter can (if authorized) directly participate in the system development. Those limitations in CDSSs are described in detail below.

Evidence-based medicine is rapidly increasing knowledge. However, to implement new knowledge into CDSSs is very tedious and slow. In other words, to keep the system updated with new and heterogeneous healthcare data sources requires great efforts [9]. Bennet and coworkers [6] pointed out that "there is stark evidence of a 13-17-year gap between research and practice in clinical care". It indicates that effective methods for transforming scientific results into clinical practice are lacking. Thus, the transformed knowledge

on evidence-based treatments are often out of date by the time they reach widespread use.

During the past 20 years, several task-network modelling languages had been developed to address this [12,14,32,35,43] (for an overview, see [41]). These provided a modelling environment that was graphical, with decision modules and their dependencies visible, in order to allow non-programmers to be active in the modelling tasks. One example was PROforma [14]. Its main purpose was to develop computer-interpretable clinical guidelines (CIG), which has some degrees of decision support and workflow support functions. However, user studies showed that the modelling software was still difficult to use by medical professionals, and the knowledge engineering tasks required extensive time and resources. Furthermore, to integrate the CIG into the system and implement the CDSS, it needed considerable time and efforts also from the software engineers [46]. In order to let the medical domain experts directly edit the CDSSs with minimal involvement of knowledge engineers and software engineers, the method and the interface need to be very simple and intuitive [2,3].

The code of a CDSS is typically not highly reusable for building a new CDSS, due to various reasons, such as, hard coding. It takes a lot of time and efforts to develop a new CDSS for a specific disease [4,17]. In practice, the possibility to transfer the programming code and software between environments or diseases is limited. Meanwhile, when a new disease (e.g. SARS[1]) suddenly breaks out in different places at the same time, it is very important to rapidly develop a CDSS to deal with it locally while gathering more information about the new disease, preferably also through the CDSS. If the code of an already existing and well established CDSS can be reused and quickly developed into a new CDSS, it will save time and potentially people's lives.

There is also a need for easy customisation. Clinicians from different countries or clinicians with different background do not necessarily apply the same methods for physical examination or the same diagnostic criteria [40]. There may be considerably large variations in the way that the clinicians interpret the data that are entered into the system. In a study of the standalone version of the CDSS for dementia diagnosis and management (DMSS) [23], which was further developed to a web version - DMSS-W and will be presented in this article, differences between countries and organisations were observed in versions of validated assessment instruments and preferences regarding what international diagnostic criteria would be applied. It is thus important to be able to tailor a system to different countries or organisations in order to be widely used, while maintaining and mediating the international consensus on the medical knowledge. However, the existing CDSSs are mostly designed for specific organisations and their particular requirements relating to their internal digital infrastructure.

Providing flexible support for skill development of users is a challenge, however, highly important, since the need for support is different depending

[1] https://en.wikipedia.org/wiki/Timeline _of _the_SARS_outbreak

on the professional's knowledge and clinical experience. Moreover, there is a continuing education seen in daily practice, where more experienced professionals guide less experienced. In an optimally digitalised healthcare, this daily evolvement of knowledge and skills should be fed into a system that provides person-adapted support in this evolvement.

Those challenges in today's CDSSs motivate us to set the following aims for this article:

- The expert clinicians of a particular domain could through purposefully designed knowledge modelling tools efficiently and directly develop a CDSS, reducing the need of knowledge and software engineers;
- The developed knowledge base (KB) and interaction design could be shared, easily accessed, and be used for developing additional CDSSs for other diseases;
- The CDSS can be easily adapted to medical guidelines and routines in different organizations without substantial recoding;
- The CDSS users could select reasoning strategies and be provided support that fit their expertise levels.

To realize these aims, the CDSS's adaptability should be greatly enhanced. In this paper, it is done by modifying the traditional CDSS structure into an Activity-Centered Knowledge and Interaction Tailored to Users (ACKTUS) based structure. In ACKTUS, the KB is expanded with more information, so that it includes not only the rules for reasoning, but also the necessary elements that are needed to construct the CDSS user interface. Two additional units are developed in ACKTUS: a content management system (CMS) and a graphical user interface (GUI) generator. Domain experts directly model domain knowledge through the CMS and the knowledge stores in the KB. The GUI generator fetches data from the KB and automatically generates the DMSS user interface whenever the user logs in, so that the interface is synchronized with any alterations in the KB. The inference engine is also developed, and fetches rules from the KB.

In practice, when the knowledge is updated through the CMS to the KB, the knowledge rules and the user interface of CDSS are revised automatically, whereas the code of CDSS does not need to change and the website does not need to be redeployed. Hence, the maintenance of the CDSS is much earlier than ever. As the inference engine is independent, the KB is filled in through the CMS, and the interface is automatically generated by the GUI generator, the code can be easily reused in the new system as well. Each element for constructing the interface is assigned with an organization property to label it with the according organization or country. In this way, different organizations can have their specifically tailored CDSSs. Finally, this system is also able to provide tailored education towards end users with different levels of skills.

In the next section, the methods and theoretical frameworks are described, followed by an introduction to the system architecture in Section 3 and the proposed solution addressing the described limitations in Section 4. In Section

5 the related work is discussed. The article ends with conclusions and future work.

2 Methodology

ACKTUS is a platform for knowledge engineering and interaction design that aims to facilitate the user-driven development of knowledge-based systems by health care professionals. The five modules shown in the right side of Fig. 1 are all part of ACKTUS, where we add the CMS and GUI generator to facilitate authorized medical domain experts to input information directly through the CMS, which is reflected to the user interface by the GUI generator. The new architecture of ACKTUS is an effective tool to generate CDSS. When the CDSS is generated, what identifies the particular CDSS is the knowledge content and behavior.

We also define core ontology as a set of explicit specification of shared conceptualization. It is a basic ontology consisting of only the minimal concepts required to understand the other concepts. The commonalities (general and domain-independent) knowledge between the different CDSSs targeting for diagnosis are extracted and defined as core ontology, which is defined based on the International Classification of Function, Disability and health (ICF)[2] and other medical and health terminologies, for representing medical and health information. An extended version of the Argument Interchange Format (AIF) [10] is used as core ontology that captures relationships that build the guideline-based medical knowledge required for automated reasoning. In this way, when developing a new CDSS, we can just copy this core ontology to generate a new KB. Then the authorized experts model domain-dependent knowledge through the CMS and generate the domain ontology to be shared among different parts of the CDSS. The details about core ontology are elaborated in the next section.

The medical domain, such as dementia, usually contains knowledge that is considerably complex. By virtue of the introduction of ACKTUS and core ontology, we let domain experts develop a CDSS to be used by their colleagues. These experts are no longer passive users, but rather active participants in the development of CDSS. Hence, complex knowledge can be effectively handled.

Finally, for GUI design and for supporting a continuing medical education of both experts and novices, different strategies for reasoning and decision making are taken into accout along with activity-theoretical models of knowledge development. For example, based on how novices and experts in medical fields typically conduct reasoning and decision making [36], two alternative reasoning strategies and their combination are supported in the design and automatic generation of the user interface. Furthermore, based on activity-theoretical models of knowledge development [19], this construction allows the flexibility to move between the strategies, depending on whether the user finds a particular patient case difficult or more straightforward.

[2] http://www.who.int/classifications/icf/en/

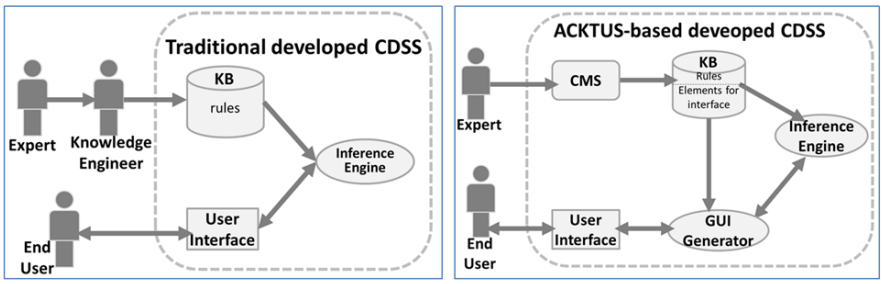

Fig. 1 Traditional and ACKTUS-based CDSS architectures.

The specific technologies used in this research include: Java, Semantic Web, ontology, rdf, Sesame, Apache Tomcat Server, CSS, JSP and so on.

3 System Architecture

Normally, a CDSS contains three modules, as shown in the left part of Fig. 1:

- A *KB* that stores complex structured rules. It is a centralized repository or a database of related information about a particular domain;
- A friendly *user interface* which allows the system to display the decision results to the user as well as have input into the system;
- An *inference engine* that applies the knowledge stored in the KB to patient data retrieved from the user interface to deduct patient-specific recommendations.

Usually the knowledge engineer consults the medical domain experts, e.g. via interview, and fills in the obtained domain knowledge to the KB. The user enters the patient symptoms from the user interface. The inference engine retrieves the knowledge from the KB and patient-specific information from the user interface and performs decision making. The decision is returned to the user interface and shows to the user.

In this paper, for easy maintenance of the CDSS and the broad adaptability across applications, a new architecture is designed which adds two subsidiary units. The new ACKTUS architecture is shown in the right panel of Fig. 1.

The additional two subsidiary units are:

- A *CMS* attached to the KB, which is built on Semantic Web technology to achieve modularity, reusability, customization, and the possibility for medical experts to model the medical knowledge and structure the information that builds up the design of the user interface;
- A *GUI generator* attached to the user interface, that automatically generates the user interface whenever the user logs in, so that the interface is synchronized with updates of the KB without software developer's intervention.

In a conventional CDSS, the KB does not have the function of synchronization with the user interface, hence it contains only rules obtained from the medical domain experts via knowledge engineers. However, in the ACKTUS-based KB, the knowledge includes not only rules, but also the elements for the GUI generator to generate the user interface.

In the ACKTUS architecture, each part of the system is developed separately. The authorized medical domain experts directly model domain knowledge from the CMS, reducing the need of the knowledge engineer, and the knowledge stores in the KB. In this paper, authorized medical domain experts are called authorized experts for simplicity. The GUI generator fetches data from the KB and generates user interface automatically. In the clinical practice, the end user (include both authorized experts and other physicians) fills in the patient symptoms via the CDSS user interface. The separate inference engine module uses these symptoms obtained from the interface and the rules extracted from the KB to conduct reasoning. The engine's assessment is fed back to the interface as an overview of potential hypothetical diagnoses and their supportive and contradicting findings, and advice regarding intervention.

Since the authorized experts can directly model knowledge from CMS, the new knowledge can be included in the KB easily and fast. Meanwhile, because of the GUI generator automatically generates the user interface whenever it is loaded, the web page is synchronized with updates of the KB. Since each part is developed separated and domain-independently, they can be used in other CDSSs easily. This feature is especially crucial when new disease comes. When the GUI generator generates the interface, it will check who the user is and only shows user-related information. That is, the user interface is tailored to the particular user, based on his/her organization and professional skill.

To enable the different modules work together harmonically, the ontology technology is used. Ontology is a concept originally from philosophy and borrowed by computer science to define the types, properties, and interrelationships among the entities that exist in a particular domain. As an explicit specification of a shared conceptualization, an ontology is a knowledge model that represents a set of concepts and the relationships among these concepts within a domain [16]. Ontology is one of the most successful ways to represent medical knowledge [11,33,44], because it helps capture medical knowledge in a formal but simple, powerful and incremental manner, and it can be easily applied in CDSS reasoning process [42].

Ontology is part of *Semantic Web* which was first introduced by Berners-Lee and colleagues to allow data to be shared and reused in the internet across application, enterprise, and community boundaries [7]. A number of languages were defined to provide basic machinery to represent ontologies in the Semantic Web context, such as RDF[3] and OWL[4]. RDF stands for *Resource Description Framework*, a standard model for data interchange on the Web. It uses a triple format of $\langle subject, predicate, object \rangle$, which is a standardized way of describing

[3] https://www.w3.org/RDF/

[4] https://www.w3.org/OWL/?

something. Sesame is a powerful Java framework for processing and querying RDF data. The query language for RDF is SPARQL. In ACKTUS, the RDF, Sesame and SPARQL are applied for handling the ontology.

3.1 The Knowledge Base and Content Management System

The CMS is a web-based knowledge management platform used by authorized experts to represent medical knowledge and design the user interface and interaction. With the CMS, authorized experts with mark-up training can model domain knowledge with correct syntax and semantics. The knowledge can be understood, interpreted and utilized by ACKTUS-based CDSSs through the *ACKTUS core ontology*. The core ontology defines some *key classes* that functions as a universal data structure and shared vocabulary between different ACKTUS-based applications. It is extended with sub-classes and instances as a result of the authorized experts' modeling for each knowledge domain. The core ontology largely consists of three major parts:

- Patient information: a *concept-node system* consisting of a combination of the ICF and other medical terminologies, and extended with *scales* for evaluating the findings;
- Clinical knowledge: an extended version of the AIF developed for exchanging arguments on the Internet [10], mainly consisting of *scheme*, *information nodes (i-node)* and *scheme nodes (s-node)* from which rules are extracted;
- Interaction and GUI design: an ontology for GUI objects and their relations, mainly consisting of templates for information collection and structuring (*interaction object (IO)* and *assessment protocol (AP)*), but also for reasoning guidance (*reasoning-context* and *critical-question (CQ)*).

All information and knowledge that the authorized experts model is basically the instances of these classes or the relevant sub-classes of them. The working mechanism of the core ontology is shown in Fig. 2, where the dashed frame indicates the KB. Basically, the instances of APs, IOs and schemes are created by the authorized experts through the CMS. An IO with each of its scale values automatically forms an i-node. The i-nodes are combined into an s-node which is an instantiation of a scheme. A scheme is a part of a reasoning context. The logic relations used to link these elements together are obtained from the CMS, which is again the experts' input. The classes of the *core ontology* have different properties that describe the classes as comprehensively and detailedly as necessary, which are shown in Table 1. Now we give details about the key classes.

3.1.1 Interaction Object and Assessment Protocol

The authorized experts use IOs to compose structured information templates for the data collection, following the expressions in medical literature. Each *IO* is used for elaborated knowledge such as symptom manifestations, syndromes

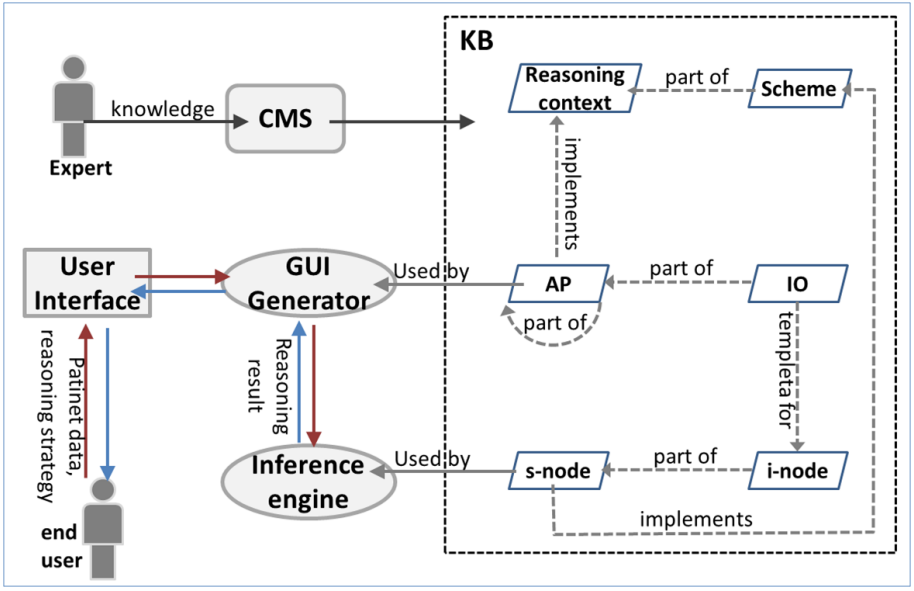

Fig. 2 Workflow of knowledge engineering and automated process that result in a CDSS that can be immediately verified by the expert and used by the user.

and diseases and the evaluated observations obtained from laboratory examinations. An IO has a *reliability scale* for measuring the presence of a certain phenomenon (e.g. [*normal, unknown, affected*] or [*absent, unknown, present*]), which is an obligatory input for the experts. If a phenomenon (e.g. syndromes, disease) is present, it can activate an additional scale - *severity scale* (e.g. [*not specified, mild, Significant*]), for assessing the severity of the phenomenon. For instance, the IO "Judgement" in Fig. 3 contains a reliability scale [*normal, unknown, affected*] and a severity scale [*not specified, mild, Significant*]. Apart from these two types of scales, other types are also defined,e.g., *miscellaneous scale* and *time scale*.

The APs, at different levels of specificity, are ordered collections of IOs and/or other sub APs, composed as protocols for assessment. It helps the user in basic data capture activities. The APs and IOs form a hierarchical tree structure, while the top AP is the root node and the IOs are the leaf nodes. There is no restriction on the depth and width of the tree, except from a usability and readability perspective.

In order to realize the flexibility and adaptability in the generation of user interfaces and results of the inference engine, the following four *"key AP instances"* and their relationships are defined and stored in core ontology. They are key functionalities in most CDSSs that target diagnosis:

The **application AP** (the root node):
- The **data capture AP**,
  - The **reasoning-context-based AP**;

Table 1 Key classes in the core ontology and their properties.

class name	**property name**	**example of implementation**
interaction object	has-term	Judgement (en)
	has-term	omdöme (sv)
	has-info	Specific mental functions especially dependent on the frontal lobes of the brain, including deciding which behaviours are appropriate under what circumstances (ICF). The symptom is a key criterion for frontotemporal dementia. (en)
	has-reliability-scale	normal/unknown/affected
	has-severity-scale	Not specified/Mild/Significant
	has-concept	concept id of "Judgment"
	has-organization	specific organization/individual
assessment protocol	has-name	Status (en)
	has-description	current state of the patient (en)
	has-order	the included APs and IOs
	has-organization	specific organization/individual
i-node	is-related-to	IO id of "Agressiveness"
	has-text-value	Significant episodic memory deficit is present
	has-reliability-value	affected
	has-severity-value	significant
scheme	has-premise-description	Apraxia and aphasia is present.
	has-conclusion-description	Desc_CBD_present.
	has-knowledge-source	knowledge source: CBD_GL
s-node	has-premise	i-node id of "Significant episodic memory deficit is present."
	has-conclusion	i-node id of "A state of dementia is present."
	has-status	validated/under discussion
reasoning context	is-activated-by-cq	CQ id of "Which type of cognitive disorder is present?"
	context-includes-scheme	Dementia_DSM-IV_Scheme
	has-prev-step	reasoning context id of "Step: Is there a cognitive disorder?"

– The **diagnosis and intervention AP**.

These key instances have dedicated purposes and fixed ids in the application, so that the GUI generator understands where to retrieve the relevant data. However, the name, description and included sub AP/IO of the key instances are modifiable through the CMS. The *application AP* is the top level AP (the root node in the tree structure), that defines the "application". From the application AP, the GUI generator retrieves all included APs and IOs that are the children and grandchildren of it. The application AP has at least two second-level sub APs: the *data capture AP* that dedicates for capturing the patient-specific data and the *diagnosis and intervention AP* for showing the diagnosis and intervention results. The *reasoning-context-based AP* is a sub AP of the data capture AP for guiding the users to speed up the reasoning procedure. The ids of the four key APs are stored in the corresponding interface also, so that the inference engine knows where to fetch/feed the data. By defining these key instances, the GUI generator and the inference engine

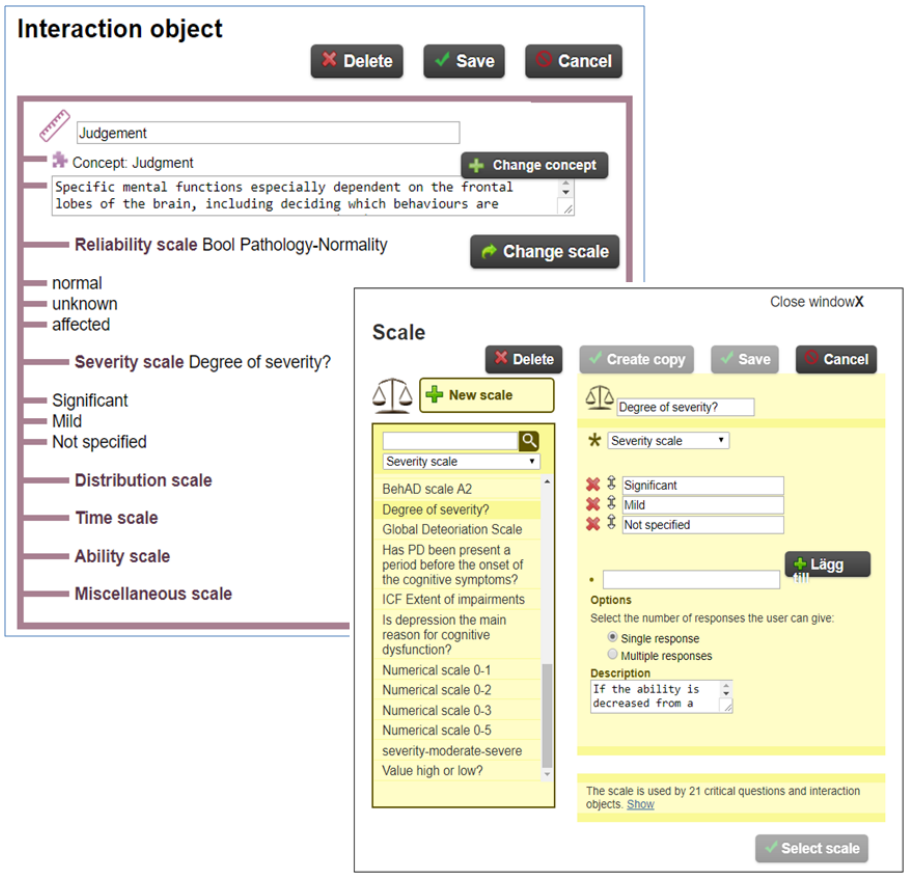

Fig. 3 The snapshot of the IO "Judgement" under edit mode in the CMS. The lower right panel appears when the user clicks "Change scale" in the upper left panel.

can be applied in a new ACKTUS-based application without modifying the programming code.

3.1.2 Scheme and Knowledge Source

The *schemes*, also known as *argument schemes* in our case, constitute an important structure in argumentation theory, which enables the application of general patterns of reasoning to arguments expressed in a local context of argumentation [8,45]. The schemes are described as reasoning patterns that provide a structure of inference in the valuation of arguments. In our approach, a scheme is a semi-structured partial interpretation of a clinical guideline, or diagnostic criteria. To completely represent a diagnostic criteria provided in natural language, a set of reasoning patterns, or schemes are typically defined by using the CMS. For each scheme a set of s-nodes can be defined, that becomes the instantiation of the scheme, from which rules can be extracted.

Each scheme is associated to a *knowledge source* (e.g. *best practice guideline* or *clinical practice guideline*) and each knowledge source is categorized into different types (from high to low): clinical-practice-guideline, consensus-guideline, best-practice-guideline, general-literature, domain expert (rule-of-thumb), knowledgeable professional, novice professional. Based on the type of the source, the scheme is assigned with a *preference level* determining the priority level when it is transformed into a rule by the inference engine, e.g. evidence-based medical studies (include *clinical-practice-guideline* and *consensus-guideline*) are most reliable, while the rule-of-thumb from a doamin expert's experiment is less reliable.

3.1.3 Information Node and Scheme Node

An i-node is automatically generated based on an IO together with its scale value when the IO is created by the authorized expert. It is typically labeled for enhancing the usability. An s-node combines i-nodes into structures which carry *procedural* knowledge. Each s-node is associated to a *scheme*, which in turn is associated to a *knowledge source*. The i-nodes are used as premises or conclusions of the s-nodes, and as such, also as components for generating *explanations* of hypothetical diagnoses, i.e. arguments with certain strength. Fig. 4 demonstrates an example of an s-node and its related scheme (AD_DSM-IV_Scheme) and the knowledge source (see "DSM-IV-R" above the scheme)).

In the ACKTUS ontology the AIF nodes are extended to incorporate values representing strength and severity, as well as a concept identifier drawn from international medical classifications or terminologies whenever possible. The purpose is to verify that two arguments about clinical evidence deal with the same piece of evidence. The ability to add a concept identifier and use terms from international classifications is essential in order to verify the content of the reasoning and for allowing reasoning across professional, language and organization borders. This functionality promotes the development of a common understanding of the content.

It should be noted that in our approach all findings (i-nodes) are considered as *defeasible facts*, since a second assessment by a different person may contradict the current information, and each assessment can be questioned due to a progress of the disease.

3.1.4 Reasoning Context and Critical Question

The s-nodes in the ACKTUS ontology implement the content of knowledge sources. The schemes implement the different contexts of interpretation of evidence including associated value orders extracted from the clinical guidelines. In complex knowledge domains such as dementia, diagnostic criteria contain circular definitions, apply partially overlapping findings and two different diagnostic criteria for the same diagnosis, and may contain contradicting information. Thus, it is valuable to define *reasoning contexts* as a selection of guidelines to be applied in different steps in a diagnostic reasoning process.

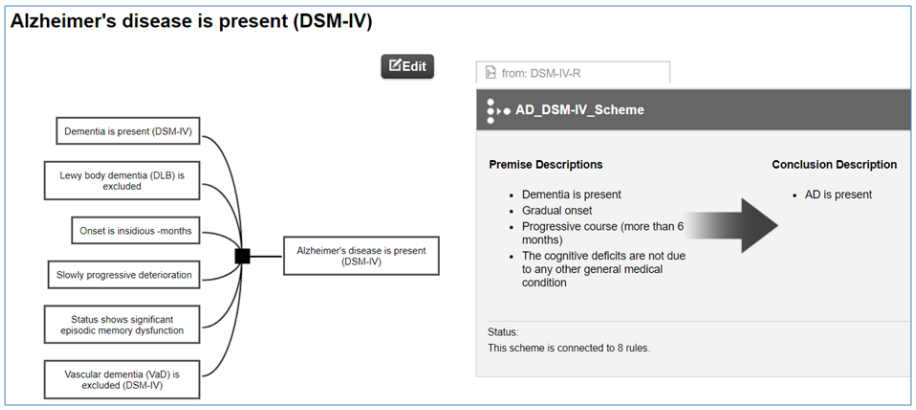

Fig. 4 Screenshot of an s-node and the related scheme extracted from the CMS.

It helps the user in higher level reasoning and decision making. Typically, the process begins broadly for detecting pathology towards refining information and narrowing down the decision space to a set of hypothetical diagnostic conclusions with high specificity and reliability.

The authorized expert defines a reasoning context by the *CQ* that will be answered through the set of schemes that are associated to the reasoning context. When a user activates the *reasoning-context-based guide button*, e.g. clicking the button *What to do next?* in DMSS-W interface (in the bottom left part in Fig. 5), the system guides the user through three steps to find the answers to the three critical questions, following the subsets of medical guideline contents defined by the authorized experts.

3.2 User Interface and GUI Generator

The GUI generator is a program developed using Java, jquery[5] and CSS. It searches the tree structured data and extracts the data by a filter based on the properties *has-organization* of the data, the user's professional skill and the selected language, until finally turns the filtered data into the actual interface. The user interface is changed simultaneously with the contents in the KB without redeploying the website. Hence it is easy to extend its content.

When a user logs in, the GUI generator fetches the *application AP* and retrieves all the sub APs and IOs contained in its hierarchy. The second-level APs are used for generating tabs (e.g., *data capture* tab and *diagnosis and intervention* tab) and the lower levels APs are for menus and submenus (See Fig. 5). However, the *reasoning-context-based AP* is special and is used for generating a button (e.g., *What to do next?* in Fig. 5).

The IOs are primarily used as checklists as shown in Figs. 5 and 6. If it is in *diagnosis and intervention* tab, the items in the checklist can not be checked

[5] https://jquery.com/

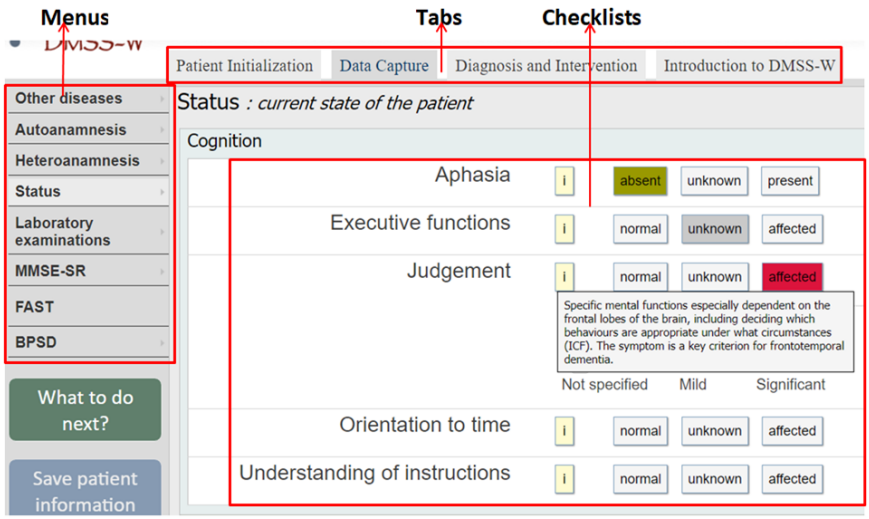

Fig. 5 Example of the *data capture* tab in DMSS-W demonstrating the function of the GUI generator.

by the users as in other tabs, since its selection lies with the inference engine. Also there are two more buttons for each checklist, where *Base for diagnosis* is to show the explanation of system's decision and *select diagnosis* is for the end user to express his/her own decision (See Fig. 6). The information associated to each IO (*has-info* property) that is shown when hovering over the yellow i-button is to provide the users knowledge domain specific explanation of the concept and instruction.

3.2.1 Multilingual Function

The KB supports multilanguage, thanks to the RDF standard. For example, the authorized experts can enter an IO's name (property name is *has-term* in the ontology) and description information (property name is *has-info*) in different languages from the CMS. Fig. 7 is an example for editing an IO's English name through the CMS. Presently, ACKTUS supports five languages: *English* (default), *Swedish*, *Chinese*, *Japanese* and *Korean*. In addition, a new language can be easily added with only minor changes in the ACKTUS code. When a user logs in to the CDSS, he/she chooses a preferred language. Then the GUI generator generates the interface with the chosen language. If the data of the preferred language is missing, the English version will be shown.

3.2.2 Designing Support for Various Reasoning Strategies

The GUI generator supports two complementary reasoning processes, according to how medical professionals conduct clinical reasoning and decision making [36]. When a clinician applies the *forward-chaining* diagnostic reasoning

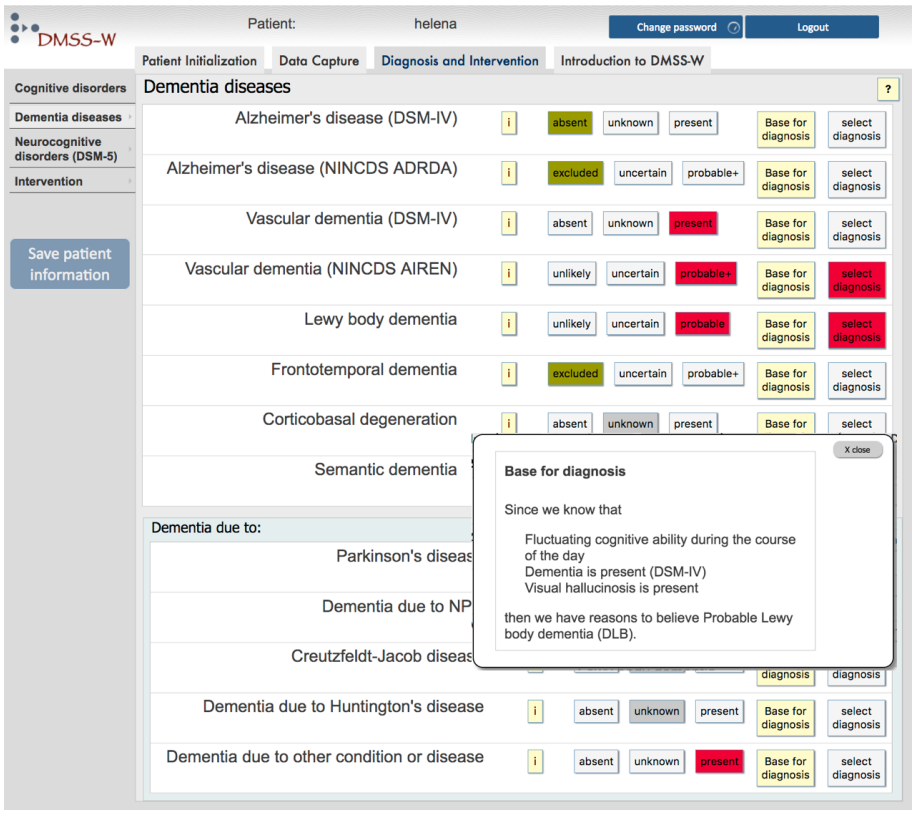

Fig. 6 Overview of the diagnostic results.

method, clinical assessment and investigations are typically conducted before potential hypotheses are generated and evaluated. The corresponding procedure when using the ACKTUS-based CDSS is entering all available information and findings in the checklist format generated in the *data capture* tab of the application, and then apply automated reasoning generated by the engine by activating the *diagnosis and intervention* tab. Based on information available in the *data capture* tab, the system generates hypothetical diagnoses and their strengths, in accordance to a set of international medical diagnostic guidelines (Fig. 6). The results are presented to the user through *diagnosis and intervention* tab as diagnostic conclusions and their strengths and support, based on different diagnostic criteria. If the patient information is not sufficient for a diagnosis, this is also presented, along with what information is missing. The tentative diagnoses may be conflicting in case different diagnostic guidelines are providing different and contradictory results. This kind of information is also important knowledge to be mediated to the user, and the user can then make an informed decision by e.g. selecting which medical source is preferred in his/her decision making.

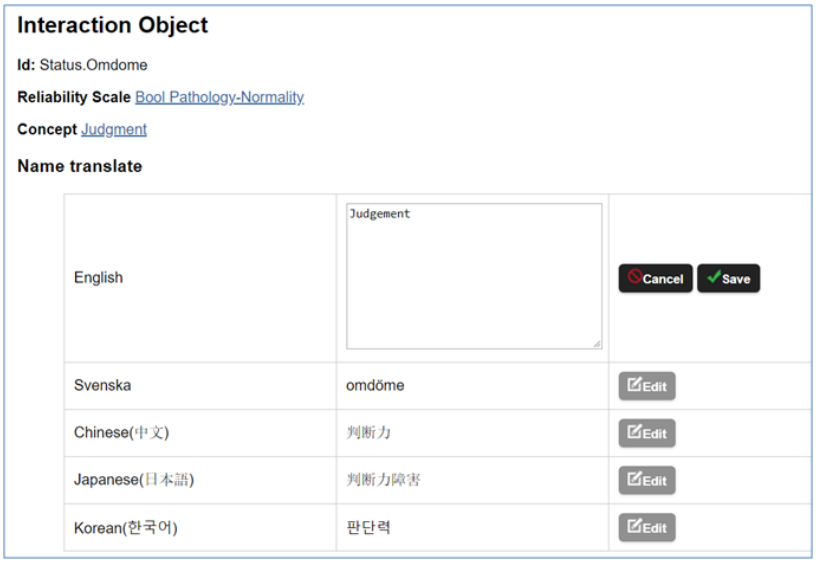

Fig. 7 Snapshot of the CMS interface in edit mode for editing an IO's English name.

Another *forward-chaining* approach that the user can apply, if not familiar with the medical domain, is to use the *reasoning-context-based guide* button (called *What to do next?* in DMSS-W), which will guide the user one step at a time towards diagnosis and intervention (Fig. 8). When the user activates this functionality, the system generates a small checklists with a subset of IOs for the user to fill in, and for each step, information about how to proceed and the sub-conclusions that can be made about diagnosis are provided. The sub-conclusions are answers to the critical questions that defines each step. When completed the final step, a list of supported hypothetical diagnoses are presented to the user to reflect upon.

The opposite strategy, which is typically seen in novice clinicians, is the *backward-chaining* causal reasoning method where the reasoning begins in a hypothetical diagnosis, e.g. Alzheimer's disease, since it is the most common dementia disease. The risk with jumping to conclusions is to miss less common diseases and, therefore, the interaction design of the CDSS is promoting the diagnostic reasoning strategy. The CDSS design allows the user to use the inference engine without conducting a thorough assessment. Then the user will be provided the overview of weakly supported, or unknown support for different potential diagnoses, with information about what patient information is missing for each potential diagnosis.

3.3 Inference Engine

The inference engine is developed using Java. From the *data capture* tab of the CDSS interface, the end user inputs patient-specific scale values of the IOs.

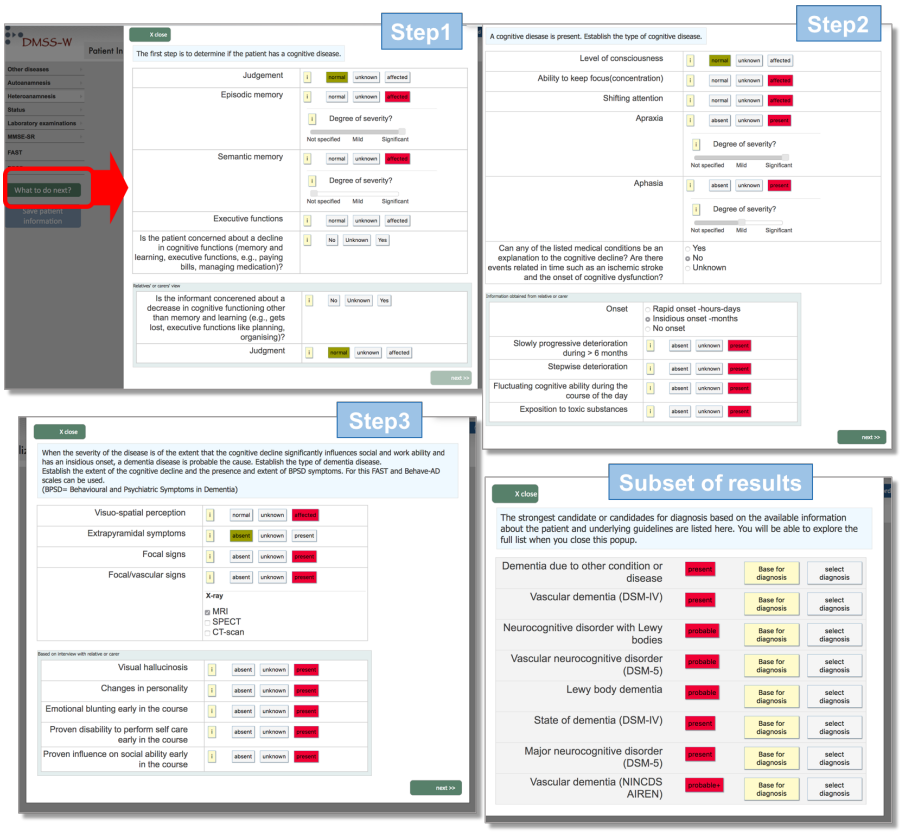

Fig. 8 Overview of the three steps of assessment following the reasoning context-based guide.

The IOs together with the inputted scale values are regarded as *facts*. If-then *rules* are generated from s-nodes in the KB. Using the *facts* and *rules*, the inference engine can perform reasoning.

The information in the KB can be inconsistent, since the sources of the knowledge may be conflicting and ambiguous in complex medical domains such as dementia. Therefore, some conflicting *arguments* could be generated during the intermediate reasoning process. To manage the conflicting results and maintain transparency, each rule is assigned with a scheme that relates to a knowledge source with a certain type. For example, a rule-of-thumb based on an expert's experience is less reliable than a rule obtained from an international consensus guideline. Each argument is given its possibilistic value based on the reliability of the knowledge sources it applies. Using the possibilistic value, a result can be reached by the inference engine. That is, the argument with higher reliability wins. An overview of the reasoning results are shown to the user in the *diagnosis and intervention* tab (Fig. 6), where the user can explore the arguments in favor for and against different diagnoses based on the different

knowledge sources by clicking the button *Base for diagnosis*. However, it is the end user that finalizes the decision by clicking the button *select diagnosis*.

4 Results

As mentioned before, the ACKTUS architecture for building CDSSs aims at addressing the four challenges for rule-based systems. In the following sub sections, the application results are summarized.

4.1 Knowledge Acquisition When New Knowledge Appears

Today, we live in an era of knowledge explosion, and efficient tools for acquiring and disseminating new knowledge obtained from evidence-based clinical studies to clinical practice are needed. The CMS can create semi-formal interpretations of clinical diagnostic guidelines, and also has an ontology-based knowledge editor.

When new versions of diagnostic guidelines are published, the authorized experts can add or update the knowledge through the CMS with minor effort and the knowledge goes to the KB. With the GUI generator, the changes are immediately shown in the user interface, which allows the experts or the colleagues to review the content and verify the interpretations of the informal text-based guidelines if desired. The dynamic generation of the user interface that is not restricted by the limitations of standard relational databases, also allows for rapid modification of content. This is particularly useful during the development phase.

4.2 Rapid Prototyping, Evaluation and Deployment for a Different Disease

The modules in ACKTUS are developed separately. Each module can be replaced or upgraded individually. However, each part strictly follows the same data structure so that they can be integrated easily and communicate with each other seamlessly.

When constructing a new system, a KB is deployed, either using the basic core ontology, or reusing one of the extended KBs, e.g. the dementia KB. A new project in ACKTUS is created by defining a link to the new KB, and linking the inference engine and the GUI generator to the project. This can be finished within one hour. The new CDSS is ready for testing and evaluation as soon as the authorized experts enter the content and knowledge structures from the CMS, which transform into interface elements and rules. This can take from hours to weeks or months, depending on the complexity of the domain, and to what extent the content needs to be negotiated among experts.

Actually, we have tried our approach in another project - Rehab [30, 22], which is for supporting elder adults at home. In that project, experts in the occupational therapy fill in domain knowledge through the ACKTUS. Then,

we link the GUI generator and the inference engine initially developed for DMSS-W to Rehab ontology and, quickly, a new project emerges. Rehab and DMSS-W share the same inference engine and similar interface style, although their KBs are totally different in content.

4.3 Customization towards Different Organizations

When the first version of the standalone DMSS was developed, the KB was built in collaboration with a group of domain experts active in one particular region. Then we evaluated it with physicians in this region and received positive feedback. However when we tested it in another country, the doctors informed us they do not use certain checklists and would like us to remove them from the interface. This shows that the routines and preferences regarding diagnostic guidelines were different between regions and countries [24]. Therefore, DMSS was transferred to the ACKTUS platform to facilitate the development of customized versions, and allowing the domain experts from different regions to develop their customized versions [25,31].

Each element for building the interface can be labeled with an organization property. When a user logs in, the GUI generator is triggered and fetches the GUI objects. The filter in the GUI generator plays a key role. If the organization property is empty, the object is shown to all the users; otherwise, the generator only shows it to the relevant users. In this way, different organizations can have their tailored CDSS.

Because of the language-tagged feature of the RDF format, and the use of the GUI generator, different language can be shown depending on the user's choice.

4.4 Flexible Support tailored for Users with Different Expertise Levels

Since the CDSS interface supports different reasoning procedures, different users can choose different procedures that are suitable to individuals. The novice physician can choose backward reasoning, that is, he/she goes to *diagnosis and intervention* tab to check which symptoms the patient is supposed to have for a particular disease and then goes to *diagnosis and intervention* tab to fill in the symptoms. Another method for the novice physician to apply is to use the *What to do next?* functionality, so that the system can guide the user step by step towards the decision. In contrast, the expert physician usually uses forward chaining to diagnose patients. He/she can collect patient symptoms from *data capture* tab and then go to *diagnosis and intervention* tab to check the system's decision.

4.5 End-User Development

From the description of how the ACKTUS-based system addresses the four challenges above, one can identify a clear feature of our system is the end-user development. Previously, we have conducted studies to evaluate how medical experts without experience in knowledge engineering approach the tasks of knowledge modeling and designing the interaction using ACKTUS [25,28–31]. In our case, the authorized experts are able to model the content and interaction with the system [28,30]. Moreover, they can revise and test the application in order to follow international medical knowledge [25,31]. It is also observed that when experts model the knowledge, they become more careful in how to interprete the underlying clinical guideline and resolve ambiguities, and tend to create more strict rules compared to when mediating their knowledge through a knowledge engineer [24,29,31].

4.6 A Case Study

A case study is conducted to assess how four physicians with different levels of expertise in diagnosing dementia apply DMSS for the first time in two patient cases each. One purpose is to distinguish between obstacles that are due to limitations in dementia knowledge, and obstacles due to the interaction design of the CDSS. We choose to focus on the clinicians' first two cases since these would reveal interaction design issues that may prevent the users from proceeding to becoming a skilled user of the CDSS. This is essential especially in the primary care environment where the frequency that each physician assesses a new dementia cases is low, possibly only a few cases each year. The participants' experience ranges from being novice (two participants), somewhat knowledgeable (one participant) and knowledgeable (one participant). We are particularly interested in how the two reasoning methods are used, and how the participants interpreted the severity values.

Preliminary results show that the two novice users find the assessment of each symptom as a challenge, with their limited knowledge in the domain. They use the information related to each symptom to learn. However, they have difficulties to perceive the overall scope of dementia assessment and its different areas of interest. An initial conclusion is that the participants with this level of experience (they have met only a few cases) need more substantial introduction to dementia assessment guided by a more experienced physician.

The two participants who have more experience, but are not yet experts are able to assess their dementia cases using the system in the linear, checklist manner. The person who considers herself knowledgeable with experience from about 100 cases can efficiently utilise the system to evaluate her own assessment, moving back and forth between data capture and diagnosis functionalities. The person with some experience after meeting about 30 patients, can also complete assessments. However, with less confidence in her own assessments, she is also less certain about the suggestions provided by the system.

These observations suggest that the DMSS has the potential to function as the instrument for a continuing medical education in the dementia domain for clinicians with some experience in dementia assessment, while novices with very minor or no experience need to combine the use with medical education and training.

Between the two methods for assessment, the checklist approach and the context-based guide, the participants select the checklist approach in the few observed cases, which gives the overview of symptoms and a list of assessment forms for different categories of symptoms. They perceive also this as the major benefit in a continued use of the system, to prevent that they could miss some vital information. When asked, they also test the context-based approach.

Key to dementia diagnosis is to assess progression of the symptoms, and decide when symptoms are *mild* and when they have progressed into being *significant*, meaning that they affect performance in daily activities. How to assess this is brought up by the two more knowledgeable participants, and the person with most experience find that she will adjust her assessments based on the suggestion from the system, which mediates the definition that is applied in the major clinical guideline [1]. This case study is an example confirming that apart from the diagnostic function, our DMSS can also provide a continued medical education, where supports tailored to individual user's expertise level are feasible by our approach.

5 Discussion

PROforma was a task-network modelling language used to model clinical guidelines into computer interpretable knowledge. The system developed using PROforma normally had to be linked to a separate and appropriately tailored application-specific GUI in order to work properly. In some aspects, ACKTUS follows the generic symbolic decision theory including arguments, provided by the CREDO program that includes PROforma [13]. PROforma contained a simple version of argumentation, where strict rules and defeasible rules, or arguments, could be defined and executed. The number of defeasible arguments in favour or against a conclusion were simply counted to aggregate strength for a conclusion. However, a strict rule would defeat all defeasible rules. This approach was found to be too limited for the dementia domain, where more levels of uncertainty were expressed in the guidelines (as shown in Fig. 6). In contrast, ACKTUS includes a possibilistic logic framework for extending the management of uncertainty. ACKTUS is designed to address the following limitations of earlier task-network modelling languages: i) usability for non-programmers, ii) expressiveness of uncertainty, and iii) the possibility to define more loosely coupled workflows that allows different reasoning strategies in the end user (e.g. novice vs. expert), which is accomplished using reasoning contexts and assessment protocols in ACKTUS.

There are other tools for developing expert systems, available both as commercial software (e.g., *Exsys*[6]) and freeware (e.g., *DEXi* [18]). They are generic approaches, but the users need to have expertise in programming. Compared to ACKTUS they are more complicated to use and require installation. Moreover, typically standalone applications are used for building the KB, whereas ACKTUS allows multiple people to work on it simultaneously through the web application. Unlike those, ACKTUS also allows for the customisation to different organisations and provides multilingual functionality. There are benefits from allowing flexible generation of user interfaces. *Exsys* is one example, which generates user interfaces by asking the user questions one by one, or category by category, and finally it finishes with a summary page. A similar strategy is applied in our work on risk management for preventing occupational injuries in the mining and construction industry [27], and in the rehabilitation domain [26] when the end users are potential patients. However, this approach would not work when an overview of the information is necessary, which is typically the case for medical professionals. The alternative method applied in the CDSS exemplified in this article, offers at least two benefits: 1) an overview of all the questions is provided; and 2) the users can go back and forth between different tabs and menus without losing any data. This has been accomplished by condensing the questions (checklists) into just a few pages to allow overview, and these pages are designed such that they are in essence one single jsp page in the backend software, although techniques are used to make them appear as if they are in several pages when the users click on them. In this way, we can make sure that the users can navigate back and forth between the "pages" and menus without losing any information.

6 Conclusions and Future Work

We have presented a data management and end-user development solution for reducing the resources required for knowledge acquisition, knowledge management and customisation of CDSSs. The CDSS contains: I) a KB and a CMS built on semantic web technology to achieve modularity, reusability, customisation, and the possibility to allow medical experts to model the medical knowledge as well as structuring the information that builds up the design of the user interface, and immediately evaluate their results; II) a user interface and a GUI generator that automatically generate the user interface whenever the user logs in, so that the interface is synchronised with updates of the KB; and III) an inference engine that can be reused by other CDSSs. We note that the CMS and GUI generator are two additional element in the ACKTUS-based architecture as compared to conventional structures.

When the data of the KB is changed through the CMS, the knowledge rules and the user interface of CDSS are revised automatically, whereas the code of CDSS does not need to change and the website does not need to be redeployed.

[6] http://www.exsys.com/

Importantly, all of these can be conducted merely by the expert users with only minimal involvement of knowledge and software engineers. In this way, the maintenance of the CDSS is improved compared to traditional CDSSs, and the code can be easily reused in a new system. The different parts in the ACKTUS-based architecture is developed separately, and can work together harmonically. Using our approach, the four challenges currently existing in CDSSs are solved or relieved.

Since disseminating new knowledge and providing a continued medical education (intelligent learning) in order to improve care is also a purposes of CDSSs, transparency, explanations and interactive support for reasoning and decision making should be provided. A case study was conducted to evaluate the reasoning strategies supported by the solution. Results indicate that the strategies are complementary and serves different purposes, and can support users with different levels of experience and skills.

Future work includes further user studies of clinicians using the CDSS for the dementia domain exemplified in this research, with a special focus on the end user development. Also the development of person-tailored support based on patterns of reasoning and decision making is in the plan. Finally, as assessments in clinical practice generate patient data and situated decisions, new knowledge can be generated using data-driven methods. Future work thus includes exploring methods for generating and explaining this new knowledge by combining learning methods with symbolic methods.

Acknowledgements The authors declare that they have no conflict of interest.

References

1. Association, A.P.: Diagnostic and statistical manual of mental disorders, 4th edn (DSM-IV). Washington: American Psychological Association (1994)
2. Bardram, J.E.: A novel approach for creating activity-aware applications in a hospital environment. In: IFIP Conference on Human-Computer Interaction, pp. 731–744. Springer (2009)
3. Bardram, J.E., Hansen, T.R.: Why the plan doesn't hold - a study of situated planning, articulation and coordination work in a surgical ward. In: Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, pp. 331–340. ACM (2010)
4. Bassen, D., Nayak, S., Li, X.C., Sam, M., Sidhu, J., Nelson, M.F., Land, W.H.: Clinical decision support system (cdss) for the classification of atypical cells in pleural effusions. Procedia Computer Science **20**, 379–384 (2013)
5. Bemmel, J.H.v., Musen, M.A.: Handbook of medical informatics. Heidelberg: Springer-Verlag (1997)
6. Bennett, C.C., Hauser, K.: Artificial intelligence framework for simulating clinical decision-making: A markov decision process approach. Artificial Intelligence in Medicine **57**(1), 9–19 (2013)
7. Berners-Lee, T., Hendler, J., Lassila, O., et al.: The semantic web. Scientific American **284**(5), 34–43 (2001)
8. Bex, F., Prakken, H., Reed, C., Walton, D.: Towards a formal account of reasoning about evidence: argumentation schemes and generalisations. Artificial Intelligence and Law **11**(2), 125–165 (2003)
9. Bottrighi, A., Terenziani, P.: Meta-glare: A meta-system for defining your own computer interpretable guideline system - architecture and acquisition. Artificial Intelligence in Medicine **72**, 22–41 (2016)

10. Chesñevar, C., Modgil, S., Rahwan, I., Reed, C., Simari, G., South, M., Vreeswijk, G., Willmott, S., et al.: Towards an argument interchange format. The Knowledge Engineering Review **21**(4), 293–316 (2006)
11. Dang, J., Hedayati, A., Hampel, K., Toklu, C.: An ontological knowledge framework for adaptive medical workflow. Journal of Biomedical Informatics **41**(5), 829–836 (2008)
12. Eccher, C., Ferro, A., Seyfang, A., Rospocher, M., Miksch, S.: Modeling clinical protocols using semantic mediawiki: The case of the oncocure project. In: ECAI workshop on Knowledge Management for Health care Procedures (K4HelP), pp. 42–54. Springer (2008)
13. Fox, J.: Cognitive systems at the point of care: The credo program. Journal of Biomedical Informatics **68**, 83–95 (2017)
14. Fox, J., Johns, N., Rahmanzadeh, A.: Disseminating medical knowledge: the PROforma approach. Artificial Intelligence in Medicine **14**(1), 157–181 (1998)
15. Gebus, S., Leiviskä, K.: Knowledge acquisition for decision support systems on an electronic assembly line. Expert Systems with Applications **36**(1), 93–101 (2009)
16. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition **5**(2), 199–220 (1993)
17. Hassan, A.R., Subasi, A.: A decision support system for automated identification of sleep stages from single-channel EEG signals. Knowledge-Based Systems **128**, 115–124 (2017)
18. Jereb, E., Rajkovic, U., Rajkovic, V.: A hierarchical multi-attribute system approach to personnel selection. International Journal of Selection and Assessment **13**(3), 198–205 (2005)
19. Kaptelinin, V., Nardi, B.A.: Acting with technology: Activity theory and interaction design. MIT press (2006)
20. Kawamoto, K., Houlihan, C.A., Balas, E.A., Lobach, D.F.: Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. BMJ **330**, 765 (2005)
21. Kidd, A.L.: Knowledge acquisition for expert systems: A practical handbook. Plenum Press, New York (1987)
22. Lindgren, H.: Personalisation of internet-mediated activity support systems in the rehabilitation of older adults–a pilot study. proc. Personalisation for e-Health pp. 20–27 (2009)
23. Lindgren, H.: Integrating clinical decision support system development into a development process of clinical practice – experiences from dementia care. In: M. Peleg, N. Lavrac, C. Combi (eds.) Artificial Intelligence in Medicine (AIME 2011), *Lecture Notes in Computer Science*, vol. 6747, pp. 129–138. Springer (2011)
24. Lindgren, H.: Towards personalized decision support in the dementia domain based on clinical practice guidelines. User Modeling and User-Adapted Interaction **21**(4-5), 377–406 (2011)
25. Lindgren, H.: Knowledge artifacts as tools to communicate and develop knowledge in collaborative user-driven design. In: 25th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2012), pp. 210–215 (2012)
26. Lindgren, H., Baskar, J., Guerrero, E., Nieves, J.C., Nilsson, I., Yan, C.: Computer-supported assessment for tailoring assistive technology. In: Proceedings of the 6th International Conference on Digital Health Conference, DH'16, pp. 1–10. ACM, New York, NY, USA (2016)
27. Lindgren, H., Burström, L., Järvholm, B.: Developing ambient support technology for risk management in the mining industry. In: C. Ramos, P. Novais, C.E. Nihan, J.M. Corchado Rodriguez (eds.) Ambient Intelligence – Software and Applications, *Advances in Intelligent Systems and Computing*, vol. 291, pp. 161–169. Springer International Publishing (2014)
28. Lindgren, H., Lundin-Olsson, L., Pohl, P., Sandlund, M.: End users transforming experiences into formal information and process models for personalised health interventions. Studies In Health Technology And Informatics **205**, 378–382 (2014)
29. Lindgren, H., Winnberg, P.: Evaluation of a semantic web application for collaborative knowledge building in the dementia domain. In: M. Szomszor, P. Kostkova (eds.) Electronic Healthcare: Third International Conference, eHealth 2010, Casablanca, Morocco,

December 13-15, 2010, Revised Selected Papers, pp. 62–69. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
30. Lindgren, H., Winnberg, P.J., Winnberg, P.: Domain experts tailoring interaction to users - an evaluation study. In: P. Campos, T.C.N. Graham, J.A. Jorge, N.J. Nunes, P.A. Palanque, M. Winckler (eds.) INTERACT (3), *Lecture Notes in Computer Science*, vol. 6948, pp. 644–661. Springer (2011)
31. Lindgren, H., Winnberg, P.J., Yan, C.: Collaborative development of knowledge-based support systems: A case study. Studies In Health Technology And Informatics **180**, 1111–1113 (2012)
32. Miksch, S., Shahar, Y., Johnson, P.: Asbru: a task-specific, intention-based, and time-oriented language for representing skeletal plans. In: Proceedings of the 7th Workshop on Knowledge Engineering: Methods and Languages (KEML-97), pp. 9.1–9.20. The Open University, Milton Keynes, UK (1997)
33. Mol, A.: The body multiple: Ontology in medical practice. Duke University Press (2002)
34. Musen, M.A., Middleton, B., Greenes, R.A.: Clinical decision-support systems. In: E. Shortliffe, J. Cimino (eds.) Biomedical Informatics, pp. 643–674. Springer, London (2014)
35. Patel, V.L., Allen, V.G., Arocha, J.F., Shortliffe, E.H.: Representing clinical guidelines in GLIF: individual and collaborative expertise. Journal of the American Medical Informatics Association **5**(5), 467–483 (1998)
36. Patel, V.L., Groen, G.J.: Knowledge based solution strategies in medical reasoning. Cognitive Science **10**(1), 91–116 (1986)
37. Patel, V.L., Groen, G.J.: The general and specific nature of medical expertise: a critical look. In: A. Ericsson, J. Smith (eds.) Towards a general theory of expertise: prospects and limits, pp. 93–125. Cambridge University Press (1991)
38. Patel, V.L., Kaufman, D.R., Arocha, J.F.: Emerging paradigms of cognition in medical decision-making. Journal of Biomedical Informatics **35**(1), 52–75 (2002)
39. Patel, V.L., Shortliffe, E.H., Stefanelli, M., Szolovits, P., Berthold, M.R., Bellazzi, R., Abu-Hanna, A.: The coming of age of artificial intelligence in medicine. Artificial Intelligence in Medicine **46**(1), 5–17 (2009)
40. Peleg, M.: Computer-interpretable clinical guidelines: a methodological review. Journal of Biomedical Informatics **46**(4), 744–763 (2013)
41. Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R.A., Hall, R., Johnson, P.D., Jones, N., Kumar, A., et al.: Comparing computer-interpretable guideline models: a case-study approach. Journal of the American Medical Informatics Association **10**(1), 52–68 (2003)
42. Riaño, D., Real, F., López-Vallverdú, J.A., Campana, F., Ercolani, S., Mecocci, P., Annicchiarico, R., Caltagirone, C.: An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients. Journal of Biomedical Informatics **45**(3), 429–446 (2012)
43. Shalom, E., Shahar, Y., Taieb-Maimon, M., Bar, G., Martins, S.B., Young, O., Vaszar, L., Liel, Y., Yarkoni, A., Goldstein, M.K., et al.: Can physicians structure clinical guidelines? experiments with a mark-up-process methodology. In: ECAI workshop on Knowledge Management for Health Care Procedures (K4HelP), pp. 67–80. Springer (2008)
44. Valls, A., Gibert, K., Sánchez, D., Batet, M.: Using ontologies for structuring organizational knowledge in home care assistance. International Journal of Medical Informatics **79**(5), 370–387 (2010)
45. Walton, D., Reed, C.A.: Argumentation schemes and enthymemes. Synthese **145**(3), 339–370 (2005)
46. Wilson, M.: Knowledge engineering. In: Proceedings of SOFSEM, vol. 93, pp. 357–373. Citeseer (1993)