Technical Report / UMINF 18.08, ISSN 0348-0542

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 transparent and explainable to the end user. However, the development and maintenance of such solutions is resource demanding. 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 for allowing medical experts to model the medical knowledge and to structure the information that builds up the design of the user interface; II) A graphical user interface (GUI) and a GUI generator that keeps the interface is synchronised with updates of the knowledge base; III) An inference engine that utilises patient-specific data and extracts rules from the knowledge base for supporting reasoning and decision making. These modules can be reused when adapting to new situations. A CDSS for dementia diagnosis was developed and used as an example in the presentation 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 easy knowledge representation and management of medical knowledge, supports a continuing medical education and may improve clinical decision making and care provision.

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

Keywords Clinical Decision-support System · Semantic Web · Ontology · Data Management · Knowledge Acquisition · Knowledge Engineering · Dementia

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 [12]. 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 [17]. Studies have shown that a CDSS increases the effectiveness of prescribing medication without increasing cost [35]. 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 daily clinical practice, provided it applies artificial intelligence methods that are transparent and that can explain its inferences. Consequently, CDSSs are used for educational purposes and to disseminate consensus guidelines for care, developed by the medical community.

Despite the general consensus that CDSSs have the potential to improve the healthcare, there are some challenges preventing its broad use. 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 based on 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.

It takes a lot of time and efforts to develop a new CDSS for a particular disease [4,15]. In practice, the possibility to transfer the programming code and software between environments or diseases is limited. In the scenario 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 could 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 [36,19]. It is

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

therefore important to be able to tailor a system to different national treatment protocols, 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 person-tailored support for skill development in users of a CDSS is also a challenge, yet highly important, since the need for support is different depending on the professional's knowledge and clinical experience. Moreover, there is a continuing education seen in daily practice, partly where more experienced professionals guide less experienced. In an optimally digitalised healthcare, this daily evolvement of knowledge and skills could be supported by a system that provides person-adapted support.

The challenges addressed in this research are the following:

1. the knowledge acquisition and management bottleneck and in knowledge engineering of medical information [18,14];
2. the limitations in code reusability [9];
3. customisation to the routines at different care providing organisations, e.g., following different national treatment protocols [30], and
4. flexible support for the development of skills in an individual user (e.g., adaptation to different levels of expertise) [33,34].

The aim is to develop a web-based architecture that facilitates knowledge engineering and design of CDSSs conducted by medical domain experts, that facilitates personalisation, customisation and reuse of CDSS modules, and to demonstrate its applicability in the dementia domain.

In the next section, the related work is discussed. An introduction to the system architecture is provided in Section 3 and a case study of the CDSS for dementia is presented in Section 4. In Section 5 the results are discussed. The article ends with conclusions and future work.

2 Related Work

During the past 20 years, several task-network modelling languages have been developed to address the knowledge acquisition and management bottleneck in the development of knowledge-based clinical guidelines and treatment protocols [11,13,29,31,38] (for an overview, see [37]). The main purpose was to develop computer-interpretable clinical guidelines (CIG), which has some degrees of decision support and workflow support functions. These provide a modelling environment that is graphical, with decision modules and their dependencies visible, in order to allow non-programmers to be active in the modelling tasks. One example is PROforma [13]. However, user studies show that the modelling software is 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 [40]. 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].

ACKTUS (Activity-Centered Knowledge and Interaction Tailored to Users) [28] 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. ACKTUS is designed to address the following limitations of 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. Studies have been conducted to evaluate how medical experts without experience in knowledge engineering approach the tasks of knowledge modelling and designing the interaction using ACKTUS [21,24–27]. It was observed that the authorised experts were able to model the content and interaction with the system [24,26]. Moreover, they were able to revise and test the CDSS prototype in order to follow international medical knowledge [21, 27]. It was also observed that when experts model the knowledge, they become more careful in how to interpret the underlying clinical guideline and resolve ambiguities, and tend to create more strict rules compared to when mediating their knowledge through a knowledge engineer [20,25,27].

In some aspects, ACKTUS is similar to the generic symbolic decision theory including arguments, provided by the CREDO program that includes PROforma [12]. PROforma contains a simple version of argumentation, where strict rules and defeasible rules, or arguments, can be defined and executed. The number of defeasible arguments in favour or against a conclusion are counted to aggregate strength for a conclusion, assuming that all defeasible arguments have the same strength. 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 [20]. As a consequence, ACKTUS was developed to include a possibilistic logic framework for extending the management of uncertainty [41].

The systems developed using PROforma need to be linked to a separate and tailored application-specific GUI, as is the case also with ACKTUS-based applications. However, ACKTUS builds upon a core ontology that contains classes that build the structure and content of the graphical user interfaces of different applications, which was further developed as part of this research [28].

3 System Architecture

A CDSS contains typically the following three modules, as shown in the left part of Figure 1:

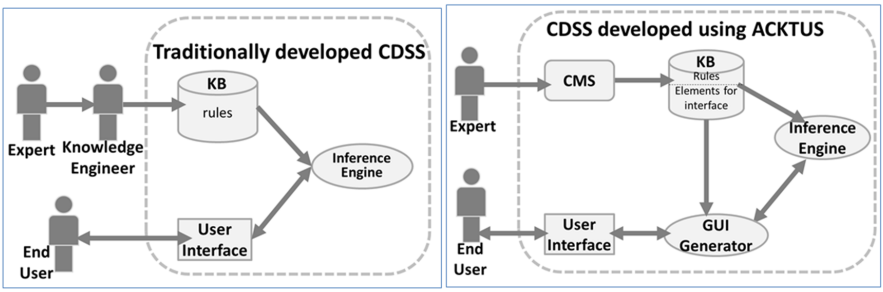

Fig. 1 Traditionally and ACKTUS-based CDSS architectures.

- A *knowledge base (KB)* that stores rules and information relevant for a particular medical domain,
- An *inference engine* that applies the knowledge stored in the KB to patient data retrieved from the user interface to deduct patient-specific recommendations,
- A *graphical user interface (GUI)*, which allows the system to display the results to the user as well as allow the user to input information into the system.

The knowledge engineer typically consults the medical domain experts when modelling the domain knowledge to be implemented in the system. The inference engine retrieves the knowledge from the KB, patient-specific information from the GUI, and generates a basis for decision making. The results of inferences are presented to the user through the GUI.

In this paper, the traditional architecture is extended to include two additional modules by further develop the ACKTUS architecture and core ontology. This for the purpose of facilitating maintenance of the CDSS and the broad adaptability across applications. The modified ACKTUS architecture is shown in the right part of Figure 1. The two additional modules compared to the traditional architectures are:

- A *content management system (CMS)* connected to the KB, which is built on Semantic Web technology to achieve modularity, reusability, customisation, 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 and keeps it synchronised with updates of the KB without software developer's intervention.

In a conventional CDSS, the KB is not synchronised with the user interface. However, in the ACKTUS-based KB, the core ontology includes not only the classes and their relationships from which rules can be extracted, but also the elements that the GUI generator uses for generating the user interface.

In the ACKTUS architecture, each part of the system is developed separately. The authorised medical domain experts (or simply, domain experts) model domain knowledge using the CMS, reducing the need of the knowledge engineer, and the knowledge is stored in the KB. The GUI generator fetches data from the KB and generates the user interface. In clinical practice, the end user (including both domain experts and other clinicians) fills in the patient symptoms via the CDSS user interface. The 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, together with advice regarding intervention.

Since the domain experts can directly model knowledge using the CMS, the update of the KB is facilitated.

In the following subsections, the different modules of the architecture are presented in more detail, with examples from the CDSS developed using ACKTUS DMSS-W (Dementia Diagnosis and Management Support System). The next section presents the knowledge base and CMS, Section 3.2 presents the user interface and GUI generator, and Section 3.3 presents the inference engine.

3.1 The Knowledge Base and Content Management System

The knowledge base is built using ontologies and managed through Semantic Web technologies. The *Semantic Web* 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[2] and OWL[3]. 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 standardised way of describing things and their relations. Sesame is a powerful Java framework for processing and querying RDF data. The query language for RDF is SPARQL. RDF, Sesame and SPARQL are applied for managing the ACKTUS ontologies.

The CMS is a web-based knowledge management platform used by authorised experts to represent medical knowledge and design the user interface and interaction. With the CMS, authorised experts with mark-up training can model domain knowledge with correct syntax and semantics. The knowledge can be understood, interpreted and utilised 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

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

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

a result of the authorised experts' modelling 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[4] and other medical terminologies, and extended with *scales* for evaluating the findings;
- Clinical knowledge: an extended version of the Argument Interchange Format (AIF) developed for exchanging arguments on the Web [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 domain 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 Figure 2, where the dashed frame indicates the KB. Basically, the instances of APs, IOs and schemes are created by the domain 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. In the following subsections details about the key classes are provided with examples from DMSS-W, the CDSS for dementia diagnosis and management.

3.1.1 Interaction Object and Assessment Protocol

The domain 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 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 Figure 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*.

[4] the International Classification of Function, Disability and health (ICF): http://www.who.int/classifications/icf/en/

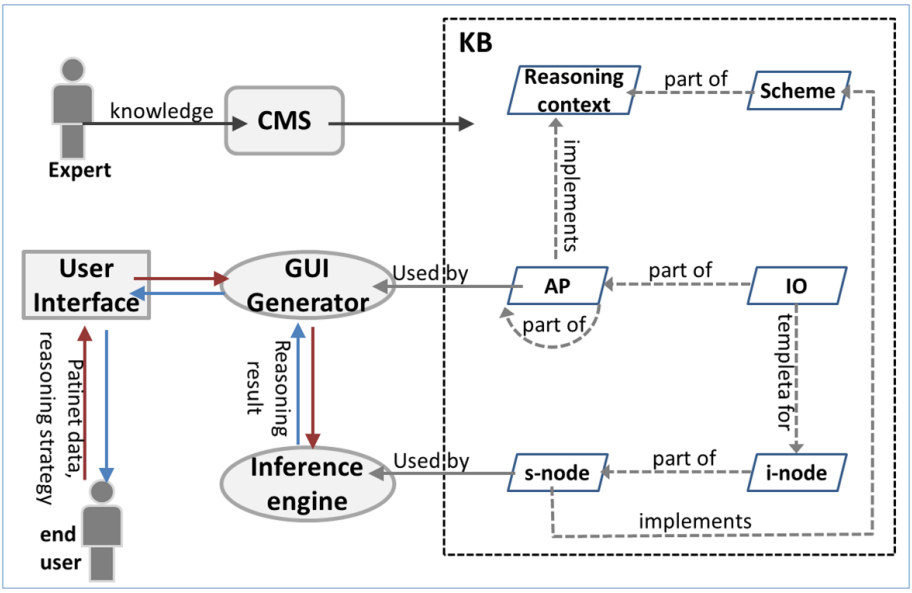

Fig. 2 Workflow of knowledge engineering and automated process that result in a CDSS that can be immediately verified by the expert and used by the user. AP is assessment protocol and IO is interaction object (see Table 1).

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**;
- 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

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

Class name	Property name	Type and example of implementation
interaction object (IO)	has-term	text: "Judgement" (en)
	has-term	text: "omdöme" (sv)
	has-info	text: "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	scale ID: normal/unknown/affected
	has-severity-scale	scale ID: Not specified/Mild/Significant
	has-concept	concept ID for e.g., "Judgement"
	has-organization	ID for specific organization/individual
assessment protocol (AP)	has-name	text: "Status" (en)
	has-description	text: "Current state of the patient" (en)
	has-ordered-list	list with the included APs and IOs
	has-organization	specific organization/individual
i-node	is-related-to	IO ID for e.g., "episodic memory"
	has-text-value	text: "Significant episodic memory deficit is present"
	has-reliability-value	value ID for e.g., "affected"
	has-severity-value	value ID for e.g., "significant"
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."
	implements-scheme	scheme ID: e.g., Dementia-DSM-IV
	has-status	value ID: e.g., "validated"
scheme	has-premise-description	text: "Significant episodic memory deficit is present"
	has-conclusion-description	text: "Alzheimer's Disease is present"
	has-knowledge-source	knowledge source: DSM-IV
reasoning context	is-activated-by-cq	ID of the critical question (CQ) "Which type of cognitive disorder is present?"
	context-includes-scheme	scheme ID: Dementia_DSM-IV_Scheme
	has-prev-step	reasoning context ID of "Step: Is there a cognitive disorder?"

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

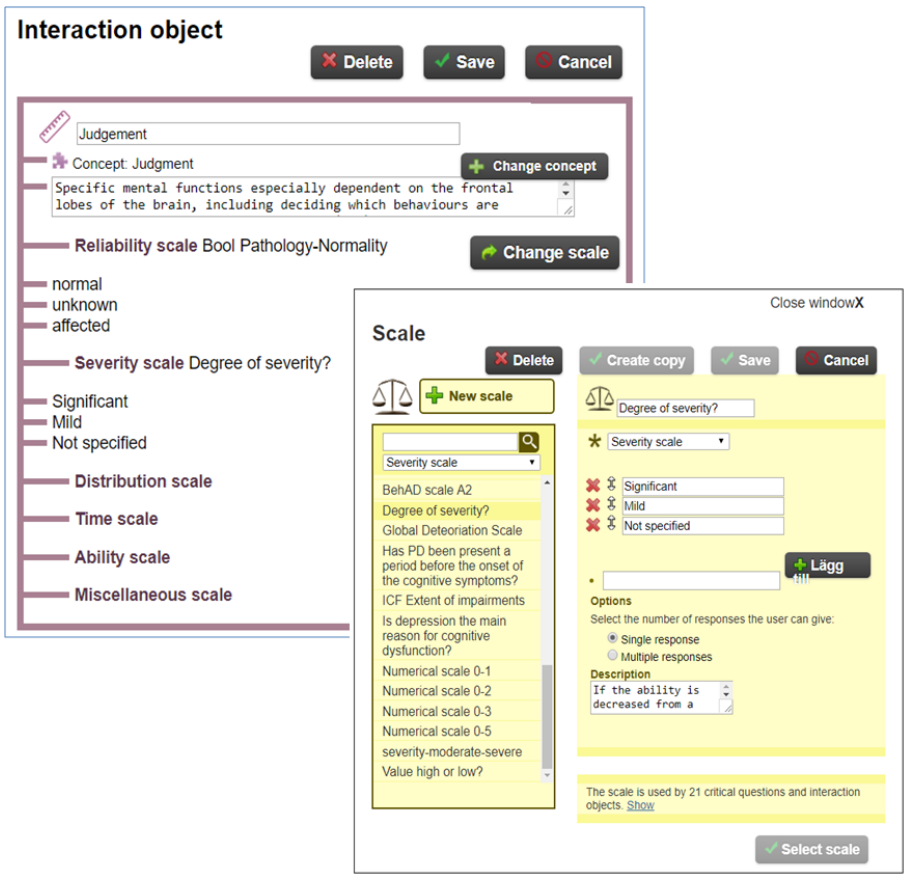

Fig. 3 The snapshot of the IO "Judgement" under edit mode in the CMS. The lower right panel appears when the user clicks "Change scale" in the upper left panel.

3.1.2 Scheme and Knowledge Source

The *schemes*, denoted *argument schemes* in research literature [8,39], 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. 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 categorised 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 provided by a domain expert is ranked 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 authorised 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. Figure 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.

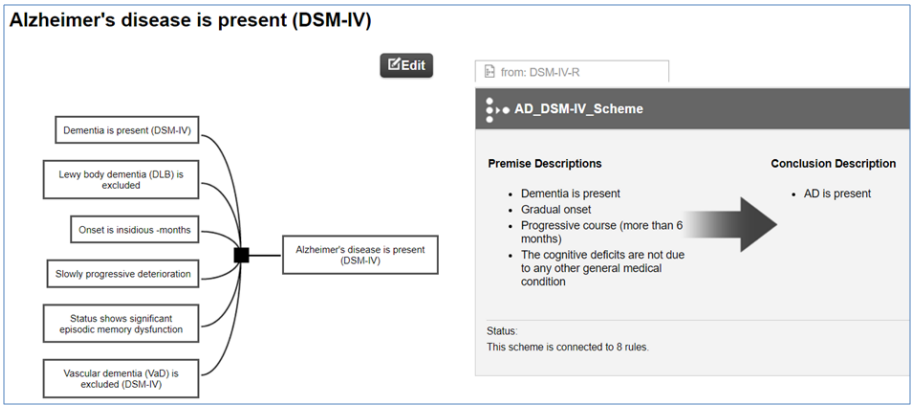

Fig. 4 Screenshot of an s-node and the related scheme extracted from the CMS.

Thus, it is valuable to define *reasoning contexts* as a selection of guidelines to be applied in different steps in a diagnostic reasoning process. 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 Figure 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

[5] https://jquery.com/

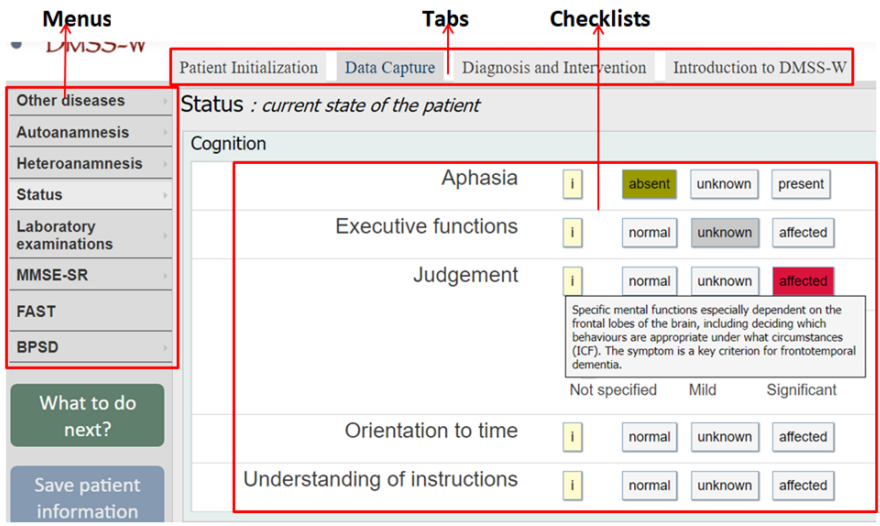

Fig. 5 Example of the *data capture* tab in DMSS-W demonstrating the function of the GUI generator.

Figure 5). However, the *reasoning-context-based AP* is special and is used for generating a button (e.g., *What to do next?* in Figure 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 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 Figure 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. Figure 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.

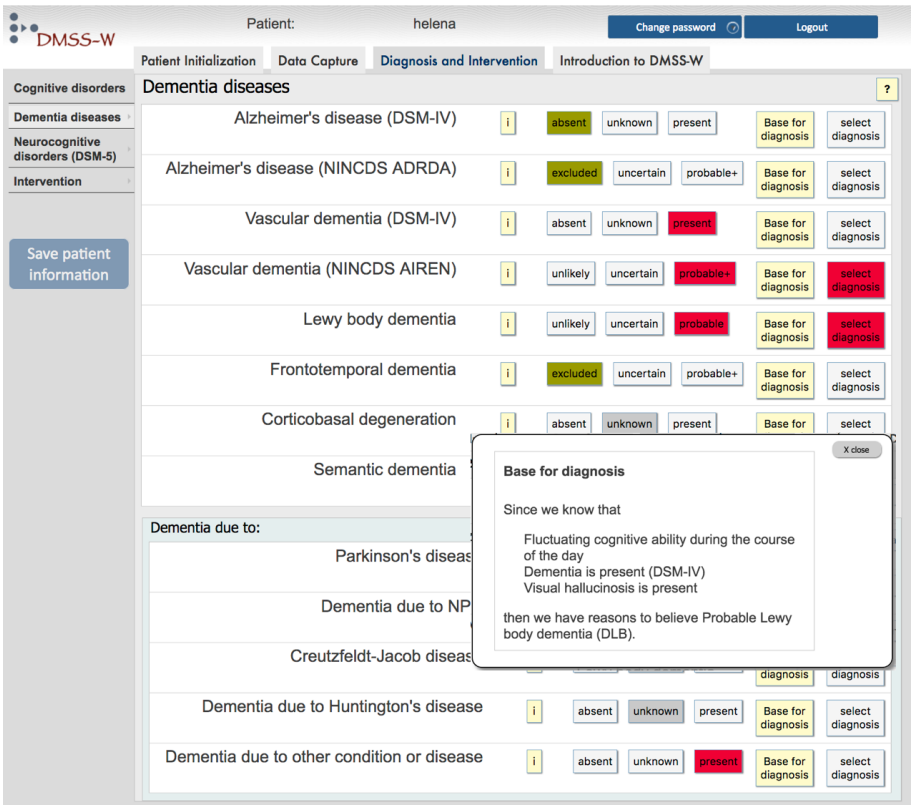

Fig. 6 Overview of the diagnostic results.

3.2.2 Designing Support for Reasoning Strategies

The GUI generator supports two complementary reasoning processes, following to how medical professionals conduct clinical reasoning and decision making [32]. When a clinician applies the *forward-chaining* diagnostic reasoning 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 (Figure 6). The results are presented to the user through the *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

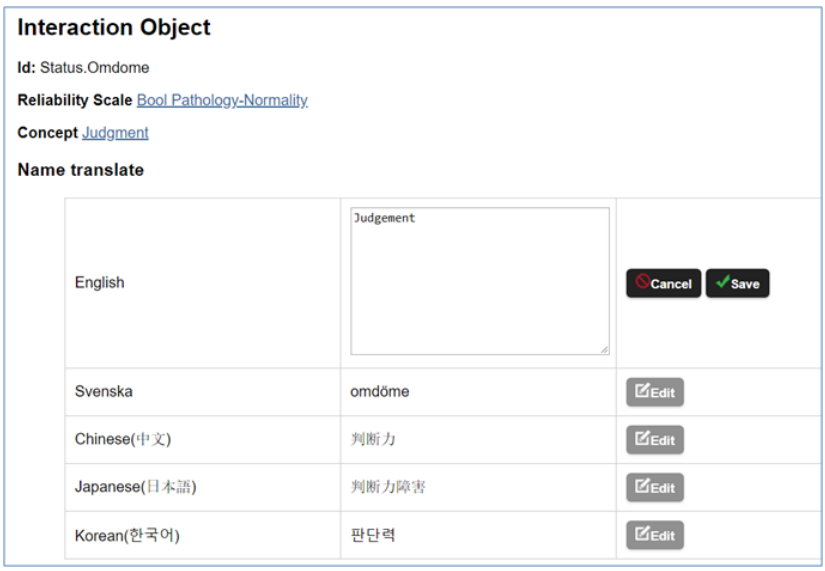

Fig. 7 Snapshot of the CMS interface in edit mode for editing an IO's English name.

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.

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 (Figure 8). When the user activates this functionality, the system generates a small checklist corresponding to a *reasoning context* with a subset of IOs for the user to fill in. 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 reasoning context, or 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

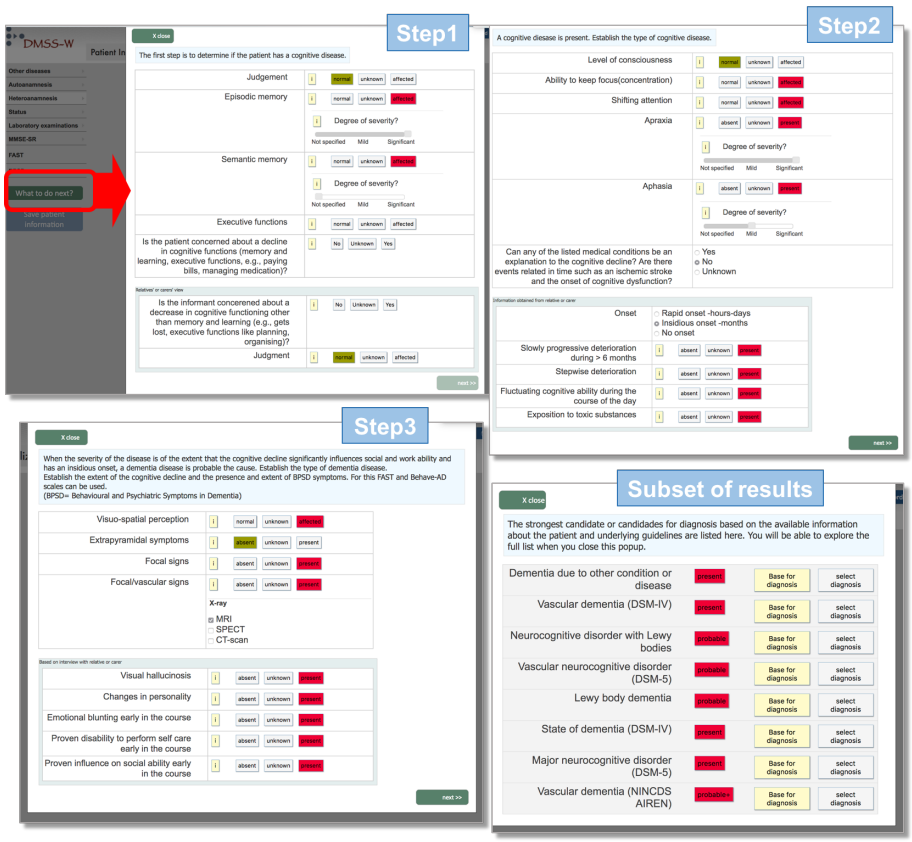

Fig. 8 Overview of the three steps of assessment following the reasoning context-based guide.

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. The IOs together with the inputted scale values are regarded as *defeasible 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 dur-

ing 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 [41]. 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 (Figure 6), where the user can explore the arguments in favour 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 finalises the decision by clicking the button *select diagnosis*.

3.4 Summary

To summarise, the traditional CDSS architecture was transformed into an architecture based on ACKTUS, where additional modules and content were developed. The core ontology was extended with more information, so that it includes not only the classes containing information extracted for reasoning, but also the necessary elements that are needed to construct the CDSS user interface. A graphical user interface (GUI) generator was developed. Domain experts model domain knowledge and the structure of the user interface through the ACKTUS CMS and the results are stored in the KB. The GUI generator fetches the information from the KB and automatically generates the CDSS interface whenever the user logs in, so that the interface is synchronised with alterations in the KB. The modules can be reused when developing additional CDSS, with minimal time required by an engineer to deploy a functioning prototype. As a consequence, knowledge engineering, evaluation and maintenance of the CDSS is facilitated, since the code of CDSS does not need to change and the website does not need to be redeployed in the process. These tasks can be done by a medical domain expert, which also facilitates the development process.

Two complementary reasoning strategies are supported, which can be used as guide to the domain expert to model the knowledge content for each. This provides users who are differently experienced in the particular medical domain the possibility to find support tailored to their level of expertise. This was done in the development of the CDSS DMSS-W.

4 Case Study

A case study was 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 was to distinguish between obstacles that are due to limitations in dementia knowledge, and obstacles due to the interaction design of the CDSS. We chose 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 ranged from being novice (two participants), somewhat knowledgeable (one participant) and knowledgeable (one participant). We were particularly interested in how the two reasoning methods are used, and how the participants interpreted the severity values.

Results showed 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 was 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 had more experience, but were not yet experts were 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 could 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, was also able to complete assessments. However, with less confidence in her own assessments, she was 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 was brought up by the two more knowledgeable participants, and the person with most experience found 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 support, DMSS can also support a continued medical education, with support tailored to individual user's expertise level.

5 Discussion

There are tools for developing expert systems, available both as commercial software (e.g., *Exsys*[6]) and freeware (e.g., *DEXi* [16]). 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 [23], and in the rehabilitation domain [22] 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

Since disseminating new evidence-based medical knowledge and providing a continued medical education in order to improve care are main purposes of CDSSs, transparency, explanations and interactive support for reasoning and decision making should be provided. A knowledge management and end-user development solution is presented in this article for reducing the resources required for knowledge acquisition, knowledge management and customisation of

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

CDSSs. The solution 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 graphical user interface (GUI) and a GUI generator that automatically synchronises the GUI with updates of the KB; and III) an inference engine that can be reused by other CDSSs. The CMS and GUI generator are two additional modules in the ACKTUS-based architecture compared to conventional CDSS architectures.

The modules can be reused when developing additional CDSS, with minimal time required by an engineer to deploy a functioning prototype. As a consequence, knowledge engineering, evaluation and maintenance of the CDSS is facilitated, since the code of CDSS does not need to change and the website does not need to be redeployed in the process. These tasks can be done by a medical domain expert, which also facilitates the development process.

Two complementary reasoning strategies are supported, which can be used as guide to the domain expert to model the knowledge content for each. This provides users who are differently experienced in the particular medical domain the possibility to find support tailored to their level of expertise. This was done in the development of the CDSS DMSS-W. 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. Future work includes also the development of person-tailored support based on patterns of reasoning and decision making. 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.

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. 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)
12. Fox, J.: Cognitive systems at the point of care: The credo program. Journal of Biomedical Informatics **68**, 83–95 (2017)
13. Fox, J., Johns, N., Rahmanzadeh, A.: Disseminating medical knowledge: the PROforma approach. Artificial Intelligence in Medicine **14**(1), 157–181 (1998)
14. Gebus, S., Leiviskä, K.: Knowledge acquisition for decision support systems on an electronic assembly line. Expert Systems with Applications **36**(1), 93–101 (2009)
15. 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)
16. 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)
17. 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)
18. Kidd, A.L.: Knowledge acquisition for expert systems: A practical handbook. Plenum Press, New York (1987)
19. 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)
20. 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)
21. 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)
22. 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)
23. 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)
24. 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)

25. 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)
26. 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)
27. 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)
28. Lindgren, H., Yan, C.: ACKTUS–A platform for developing personalized support systems in the health domain. In: Proceedings of the 5th International Conference on Digital Health 2015, pp. 135–142. ACM (2015)
29. 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)
30. 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)
31. 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)
32. Patel, V.L., Groen, G.J.: Knowledge based solution strategies in medical reasoning. Cognitive Science **10**(1), 91–116 (1986)
33. 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)
34. 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)
35. 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)
36. Peleg, M.: Computer-interpretable clinical guidelines: a methodological review. Journal of Biomedical Informatics **46**(4), 744–763 (2013)
37. 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)
38. 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)
39. Walton, D., Reed, C.A.: Argumentation schemes and enthymemes. Synthese **145**(3), 339–370 (2005)
40. Wilson, M.: Knowledge engineering. In: Proceedings of SOFSEM, vol. 93, pp. 357–373. Citeseer (1993)
41. Yan, C., Lindgren, H., Nieves, J.C.: A dialogue-based approach for dealing with uncertain and conflicting information in medical diagnosis. To appear