



City Research Online

City, University of London Institutional Repository

Citation: Wilk, S., Michalowski, W., O'Sullivan, D., Farion, K., Sayyad-Shirabad, J., Kuziemy, C. and Kukawka, B. (2013). A Task-based Support Architecture for Developing Point-of-care Clinical Decision Support Systems for the Emergency Department. *Methods of Information in Medicine*, 52(1), pp. 18-32. doi: 10.3414/ME11-01-0099

This is the accepted version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: <http://openaccess.city.ac.uk/8156/>

Link to published version: <http://dx.doi.org/10.3414/ME11-01-0099>

Copyright and reuse: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

City Research Online:

<http://openaccess.city.ac.uk/>

publications@city.ac.uk

A Task-based Support Architecture for Developing Point-of-Care Clinical Decision Support Systems for the Emergency Department

Szymon Wilk, Ph.D.[†]

*Institute of Computing Science, Poznan University of Technology, Poland
Telfer School of Management, University of Ottawa, Canada*

Wojtek Michalowski, Ph.D.

Telfer School of Management, University of Ottawa, Canada

Dympna O'Sullivan, Ph.D.

*School of Engineering and Applied Science, Aston University, United Kingdom
Telfer School of Management, University of Ottawa, Canada*

Ken Farion, M.D.

Departments of Pediatrics and Emergency Medicine, University of Ottawa, Canada

Jelber Sayyad Shirabad, Ph.D.

School of Electrical Engineering and Computer Science, University of Ottawa, Canada

Craig Kuziemy, Ph.D.

Telfer School of Management, University of Ottawa, Canada

Bartosz Kukawka, M.Sc.

Institute of Computing Science, Poznan University of Technology, Poland

Keywords: decision support system, clinical; point-of-care; system architecture; agent-oriented paradigm; ontology-driven design

[†] Corresponding author:

Szymon Wilk

Institute of Computing Science, Poznan University of Technology

Piotrowo 2, 60-965 Poznan, Poland

e-mail: szymon.wilk@cs.put.poznan.pl, tel.: +48 61 665 29 30, fax: +48 61 877 15 25

SUMMARY

Objectives: The purpose of this study was to create a task-based support architecture for developing clinical decision support systems (CDSSs) that assist physicians in making decisions at the point-of-care in the emergency department (ED). The backbone of the proposed architecture was established by a task-based emergency workflow model for a patient-physician encounter.

Methods: The architecture was designed according to an agent-oriented paradigm. Specifically, we used the O-MaSE (Organization-based Multi-agent System Engineering) method that allows for iterative translation of functional requirements into architectural components (e.g., agents). The agent-oriented paradigm was extended with ontology-driven design to implement ontological models representing knowledge required by specific agents to operate.

Results: The task-based architecture allows for the creation of a CDSS that is aligned with the task-based emergency workflow model. It facilitates decoupling of executable components (agents) from embedded domain knowledge (ontological models), thus supporting their interoperability, sharing, and reuse. The generic architecture was implemented as a pilot system, MET3-AE – a CDSS to help with the management of pediatric asthma exacerbation in the ED. The system was evaluated in a hospital ED.

Conclusions: The architecture allows for the creation of a CDSS that integrates support for all tasks from the task-based emergency workflow model, and interacts with hospital information systems. Proposed architecture also allows for reusing and sharing system components and knowledge across disease-specific CDSSs.

1. Introduction

Clinical decision support systems (CDSSs), understood as systems that apply knowledge to patient data in order to generate patient-specific advice [1], can enhance clinical performance, increase efficiency of care delivery and reduce healthcare costs (see e.g., [2-4]). Despite these benefits, CDSSs deployment and adoption in clinical practice has been limited [5]. This has triggered research into the desired characteristics such systems should exhibit. Kawamoto et al. [6] identified availability at the point-of-care and fit with clinical workflow as crucial factors in successful deployments of CDSSs. These factors have been included in the “10 commandments of effective clinical decision support” [7], a list that also refers to the system’s speed and the simplicity of interventions (support functions), the ability to discard or override the system’s recommendations, limiting the amount of information users need to provide, and easy maintenance of the decision models embedded in the system. The “commandments” have been complemented with “10 grand clinical decision support challenges” [8] that should be addressed to achieve the practical benefits of CDSSs. Among others, these challenges include disseminating best practices in CDSS design, creating architectures for sharing executable modules, and creating repositories of decision support knowledge modules.

Fit with the clinical workflow appears to be a prevailing theme related to requirements and challenges for CDSS deployment. Sittig et al. [8] list desired characteristics of a CDSS satisfying this requirement. Such a CDSS “should unobtrusively, but effectively, remind clinicians of things they have truly overlooked and support corrections, or better yet, put key pieces of data and knowledge seamlessly into the context of the workflow or clinical decision-making process, so the right decisions are made in the first place”. Moreover, Fieschi et al. [9] state that “perspective workflow and care processes need to be emphasized and technology must become a second-

ary factor” and they underline the need for a workflow model combining manual and automated activities to drive the design and implementation of any CDSS. They also maintain that healthcare providers must stay at the center of the decision-making loop, while the system has to facilitate access to patient data, provide decision support whenever it is appropriate and expand decision models with access to clinical evidence.

In this paper we focus on the architectures for developing CDSSs for use in the Emergency Department (ED) of a hospital and use the term *task-based clinical decision support (task-based support* for short) to indicate computerized clinical decision support, which includes all observation and reasoning tasks from the task-based emergency workflow model. We formalize essential requirements for task-based support and propose an architecture for developing disease-specific (here we interpret a disease as a presenting complaint) CDSSs for use at the point-of-care in the ED – we refer to this architecture as the *task-based support architecture*. Given the distributed nature of task-based support and recent research on complex CDSS architectures [10], we chose to follow the principles of the agent-oriented paradigm [11-13] to design the task-based support architecture. Moreover, to ensure the ability to share clinical knowledge embedded in CDSSs developed according to the task-based support architecture, we extended the agent-oriented paradigm with ontology-driven design [14].

In our earlier research we developed an ontology-driven architecture for a mobile CDSS that supported just the presentation stage from the task-based emergency workflow model [14,15]. The objective of the research reported here was to develop a *generic task-based support architecture using an agent-oriented paradigm and ontology-driven design* that allows for the creation of CDSSs supporting multiple tasks from the task-based emergency workflow model. In the paper we describe the process of developing such architecture and present its pilot implemen-

tation (the MET3-AE system) for managing pediatric asthma exacerbations. We consider the generic character of the task-based support architecture to be our main contribution to the body of CDSS research.

2. Related works

In [16] Wright and Sittig provided the following taxonomy of CDSS architectures:

1. Stand-alone, where decision support is implemented as a separate and self-contained system,
2. Integrated, where decision support is embedded into an existing HIS (usually an electronic patient record),
3. Service model, where decision support is realized as a set of independent entities that provide services through a well-defined interface.

Given the above taxonomy, the architecture presented in the paper follows a hybrid approach combining the stand-alone and service model architectures by consisting of entities providing and requesting services, as well as entities interacting with the user and with HISs.

We followed an agent-oriented paradigm as a design methodology. Traditionally this paradigm has been considered within artificial intelligence and associated with notions of autonomy, reactivity, proactiveness and social abilities [12]. However, recently it has been used as an approach to analyzing and designing complex distributed software systems consisting of multiple problem-solving entities that interact with each other in a flexible and unpredictable (established during run-time) manner [13]. Isern et al. in [10] state that “agents offer a natural way of tackling inherently distributed problems with heterogeneous sources, by cooperating and coordinating their activities, and also acting pro-actively to perform tasks that may be beneficial for the user”. They further list the advantages of using the agent-based paradigm for CDSS, including modu-

larity and flexibility (the ability to dynamically reconfigure at runtime) of developed systems, efficiency (multiple agents running in parallel), decentralization and better reliability (no single point of failure), and improved security (specialized agents to ensure privacy and security of processed information). All these factors are crucial for the task-based support architecture.

From a software engineering perspective the agent-oriented paradigm is closely related to the component-oriented approach [17]. They both emphasize encapsulation of state and behavior, and interactions between entities. They also allow for assembling new systems from existing entities, which significantly enhances software reuse. There are several software engineering methodologies for designing systems according to the agent-based paradigm (see [18] for a review) – with the resulting systems referred to as multi-agent systems. These methodologies assume a system as being an organization of agents that have a common goal, where agents play specific roles in order to achieve this goal. Such an assumption is consistent with the recommendations in [9] on modeling clinical processes.

Defining workflows and managing their execution has been traditionally associated with workflow management systems [19]. These systems use workflow engines to process formally represented workflows by interacting with their participants and invoking other tools and applications. Workflow management systems have been considered in business settings, but recently they have been applied in healthcare [20] where they mostly deal with scheduling and coordinating activities from a workflow. Decision making capabilities are usually limited to branching based on logical expressions – these shortcomings can be addressed by additionally using rule engines [21].

The idea of applying the agent-oriented paradigm to a workflow model was discussed in [22] and it has been used in several clinical systems for executing intervention plans and clinical

practice guidelines. The HeCaSe2 system [23] facilitates coordination and execution of services across healthcare organizations, while the K4Care system [24] coordinates a team of healthcare professionals providing home care to elderly patients. Both systems use ontological models to represent domain knowledge, and while focusing on coordination, they provide limited decision support (decision models are limited to logical expressions). Another example is the Medical Information Agents system [25] that helps with guideline-based therapy planning. It schedules therapy, checks its conformance with clinical practice guideline and retrieves clinical evidence relevant for the currently applied guideline.

3. Methods

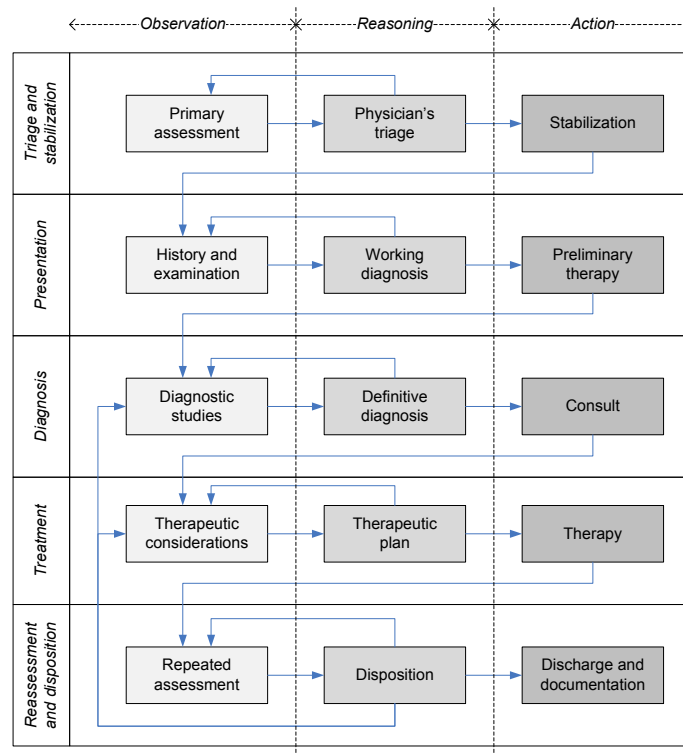
In this section we describe the methods we used to establish the task-based support architecture. We start with the model of a task-based emergency workflow that formed the backbone of the architecture. Then we discuss in details requirements defined for the proposed architecture. Finally, we describe software engineering methods we have used.

3.1. Task-based emergency workflow

In the research described in the paper we are concerned with CDSSs to be used at the point-of-care during the patient-physician encounter in the ED of an acute care hospital, where clinicians have to provide care to many patients at a time [26]. Since there are no agreed upon standards for clinical workflows [5], including the emergency workflow, we have created a task-based emergency workflow model that is at the core of the proposed architecture. It is presented in Figure 1, and specific tasks from the model are further described in Table I. The model comprises five stages (represented as horizontal layers in Figure 1) that are normally considered in

clinical practice [26,27], and each stage involves three types of tasks corresponding to collecting data (*observation*), making a decision (*reasoning*) and implementing the decision (*action*) [28].

Fig. 1. A task-based emergency workflow model



Following Fieschi et al. [9], we focus on the observation and reasoning tasks that are associated with data collection (or access) and decision making, and that benefit most from support provided by a CDSS. Action tasks implement earlier decisions by healthcare providers and most often they are completed manually or with the help of existing HISs, such as admission-discharge-transfer system or computerized physician order entry systems. In our research we do not consider the first stage (triage and stabilization) in the proposed workflow model as in an increasing number of settings it is handled by ED information systems [29], and we do not want to duplicate already available functionality.

Table I. Stages and tasks in the task-based emergency workflow model

Stage	Task	Type	Description
Triage and stabilization	Primary assessment	Observation	Collecting initial patient data
	Physician's triage	Reasoning	Triaging the patient
	Stabilization	Action	Applying the first therapy to stabilize vital symptoms and signs
Presentation	History and examination	Observation	Acquiring a patient's history, doing a physical examination and considering modifying factors (e.g., age, social status)
	Working diagnosis	Reasoning	Developing a working diagnosis based on available information
	Preliminary therapy	Action	Relieving symptoms pending the definitive diagnosis and formal treatment (only if necessary or mandated by the working diagnosis)
Diagnosis	Diagnostic studies	Observation	Ordering of investigations (e.g., X-ray, ECG, laboratory tests) specific for the working diagnosis and collecting their results
	Definitive diagnosis	Reasoning	Formulating a definitive diagnosis where a consult decision is considered as one of possible diagnostic outcomes
	Consult	Action	Consulting a specialist (only if necessary or mandated by the patient's condition and the definitive diagnosis)
Treatment	Therapeutic considerations	Observation	Collecting additional information necessary to construct a proper therapeutic plan (drug interactions, adverse effects, efficacy data, etc.)
	Therapeutic plan	Reasoning	Developing the therapeutic plan that may involve (non-pharmacologic) procedures and therapies
	Therapy	Action	Treating the disease according to the devised plan
Reassessment and disposition	Repeated assessment	Observation	Collecting necessary information to evaluate whether a therapy was effective or not and whether the patient can be discharged
	Disposition	Reasoning	Deciding whether to revise the diagnosis, continue with the therapeutic plan, modify it, or proceed to discharge
	Discharge and documentation	Action	Discharging the patient and completing the discharge summary, including patient education

3.2. *Requirements for the task-based support architecture*

The requirements were established by the authors from research on the desired features of CDSSs [6-8]. They were reviewed by a panel of five expert ED physicians (EPs) (with at least 10 years of clinical experience as postulated in [30]), and presented a consensus decision describing a final set of requirements. The resulting list includes:

1. Provision of support for all observation and reasoning tasks from the task-based emergency workflow model,
2. Access to information available in a HIS, especially in an electronic patient record,
3. Availability at the point-of-care during the entire patient-physician encounter,
4. Ability to share and reuse architectural components and domain knowledge.

These requirements do not cover issues related to security and privacy. Such issues are outside the scope of this paper, however in the MET3-AE system (described in Section 5) we used policy-based encryption [31] to maintain appropriate levels of security and privacy. Each requirement is detailed in the following sections.

3.2.1. Provision of support for all observation and reasoning tasks

In order to meet this requirement, the task-based support architecture should include a set of functions with associated models that help complete observation and reasoning tasks from the workflow. We refer to these models as functional models because they represent disease-specific knowledge that is needed by specific functions (see Table II).

Table II. Supported tasks, functions and functional models

Supported task	Function	Functional model
History and examination	Structured collection of patient data	Data model
Repeated assessment		User interface model
Diagnostic studies	Ordering of investigations and retrieving their results	Data model
Working diagnosis	Suggestion of possible diagnoses	Diagnostic model
Definitive diagnosis		
Therapeutic considerations	Provision of relevant clinical evidence	Evidence model
Therapeutic plan	Suggestion of therapeutic plans	Therapeutic model
Disposition	Suggestion of possible disposition	Disposition model

Supporting the tasks of *history and examination* and *repeated assessment* implies prompting the EP to consider clinical attributes specific to a given disease and providing a means for structured data entry (this alone may result in improved clinical outcomes [32]). This function requires a data model defining the relevant clinical attributes and a user interface model defining the structured data entry components.

Supporting the *diagnostic studies* task implies transmitting orders to a computerized physician order entry system and capturing the results from a laboratory information system. This function uses the data model to identify investigations (mostly laboratory tests) that need to be considered when managing a specific disease.

Supporting the *working diagnosis* and *definitive diagnosis* tasks translates into the ability to derive a diagnosis. To achieve this, the task-based support architecture must allow implementing non-deterministic diagnostic models, usually discovered from data, that codify diagnostic knowledge in appropriate formats [33], and have a mechanism that applies these models to available patient data (in the form of a solver).

Supporting the *therapeutic considerations* task implies retrieving clinical evidence from available libraries and presenting it to the EP at the point-of-care [34]. Presented evidence should

be relevant not only to the disease, but also to the context of the patient-physician encounter.

This function requires the evidence model that specifies how to map the context of an encounter to index terms used for retrieval of documents from an evidence library.

Supporting the task of *therapeutic plan* translates into establishing appropriate therapies for a disease and it requires knowledge about best practices. This necessitates the therapeutic model that codifies therapeutic knowledge, possibly coming from clinical practice guideline and adjusted for local practice patterns.

Finally, the *disposition* task implies deciding whether to continue provision of care or to discharge a patient. Supporting this function requires a disposition model that codifies knowledge necessary for making a disposition decision.

3.2.2. Access to information available in HIS

A vast amount of clinical data is collected before a patient-physician encounter (e.g., during nursing triage and assessments), or outside the point-of-care (e.g., in laboratories), and such data is usually available in electronic form. The task-based support architecture needs to support access to this information by providing a means for interacting with other HISs, especially with an electronic patient record. Meeting this requirement is considered one of the most important success factors for CDSS implementation [35].

Exchanging data between various HISs requires syntactic and semantic integration (exchanged information must have the same structure and meaning) [36]. The problem of data integration (particularly in terms of semantics) is a separate research field that has been intensively explored [37]. There have been attempts to introduce standards (endorsed by the ANSI Healthcare Information Technology Standards Panel) [5] aimed at alleviating the integration is-

sue in practice. While we acknowledge the importance of an integration problem, in our research we rely on available solutions for the data integration.

3.2.3. Availability at the point-of-care

This important requirement is rarely met [38]. We interpret it as the pervasive availability of a system's functionality, meaning that the task-based support architecture should allow for the development of CDSSs that run on computing devices that are easily accessible and usable at the point-of-care. This involves the variety of mobile platforms including smartphones, tablets and ultrabooks (thin and lightweight notebooks). However, to fully satisfy the postulate of "multi-device architecture" [39] we also consider desktop computers – although not "mobile" they may be a convenient tool for reviewing and completing patient data outside the point-of-care [39].

The multi-device architecture requires customization of the user interface for a specific platform (data entry modalities and display size). This is especially important for the data collection task because a cumbersome interface may render the system worthless [40]. We addressed this customization in our earlier research by proposing platform-specific user interface models [14] and we apply these models here.

3.2.4. Ability to share and reuse architectural components and domain knowledge

This requirement does not only follow from research on CDSSs [8], but its importance is also supported by our experience with constructing flexible and easy to maintain mobile CDSSs [14]. In order to satisfy this requirement, the task-based support architecture needs to enable the sharing of functional models (see Section 3.2.1) among several CDSSs, as well as reusing their parts when defining new models. The architecture also needs to allow sharing of components that implement the functions listed in Table II.

3.3. *Design of the task-based support architecture*

To create the task-based support architecture we used the O-MaSE (Organization-based Multi-agent System Engineering) [41] method. It provides a systematic and formalized way of designing multi-agent systems well rooted in software engineering techniques. O-MaSE does not impose any requirements on the agents (they can be autonomous and proactive, or reactive), and it allows designing agents of varying complexity and decisional autonomy.

Building on our earlier research [14] we expanded the O-MaSE method with ontology-driven design. Ontology-driven design is an active research area in CDSS design [42,43] and it advocates describing processed information and knowledge, and essential architectural components using ontological models derived from a domain ontology (an ontological model is comprised of instances of classes from the domain ontology – often it is referred to as a knowledge base [44]).

Designing the architecture involves construction of several models associated with its important aspects (e.g., components and communication). Use of the expanded O-MaSE method resulted in the following models (the tasks of creating specific models were divided among the authors, who performed them in pairs or alone):

- A goal model translating the requirements into specific goals that need to be satisfied by the architecture,
- A domain ontology defining concepts associated with the processed information and knowledge, and with essential components of the architecture,
- An agent model identifying agents required to satisfy the goal set for the architecture in the goal model and capturing interactions among these agents,

- A set of protocol models describing details of interactions between agents in terms of exchanged messages,
- A set of plan models describing algorithms used by the agents to achieve specific goals from the goal model.

4. Results

The main result of our research is a task-based support architecture (represented by a set of O-MaSE models) for creating different CDSSs that is described in detailed here. Specifically we focus on the goal model, domain ontology and agent model. The protocol models rely on the request-response pattern, and are relatively simple, and therefore they are not discussed. Finally, the plan models are complex and are described elsewhere [45-47].

4.1. Goal model

The goal model for the task-based support architecture is presented in Figure 2. It translates the requirements described in Section 3.2 into a tree-like structure representing goals and the relationships between them.

The top goal – *provide task-based support* – is decomposed into four subgoals. The first two subgoals – *provide assistance for encounter* and *provide suggestions and evidence* – come from the requirement of providing support for all observation and reasoning tasks from the task-based emergency workflow model (see Section 3.2.1). *Provide assistance for encounter* represents the overall, workflow-oriented perspective and it is further decomposed into eight leaf goals corresponding directly to providing assistance with specific observation and reasoning tasks. This ensures the EP is given appropriate support when completing these tasks. *Provide suggestions and evidence* is associated with a decision support-oriented perspective and it is de-

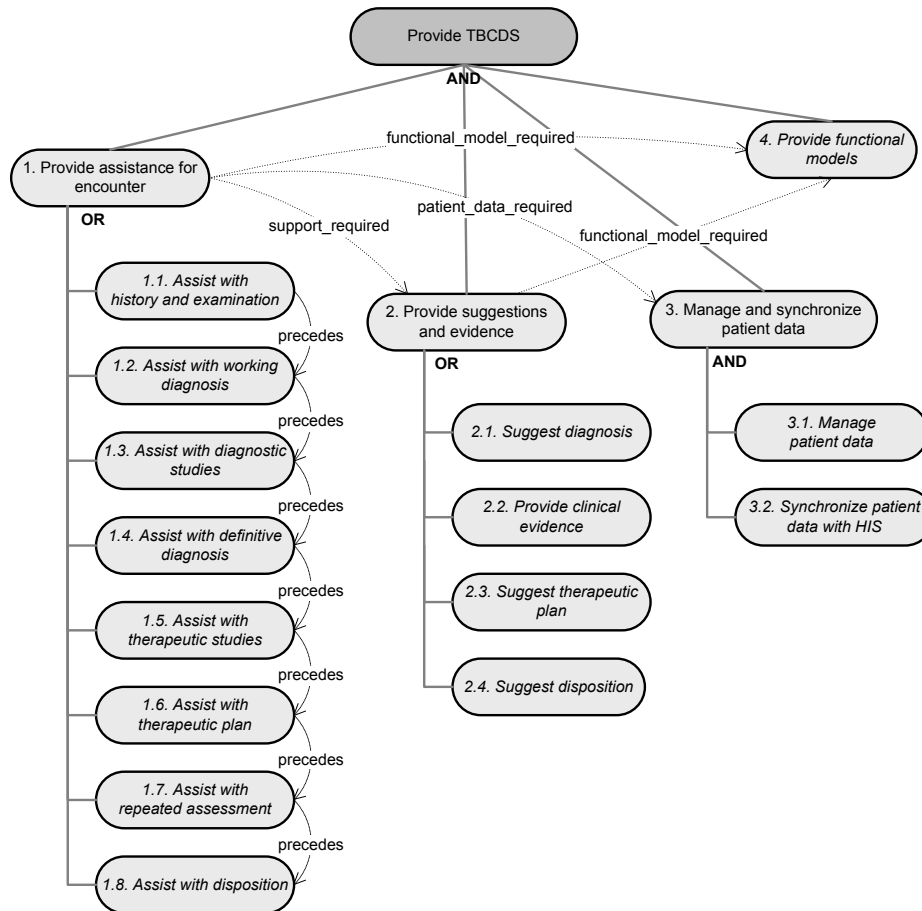
composed into four subgoals corresponding to advanced decision support modalities that may be requested by EPs as they complete specific tasks. These modalities include suggesting diagnoses (working and definitive), providing clinical evidence, and suggesting therapeutic plans and dispositions.

The third goal – *manage and synchronize patient data* – is associated with the requirement of accessing information available in a HIS (see Section 3.2.2). This goal is decomposed into two leaf goals corresponding to patient data management and synchronization accordingly.

Finally, the goal – *provide functional models* – is indirectly associated with the requirement of supporting all observation and reasoning tasks. More specifically, it reflects the activities associated with handling functional models that are needed for completion of the tasks (see Table II). This goal has no subgoals.

The goal model does not include goals associated with the requirement of point-of-care availability (see Section 3.2.3) and with the ability to share and reuse components (see Section 3.2.4). The former requirement was one of the design goals for the domain ontology and therefore did not need to be explicitly stated, and the latter one is satisfied by the virtue of the agent-oriented paradigm extended with ontology-driven design.

Fig. 2. Goal model for the task-based support architecture



The relationships between goals are described using conjunction (AND) and disjunction (OR) operators. For example, *provide suggestions and evidence* is represented as a disjunction of four lower level subgoals because each of the underlying support modalities is optional. On the other hand, *manage and synchronize patient data*, are represented as conjunctions of respective subgoals because each of them has to be satisfied for the higher-level goal to be met.

A goal model introduces a precedence relation indicating that one goal has to be satisfied before another one. This relationship exists for the subgoals of *provide assistance for encounter* to enforce their proper ordering within the task-based emergency workflow model. Another rela-

tionship introduced in the goal model is triggering, indicating that satisfying one goal triggers another one, e.g., *provide assistance for encounter*, triggers *manage and synchronize patient data*, *provide suggestions and evidence* and *provide functional models*.

Considering that the task-based support architecture should allow for developing CDSSs covering wide range of clinical problems, in the goal model presented in Figure 2 we relaxed sequencing requirements of the workflow tasks by allowing for tasks to be skipped. This is indicated by using the OR operator to describe relationships between the *provide assistance for encounter* goal and its subgoals.

4.2. Domain ontology

Figure 3 presents the main concepts and relationships in the domain ontology for the task-based support architecture. The ontology is divided into three components: the data ontology, the interface ontology, and the support ontology.

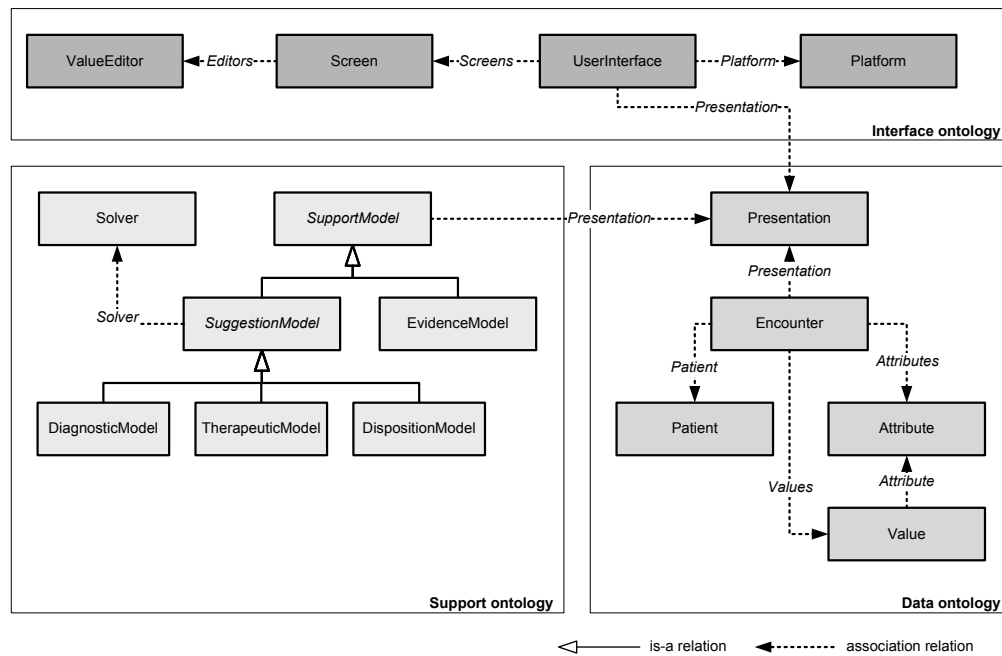
The data ontology defines concepts representing information to be processed. Its structure follows the entity-attribute-value model [48], where a patient-physician encounter is a central entity associated with a set of clinical attributes and described by their values. These attributes define what data should be collected during an encounter given the patient's disease, and are used in formulating diagnoses and therapy plans. In defining attributes we advocate using standardized terminologies endorsed by the ANSI Healthcare Information Technology Standards Panel (SNOMED CT for diseases, findings and diagnoses, LOINC for laboratory observations and RxNorm for medications) [5].

The interface ontology defines concepts representing various components of the user interface (e.g., *screen*, *value editor*). It also introduces the concept of a computing platform to enable defining models that define platform- and disease-specific user interfaces. In deriving these

models we encourage following user-centered and task-centered design [49] and adhering to the requirements from ISO 9241 that deal with broadly understood ergonomics of human-computer interactions. Details of this ontology were already reported in [14].

The support ontology defines concepts corresponding to the *diagnostic*, *therapy* and *disposition* models for suggesting working and definitive diagnoses, therapy plans and dispositions respectively, and to the *evidence model* for providing supporting evidence. It also defines the concept of a *solver* – a generic processing algorithm that is coupled with diagnostic, therapy and disposition models to solve them. Separation of solvers and models improves the flexibility of the design and is advocated in [43].

Fig. 3. Domain ontology for the task-based support architecture



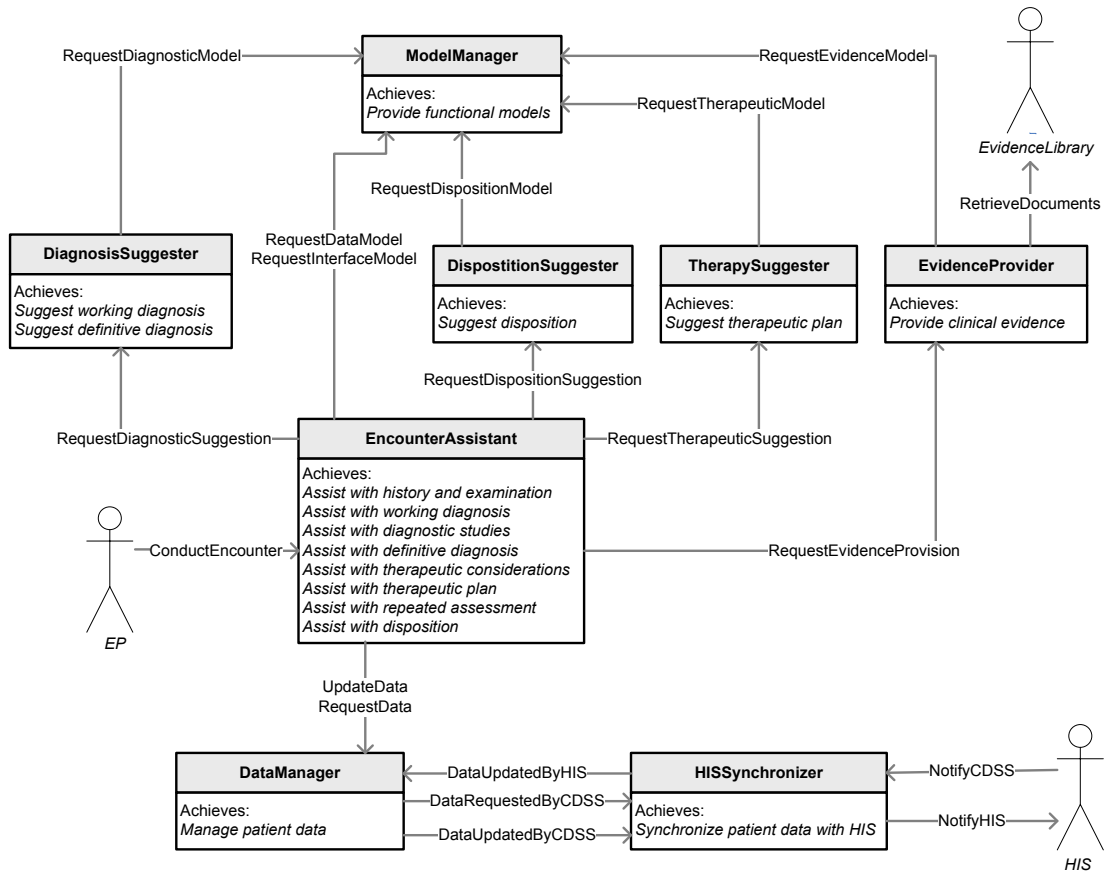
The data ontology and derived data models ensure the stability of diagnostic, therapy, decision and evidence models by presenting an abstract data view (similar to a virtual patient record used in SAGE [50]), that is independent from currently used HISs and relies on standardized

terminologies. This alleviates the so called “curly braces” problem (using CDSS-specific terminologies in decision models) [16], simplifies maintenance of the support models and facilitates their sharing among multiple CDSSs. The mapping between the abstract data view and the data schema of a particular HIS is realized as a relatively simple lookup table that can be easily modified when the data schema changes (e.g., when a different HIS is used).

4.3. *Agent model*

The agent model defines what types of agents are needed to achieve the goals from the goal model and depicts interactions between the agents and all external actors (users and other systems) in terms of communication protocols. The agent model for the task-based support architecture is presented in Figure 4. Each leaf node in the goal model is associated with one agent to ensure that the goal can be satisfied; however, one agent may be capable of achieving more than one goal. In the goal model there are 15 leaf nodes (marked in Figure 2 with italics) that are assigned to 8 agents. The *encounter assistant* agent achieves 8 goals related to assisting the EP during an encounter (they are subgoals of *provide assistance for encounter* in Figure 2). The remaining agents are assigned single goals, e.g., the *therapy suggester* agent is assigned the *suggest therapeutic plan* goal.

Fig. 4. Agent model for the task-based support architecture



The agent model also identifies external actors – the *EP*, *HIS*, and *evidence library*, and captures how agents and external actors interact – with specific interactions being represented as protocols. For example, the *encounter assistant* interacts with the *data manager* using two protocols – *request data* and *update data*.

The central agent in the task-based support architecture is the *encounter assistant*, which acts as the *EP*'s gateway to the system. It is used by the *EP* to control the workflow and to request assistance for specific tasks (this involves requesting services from other agents). Interactions between the *EP* and the *encounter assistant*, and between the agents are described in Table III. While this is not stated explicitly in the table, the *data manager* agent responds to the notifi-

cations from the *HIS synchronizer* about any changes made by other HIS to patient data to preserve its integrity.

Table III. Interactions in the task-based support architecture

Task	Interactions (EP – agent and agent-agent)
History and examination	The EP requests structured data presentation and entry. The <i>encounter assistant</i> queries the <i>model manager</i> for the interface model appropriate for the current patient's disease, and the
Repeated assessment	computing device used by the EP, and requests current patient data from the <i>data manager</i> . Then, the <i>encounter assistant</i> assembles the user interface according to the user interface model and allows the EP to enter new information or modify existing data. After the patient data has been modified, the <i>encounter assistant</i> notifies the <i>data manager</i> about any changes. The <i>data manager</i> conveys this notification to the <i>HIS synchronizer</i> so data stored in other HISs can be modified accordingly.
Working diagnosis	The EP requests a diagnostic suggestion. The <i>encounter assistant</i> passes this request to the
Definitive diagnosis	<i>diagnosis suggester</i> . The <i>diagnosis suggester</i> requests the diagnostic model from the <i>model manager</i> , couples it with a solver, applies this solver to the available patient data and responds to the <i>encounter assistant</i> with a suggested diagnosis. The <i>encounter assistant</i> reports the result to the EP for verification and possible modification. The approved (possibly revised) diagnosis is sent to the <i>data manager</i> that updates patient data and notifies other HISs via the <i>HIS synchronizer</i> .
Diagnostic studies	The EP requests support in ordering selected investigations. The <i>encounter assistant</i> examines the data model for attributes corresponding to possible investigations and presents them for verification and approval. After confirming the selection of tests by the EP, the <i>encounter assistant</i> forwards a request to the <i>data manager</i> . The <i>data manager</i> further passes the request to the computerized physician order entry system via the <i>HIS synchronizer</i> . Once the results are made available by a laboratory information system, they are captured by the <i>HIS synchronizer</i> and passed to the <i>data manager</i> . The <i>data manager</i> finally passes the results to the <i>encounter assistant</i> , so they can be reported to the EP.
Therapeutic considerations	The EP requests assistance in considering different therapies. The <i>encounter assistant</i> passes a request to the <i>evidence provider</i> . The <i>evidence provider</i> requests a proper evidence model from the <i>model manager</i> , retrieves relevant documents from an external evidence library, and passes the documents to the <i>encounter assistant</i> that presents them to the EP.
Therapeutic plan	Similar to the <i>working diagnosis</i> task, the <i>encounter assistant</i> passes a request to the <i>therapy suggester</i> that uses an appropriate therapeutic model to establish a suggestion. Once the therapeutic plan has been revised and confirmed by the EP, it is sent via the <i>HIS synchronizer</i> to the computerized physician order entry system, so it can be acted upon in a seamless fashion.
Disposition	Similar to the <i>working diagnosis</i> task, the <i>encounter assistant</i> passes a request to the <i>disposition suggester</i> that uses a proper disposition model to establish a suggestion.

5. MET3-AE – a pilot implementation of the task-based support architecture

The task-based support architecture was used to create MET3-AE – a CDSS intended to provide the EP with support during the management of pediatric asthma exacerbations [51,52]. The system helps the EP in the early evaluation (within 2 hours of nursing triage) of asthma exacerbations, where management includes establishing the severity of the child’s asthma at a presentation, developing a therapeutic plan and if required, finding supporting clinical evidence.

Interactions between clinical users and MET3-AE are best described with a simple scenario in Figure 5. As this scenario shows, MET3-AE helps with data collection (by interacting with other HIS and providing structured data entry facility), considers information available about a patient to derive upon request a suggested diagnosis, and supports development of a therapeutic plan by providing disease and patient specific medical evidence.

Fig. 5. Scenario of interactions between clinical users and MET3-AE

Peter Smith, a 2-year old boy, is brought by his parents to the ED. Following nursing triage, Dr. Jane Brown takes care of Peter’s management. After the first round of standard treatments according to hospital’s asthma pathway, Dr. Brown continues Peter’s assessment and is using MET3-AE to access and record patient data. Subsequent assessments follow, and Dr. Brown requests MET3-AE for diagnostic suggestion. Using her clinical judgment Dr. Brown diagnoses Peter as having moderate exacerbation (the same diagnosis has been suggested by MET3-AE). Following the asthma practice guideline, she decides to treat Peter with systemic corticosteroids. Before prescribing this treatment, she asks MET3-AE for related clinical evidence regarding such treatment for a young boy. The system responds with a ranked list of abstracted systematic reviews (from The Cochrane Library). Upon consulting the evidence, Dr. Brown’s final disposition is treatment with systemic corticosteroids and the patient’s discharge to care of a family physician.

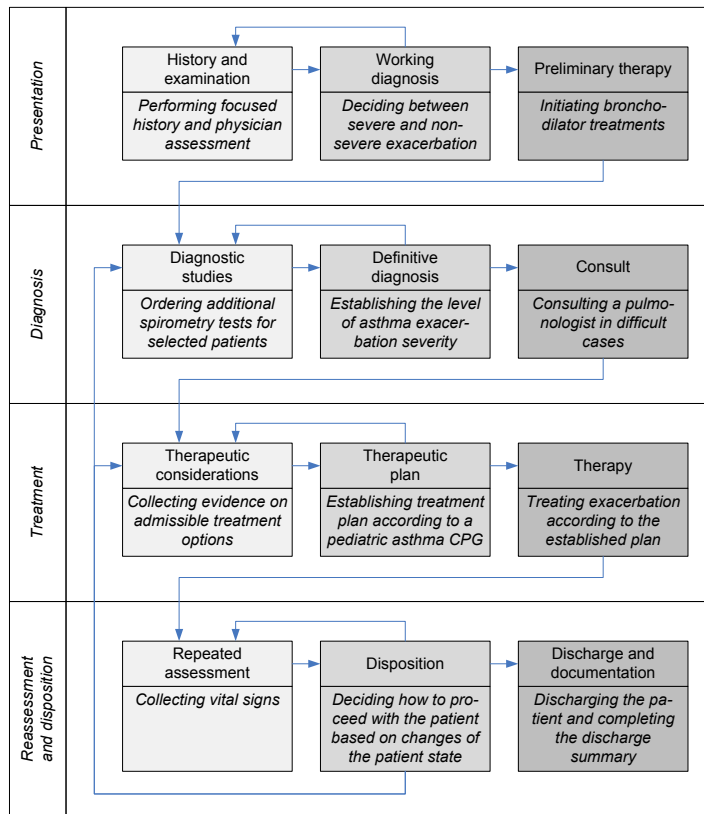
5.1. *Methods*

MET3-AE was developed as a specific instance of the task-based support architecture. Its development followed a 3-step process:

1. Creation of an asthma-specific instance of a generic task-based workflow model; this involved specializing tasks from the model (i.e., specifying the type of diagnoses, possible treatments etc.).
2. Derivation of asthma-specific functional models from the domain ontology; this involved defining clinical attributes and associated user interface entities, constructing diagnostic, therapeutic and disposition decision models with appropriate solvers, and providing an evidence model.
3. Implementation using JADE (Java Agent DEvelopment Framework) [53] and Protégé [54]. JADE is one of the most actively developed multi-agent frameworks, while Protégé is a de facto standard ontological tool in the scientific community [55].

The asthma-specific instance of the task-based workflow was consulted with collaborating EPs from the Children's Hospital of Eastern Ontario (CHEO), Ottawa ON, Canada and it is given in Figure 6 (it includes only the supported phases, i.e., triage and stabilization are not included). In clinical practice at CHEO, The Cochrane Library [56] is used as a source of medical evidence, the therapeutic plan is established according to the pediatric asthma guideline published by the Canadian Association of Emergency Physicians (CAEP) [57], and the patient's progress and disposition is made with regards to the changes in the Pediatric Respiratory Assessment Measure (PRAM) [58].

Fig. 6. The asthma-specific instance of the task-based emergency workflow model



5.2. Results

The functional models for MET3-AE represent knowledge necessary to assist the EP in conducting the observation and reasoning tasks from the workflow model given in Figure 6:

- The data model defines clinical attributes routinely considered when managing and evaluating pediatric asthma exacerbations. A list of the attributes was prepared and provided by the EPs from CHEO;
- The user interface model describes screen forms for presenting and collecting values of attributes from the data model. As advised by the EPs, this model was developed to resemble paper-based charting in terms of labeling and intuitive layout (Figure 7

shows sample user interfaces for a tablet PC, models for other devices were also created but they are not shown here);

Fig. 7. User interface for a tablet PC in MET3-AE

Patient: John Doe with complaint: Asthma Add new assessment

Staff member: Ken Farion Change consent Complete this trial

History | Triage (00:17:48) | Assessment 1

Assessment

SaO2: 94.0 % Room Air Supp. Oxygen Submit this assessment

Temp: 38.0 °C Oral Rectal Axillary Discard this assessment

Heart rate: 45.0 bpm

Resp. rate: 57.0 bpm

Color: Pink/Normal Dusky Pale

Distress: None Mild Moderate Severe

Suprasternal retractions: Absent Present

Scalene retractions: Absent Present

Air entry: Normal Diminished at bases Diminished at widespread Minimal

Wheezing: Absent Expiration Inspi/Expi Audible without a stethoscope

PRAM Score: 6

Personal prediction

The patient's current asthma exacerbation is: Mild Moderate Severe

- The diagnostic model for evaluating the severity of pediatric asthma exacerbations is realized in the form of a naive Bayes classifier. The classifier was developed from data collected in an independent retrospective chart study. We evaluated a number of classifiers frequently applied to medical problems [59] (e.g., decision rules and decision trees), and the naive Bayes classifier demonstrated superior predictive performance, therefore we implemented it;
- The therapeutic model comprises a set of decision rules that point out an appropriate treatment given the patient state. These rules were extracted from the CAEP practice guideline for management of acute pediatric asthma [57];
- The evidence model defines mappings from attributes and their domains defined in the data model to indexing terms used to retrieve evidence-based documentation from

The Cochrane Library. This mapping was constructed automatically using UMLS Metathesaurus [45];

- The disposition model includes a set of decision rules that associate changes in the PRAM score with specific disposition decisions. These rules were created in consultation with the EPs.

All of the above models are deployed in an execution environment that includes all agents from Figure 4 implemented in JADE, and data and model repositories managed by Protégé (the model repository stores asthma management functional models derived from the domain ontology, while the data repository is populated with patient data while the system is being used). The environment also hosts a rule-based solver that is applied to the therapeutic and disposition models by the *therapy suggester* and *disposition suggester* agents respectively.

The execution environment is distributed – the *encounter assistant* agent runs on a client device (e.g., a tablet, desktop PC or a smartphone), and all other agents and repositories reside on a server. The infrastructure interacts with HISs via HL7 messages captured by the Mirth Connect engine [60] and processed by the *HIS synchronizer* agent.

5.3. Clinical evaluation

The MET3-AE system underwent a preliminary evaluation in a prospective cohort study conducted in the ED of CHEO where EPs were using the system in a real time to manage the asthma patients. CHEO ethics review board approved the study under condition that system's diagnostic suggestion is available to EPs only after a disposition decision is made.

The study had two main goals – to evaluate the diagnostic accuracy of the system and to assess its fit to the workflow. The first goal was achieved by comparing diagnostic suggestions provided by the system to a gold standard. **The gold standard (correct diagnostic decisions) was**

established by a single senior EP (blinded to the system's diagnostic suggestions) during chart review and follow-up. In order to evaluate the second goal, we used a post-study questionnaire. While it has been shown that self-reported behavior may be different than the actual one [61], implementing direct observation was impossible due to the limitations imposed by the ED environment and privacy considerations.

The MET3-AE system was deployed on a dedicated server and tablet PCs (Motion Computing C5) that were used at the time in the hospital's ED as wireless client platforms for accessing the ED information system (Eclipsys Sunrise ED Manager) and the admission-discharge-transfer system (EPIC).

The study started in February 2009 and lasted for 13 months. During this time, several interruptions negatively impacted patient recruitment. These included administrative measures and excessive patient volume during the two waves of H1N1 virus activity and the organizational disturbances associated with the construction and move into a new ED facility. Despite these challenges, 102 patients were successfully enrolled and followed.

During the study period MET3-AE was used by 39 members of the ED medical staff – pediatric emergency fellows, senior medical residents and EPs, none of whom were involved in system development. Residents constituted the largest clinicians' group (56%), followed by EPs (36%) and fellows (8%). All residents and fellows were younger than 40 years old while over 50% of the EPs were at least 40 years old. The demographics of the MET3-AE users are in line with the general pediatric medical profession where senior physicians tend to be older.

The MET3-AE user group represented a wide spectrum of expertise – 36% of the participants (EPs) were clinical experts while 64% were novice (fellows and residents). The users also had diverse experience with tablet PCs. To control for this factor, they were given short orienta-

tion sessions about the study, the MET3-AE system, and if necessary, operation of the C5 Motion Computing tablet. After the session all participants were able to use the system without difficulty.

Diagnostic accuracies of MET3-AE and the ED medical staff are given in Table IV.

When computing sensitivity and specificity we labeled the class combining moderate and severe exacerbations as the positive one (in both cases patients should be treated with steroids). While the system was less accurate than the ED medical staff, according to McNemar's test the differences are not statistically significant. Diagnostic suggestions generated ex post by MET3-AE were rated as very usable by clinicians (clinicians were asked to evaluate the system's suggestions after making their disposition decisions and this information was recorded by MET3-AE).

In 80% of cases they stated that the system's suggestion would have reassured or made them revise their opinion had it been available at the time of early diagnosis (in the remaining cases the system's recommendation would have had no influence on respondents' decision).

Table IV. Diagnostic performance of MET3-AE and the ED medical staff

Measure	MET3-AE	ED medical staff
Accuracy [%]	70.7	78.0
Sensitivity	0.696	0.786
Specificity	0.713	0.769

At the end of the study, EPs, fellows and residents who had used MET3-AE, were asked to fill out a questionnaire. The question that is within scope of this paper relate to their overall experience with the MET3-AE (detailed description of the complete questionnaire and associated usability study is given in [62]). 36 clinicians (out of 39) responded to the study, with 20 answering the MET3-AE experience question (response rate for this question being 51%). 8 respondents

were very satisfied or satisfied with the system, 8 rated their experience as average, and 4 as below average. Participants' concerns were mostly related to an unreliable wireless LAN in the ED and hardware issues with tablet PCs (weight, size, lack of touch screen). These shortcomings are well summarized by the following comments – “good idea, just needs better hardware” and “personally – I'd prefer a desktop product – if a desktop could be made available in each exam room”. Due to the layout of ED examination rooms (no space for a desktop computer) MET3-AE was used solely on tablet PCs; however the system can be easily deployed on a desktop computer (an interface model for this platform was also developed).

6. Discussion

In this paper we have presented a novel task-based support architecture that meets recommendations and requirements for developing successful CDSSs. It combines stand-alone and service model architectures and has been designed using the agent-oriented paradigm expanded with ontology-driven design. The backbone of the architecture is the task-based emergency workflow model – we used the O-MaSE method to map each task from the workflow to a specific system goal, and then to associate each of these goals with architectural components (agents).

While the task-based support architecture shares some similarities with workflow systems (for example, reliance on the workflow model to drive the system's behavior and using ontological models to represent domain knowledge), it is distinctly different at its core. The task-based support architecture is focused on decision support rather than coordination (it leaves coordination to the EP) and allows for having complex decision models that go beyond logical expressions.

CDSSs created according to the task-based support architecture provide support directly at the point-of-care for all observation and reasoning tasks. Developing such CDSSs involves

two essential steps: representing a management process for a specific disease as a task-based workflow, and creating functional models required to support specific tasks. In terms of complexity and effort, the second step is more demanding and requires active participation of clinical experts as well as access to verified historical data (if some of the functional models are to be derived with knowledge discovery techniques). The first step is less complex and can be completed with minimal assistance of EPs.

The task-based support architecture allows for sharing and reusing functional models and agents, and enables integration with HISs for bi-directional exchange of information through the *HIS synchronizer* that relies on accepted interoperability standards (such as HL7 and SNOMED CT respectively). Services provided by the agents can be requested and used by other information systems and executed using the task-based support environment. In this way, a CDSS created using task-based support architecture may have functionality similar to service-oriented systems like SEBASTIAN [63] or SANDS [64].

The use of the domain ontology allows for a clear representation of functional models – they can be easily authored by domain experts and viewed by EPs. Such an explicit representation of domain knowledge not only helps with its maintenance but also facilitates data exchange (by using the data model with a mapping layer). The ability to define platform-specific interface models and their easy deployment allows for the creation of a CDSS that can run on a computing device that is most suited to the task at hand. All these features have been identified in [65] as desired characteristics of any CDSS architecture.

The task-based support architecture was implemented as the pilot MET3-AE system. A simulation experiment [51] proved efficiency (in terms of speed) and scalability of the system as well as the underlying task-based support architecture. Agents (especially the *diagnosis suggest-*

er, therapy suggester, disposition suggester and evidence provider) were able to act in parallel handling many simultaneous requests. Moreover, new agent instances could be added at runtime to upgrade performance or to respond to increased workload demands.

In future research we plan to improve the domain ontology – in particular the data and the interface ontologies. Following research on semantic interoperability in HISs [50,66] we plan to use standardized data models from the HL7 Reference Information Model (HL7 RIM) in the data ontology. Although the current approach is sufficient for exchanging single data items using HL7 messages, such an improvement will allow for closer integration of the task-based support architecture with HISs. We also plan to expand the interface ontology so it enables creating user interfaces that are customized not only for a particular computing platform, but also for the level of clinical expertise of a physician (the interface and support provided for an experienced physician should be different than that for a novice [67]).

Some limitations of our research are associated with its emphasis on aiding with decision making at the point-of-care in the ED. The task-based emergency workflow model is specific for managing acute presentations and may not be applicable to other situations (e.g. management of patients with chronic disease). However, the main concepts behind the task-based support architecture and an idea of linking CDSS development with a specific workflow model can be applied across different clinical settings and different workflows. Another set of limitations is associated with clinical evaluation of MET3-AE. Firstly, the hardware platform (tablet PC) used in the study negatively impacted the usability of the system and limited its assessment. Secondly, the study showed that the diagnostic model used by MET3-AE should be improved before it can be deployed at the point-of-care.

7. Conclusions

The task-based support architecture is based on the task-based emergency workflow model. It links CDSS development with a workflow model of disease management by providing functionality required to support all observation and reasoning tasks from the workflow directly at the point-of-care and allowing for interactions with HISs. A reliance on the agent-based paradigm expanded with ontology-driven design enables sharing and reusing of functional models and architectural components. To this end, the proposed architecture not only contributes to the dissemination of good practices in CDSS design – a challenge raised in [8], but also responds to a call for a generic multi-agent CDSS architecture stated in [10].

We have demonstrated that the task-based support architecture goes beyond a theoretical proposal by developing and implementing its instance in the form of the MET3-AE system that supports management of pediatric asthma exacerbation in the ED. While experience of clinicians using the CDSS was mixed, this result was achieved in true ED setting during regular patients' management. To the best of our knowledge, MET3-AE is one of few agent-based systems that has moved beyond a limited evaluation in a laboratory setting [10].

Acknowledgments

The authors would like to thank the reviewers for their constructive comments.

Tomasz Buchert and Tomasz Maciejewski are acknowledged for their work on programming and testing the MET3-AE system and Dr. Martin Michalowski is acknowledged for comments and suggestions on early drafts of the paper. Finally, we would like to thank all the clinicians who participated in the trial and provided us with the feedback on the MET3-AE system.

The research described in this paper was conducted while Dr. Wilk and Dr. O'Sullivan were postdoctoral fellows and Bartosz Kukawka was a research assistant with the MET Research Group at the Telfer School of Management, University of Ottawa.

The research described here was supported by grants from the NSERC-CIHR Collaborative Health Research Program and from the Polish National Science Centre.

References

- [1] Wyatt J, Spiegelhalter D. Field trials of medical decision-aids: potential problems and solutions. *Proc Annu Symp Comput Appl Med Care* 1991;3-7.
- [2] Hunt DL, Haynes RB, Hanna SE, Smith K. Effects of computer-based clinical decision support systems on physician performance and patient outcomes: a systematic review. *JAMA* 1998;280(15):1339-46.
- [3] Berner ES. *Clinical decision support systems - state of the art*. Rockville, MD: Agency for Healthcare Research and Quality; 2009.
- [4] Garg AX, Adhikari NK, McDonald H, Rosas-Arellano MP, Devereaux PJ, Beyene J, et al. Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review. *JAMA* 2005;293(10):1223-38.
- [5] Eichner J, Das M. *Challenges and barriers to clinical decision support (CDS) design and implementation experienced in the Agency for Healthcare Research and Quality CDS demonstrations*. Rockville, MD: Agency for Healthcare Research and Quality; 2010.
- [6] Kawamoto K, Houlihan CA, Balas EA, Lobach DF. Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. *BMJ* 2005;330(7494):765-72.

- [7] Bates DW, Kuperman GJ, Wang S, Gandhi T, Kittler A, Volk L, et al. Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality. *J Am Med Inform Assoc* 2003;10(6):523-30.
- [8] Sittig DF, Wright A, Osheroff JA, Middleton B, Teich JM, Ash JS, et al. Grand challenges in clinical decision support. *J Biomed Inform* 2008;41(2):387-92.
- [9] Fieschi M, Dufour JC, Staccini P, Gouvernet J, Bouhaddou O. Medical decision support systems: old dilemmas and new paradigms? *Methods Inf Med* 2003;42(3):190-8.
- [10] Isern D, Sanchez D, Moreno A. Agents applied in health care: A review. *Int J Med Inform* 2010;79:145-66.
- [11] Jennings NR, Wooldridge M. Agent-oriented software engineering. In: Bradshaw J, editor.: AAAI/MIT Press; 2001.
- [12] Wooldridge M, Ciancarini P. Agent-oriented software engineering: the state of the art. In: Ciancarini P, Wooldridge M, editors. *Agent-Oriented Software Engineering. First International Workshop, AOSE 2000 Limerick, Ireland, June 10, 2000. Revised Papers.* Berlin: Springer; 2001, p. 1-28.
- [13] Jennings N. On agent-based software engineering. *Artif Intell* 2000;117(2):277-96.
- [14] Farion K, Michalowski W, Wilk S, O'Sullivan D, Rubin S, Weiss D. Clinical decision support system for point of care use: ontology driven design and software implementation. *Methods Inf Med* 2009;48(4):381-90.
- [15] Michalowski W, Rubin S, Slowinski R, Wilk S. Mobile clinical support system for pediatric emergencies. *Decis Support Syst* 2003;36(2):161-76.
- [16] Wright A, Sittig DF. A four-phase model of the evolution of clinical decision support architectures. *Int J Med Inform* 2008;77(10):641-9.
- [17] Pokahr A, Braubach L, Jander K. Unifying agent and component concepts. In: Dix J, Witteveen C, editors. *Multiagent System Technologies. The 8th German Conference, MATES 2010,*

- Leipzig, Germany, September 27-29, 2010. Proceedings. Berlin/Heidelberg: Springer 2010, p. 100-12.
- [18] Gómez-Sanz J, Pavón J. Methodologies for developing multi-agent systems. *J Univers Comput Sci* 2004;10(4):359-74.
- [19] Hollinsworth D. The workflow reference model: Workflow Management Coallition; 1994.
- [20] Huser V, Rasmussen LV, Oberg R, Starren JB. Implementation of workflow engine technology to deliver basic clinical decision support functionality. *BMC Med Res Methodol* 2011;11:43-61.
- [21] Lee J, Kim J, Cho I, Kim Y. Integration of workflow and rule engines for clinical decision support services. *Stud Health Technol Inform* 2010;160(Pt 2):811-5.
- [22] Zhao JL. Intelligent agents for flexible workflow systems. In: Proceedings of the AIS Americas Conference on Information Systems. Baltimore: MD; 1997, p. 238-9.
- [23] Isern D, Sánchez D, Moreno A. HeCaSe2: a multi-agent ontology-driven guideline enactment engine. In: Burkhard H-D, Lindemann G, Verbrugge R, Varga L, editors. Multi-agent systems and applications V. 5th International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2007, Leipzig, Germany, September 25-27, 2007. Proceedings: Springer; 2007, p. 322-4.
- [24] Isern D, Moreno A, Sánchez D, Hajnal Á, Pedone G, Varga LZ. Agent-based execution of personalised home care treatments. *Appl Intell* 2009;34(2):155-80.
- [25] Braun L, Wiesman F, van der Herik J, Hasman A. Agent support in medical information retrieval. In: Cortes U, Fox J, Moreno A, Nealon J, editors. Proceedings of the 3rd Workshop on Agents Applied in Health Care, IJCAI 2005, AAAI. Edinburgh; 2005, p. 16-25.
- [26] Horsky J, Gutnik L, Patel VL. Technology for emergency care: cognitive and workflow considerations. *AMIA Annu Symp Proc* 2006:344-8.
- [27] Perina DG, Beeson MS, Char DM, Counselman FL, Keim SM, McGee DL, et al. The 2007 model of the clinical practice of emergency medicine: the 2009 update 2009 EM Model Review Task Force. *Ann Emerg Med* 2011;57(3):e1-15.

- [28] Lenz R, Reichert M. IT support for healthcare processes – premises, challenges, perspectives. *Data Knowl Eng* 2007;61(1):39-58.
- [29] Landman AB, Bernstein SL, Hsiao AL, Desai RA. Emergency department information system adoption in the United States. *Acad Emerg Med* 2010;17(5):536-44.
- [30] Prietula MJ, Simon HA. The experts in your midst. *Harvard Bus Rev* 1989;67(1):120-4.
- [31] Bagga W, Molva R. Policy-based cryptography and applications. In: Goos G, Hartmanis J, van Leeuwen J, editors. *Financial Cryptography and Data Security. 9th International Conference, FC 2005, Roseau, The Commonwealth of Dominica, February/March, 2005. Revised papers.* Berlin: Springer; 2005, p. 72-87.
- [32] Körner H, Söndena K, Söreide JA, Andersen E, Nysted A, Lende TH. Structured data collection improves the diagnosis of acute appendicitis. *Br J Surg* 1998;85(3):341-4.
- [33] Kong GL, Xu DL, Yang JB. Clinical decision support systems: a review on knowledge representation and inference under uncertainties. *Int J Comput Intell Syst* 2008;1(2):159-67.
- [34] Bakken S, Currie LM, Lee N-J, Roberts WD, Collins SA, Cimino JJ. Integrating evidence into clinical information systems for nursing decision support. *Int J Med Inform* 2008;77(6):413-20.
- [35] Shortliffe EH. The adolescence of AI in medicine: will the field come of age in the 90's? *Artif Intell Med* 1993;5(2):93-106.
- [36] Burkle T, Schweiger R, Altmann U, Holena M, Blobel B, Dudeck J. Transferring data from one EPR to another: content-syntax-semantic. *Methods Inf Med* 1999;38(4-5):321-5.
- [37] Mori AR, Consorti F. Integration of clinical information across patient records: a comparison of mechanisms used to enforce semantic coherence. *IEEE Trans Inf Technol Biomed* 1998;2(4):243-53.
- [38] Berlin A, Sorani M, Sim I. A taxonomic description of computer-based clinical decision support systems. *J Biomed Inform* 2006;39(6):656-67.
- [39] Ammenwerth E, Buchauer A, Bludau B, Haux R. Mobile information and communication tools in the hospital. *Int J Med Inform* 2000;57(1):21-40.

- [40] Folz-Murphy N, Partin M, Williams L, Harris CM, Lauer MS. Physician use of an ambulatory medical record system: matching form and function. *Proc AMIA Symp* 1998:260-4.
- [41] DeLoach SA. Engineering organization-based multiagent systems. In: Garcia A, Choren R, Lucena C, Giorgini P, Holvoet T, Romanovsky A, editors. *Software engineering for multi-agent systems IV: research issues and practical applications*. Berlin: Springer; 2006, p. 109-25.
- [42] Crubézy M, O'Connor M, Buckeridge DL, Pincus Z, Musen MA. Ontology-centered syndromic surveillance for bioterrorism. *IEEE Intell Syst* 2005;20(5):26-35.
- [43] Musen MA. Scalable software architectures for decision support. *Methods Inf Med* 1999;38(4-5):229-38.
- [44] Swartout W, Tate A. Ontologies. *IEEE Intel Syst App* 1999;14(1):18-9.
- [45] O'Sullivan D, Wilk S, Michalowski W, Farion K. Automatic indexing and retrieval of encounter-specific evidence for point-of-care support. *J Biomed Inform* 2010;43(4):623-31.
- [46] Michalowski W, Wilk S, Farion K, Pike J, Rubin S, Slowinski R. Development of a decision algorithm to support emergency triage of scrotal pain and its implementation in the MET system. *INFOR* 2005;43(4):287-301.
- [47] Błaszczyński J, Farion K, Michalowski W, Wilk S, Rubin S, Weiss D. Mining clinical data: selecting decision support algorithm for the MET-AP system. In: Miksch S, Hunter J, Keravnou ET, editors. *Artificial Intelligence in Medicine. 10th Conference on Artificial Intelligence in Medicine, AIME 2005, Aberdeen, UK, July 23-27, 2005. Proceedings*. Berlin/Heidelberg/New York: Springer; 2005, p. 429-33.
- [48] Nadkarni PM, Marenco L, Chen R, Skoufos E, Shepherd G, Miller P. Organization of heterogeneous scientific data using the EAV/CR representation. *J Am Med Inform Assoc* 1999;6(6):478-93.
- [49] Wilk S, Michalowski W, Farion K, Kersten M. Interaction design for mobile clinical decision support systems: the MET system solutions. *Found Comput Decis Sci* 2007;32(1):47-61.

- [50] Tu SW, Campbell JR, Glasgow J, Nyman MA, McClure R, McClay J, et al. The SAGE Guideline Model: achievements and overview. *J Am Med Inform Assoc* 2007;14(5):589-98.
- [51] Sayyad Shirabad J, Wilk S, Michalowski W, Farion K. Implementing an Integrative Multi-agent Clinical Decision Support System with Open Source Software. *J Med Syst* 2012;36(1):123-37.
- [52] Wilk S, Michalowski W, Farion K, Sayyad Shirabad J. MET3-AE system to support management of pediatric asthma exacerbation in the emergency department. *Stud Health Technol Inform* 2010;160(Pt2):841-5.
- [53] Bellifemine FL, Caire G, Greenwood D. Developing multi-agent systems with JADE. Chichester: John Wiley & Sons Ltd; 2007.
- [54] Gennari JH, Musen MA, Fergerson RW, Grosso WE, Crubézy M, Eriksson H, et al. The evolution of Protégé: an environment for knowledge-based systems development. *Int J Hum Comput Stud* 2003;58(1):89-123.
- [55] Khondoker MR, Mueller P. Comparing ontology development tools based on an online survey. In: *Proceedings of the World Congress on Engineering (WCE)*. London; 2010, p. 188-92.
- [56] The Cochrane Collaboration [Internet]. The Cochrane Library [cited August 3, 2011]. Available from: <http://www.thecochranelibrary.com/>.
- [57] Alberta Clinical Practice Guidelines Program. Guidelines for the management of acute asthma in adults and children: Alberta Medical Association; 2002.
- [58] Ducharme FM, Chalut D, Plotnick L, Savdie C, Kudirka D, Zhang X, et al. The Pediatric Respiratory Assessment Measure: a valid clinical score for assessing acute asthma severity from toddlers to teenagers. *J Pediatr* 2008;152(4):476-80, 80 e1.
- [59] Lavrac N. Selected techniques for data mining in medicine. *Artif Intell Med* 1999;16(1):3-23.
- [60] Mirth Corporation [Internet]. Mirth Connect [cited 2011 Aug 8]. Available from: <http://www.mirthcorp.com/community/mirth-connect>.
- [61] Henry K, Campbell S, Maki M. A comparison of observed and self-reported compliance with universal precautions among emergency department personnel at a Minnesota public teaching

- hospital: implications for assessing infection control programs. *Ann Emerg Med* 1992;21(8):940-6.
- [62] O'Sullivan D, Doyle J, Michalowski W, Wilk S, Farion K, Kuziemy C. Assessing the Motivation of MDs to use Computer-based Support at the Point-of-Care in the Emergency Department. *AMIA Annual Symposium Proceedings* 2011:1045-54.
- [63] Kawamoto K, Lobach DF. Design, implementation, use and preliminary evaluation of SEBASTIAN, a standards-based Web service for clinical decision support. *AMIA Annu Symp Proc* 2005:380-4.
- [64] Wright A, Sittig DF. SANDS: a service-oriented architecture for clinical decision support in a National Health Information Network. *J Biomed Inform* 2008;41(6):962-81.
- [65] Wright A, Sittig DF. A framework and model for evaluating clinical decision support architectures. *J Biomed Inform* 2008;41(6):982-90.
- [66] Lopez DM, Blobel B. Enhanced semantic interoperability by profiling health informatics standards. *Methods Inf Med* 2009;48(2):170-7.
- [67] Hine MJ, Farion K, W. Michalowski, Wilk S. Decision making by emergency room physicians and residents: implications for the design of clinical decision support systems. *Int J Healthc Inf Syst Inform* 2009;4(2):17-35.