Task Knowledge Model for Triage Decision-Support

Shamimi A. Halim[1], Muthukkaruppan Annamalai[1], Rashidi Ahmad[2] and Mohd Sharifuddin Ahmad[3]
[1]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam, Malaysia*
[2]*Trauma and Emergency Department, University Malaya Medical Centre, Kuala Lumpur, Malaysia*
[3]*Centre for Agent Technology, Universiti Tenaga Nasional,* Kajang, *Malaysia*

Keywords: Triage Decision-Support, Task Modelling, Commonkads.

Abstract: The paper discusses the construction of the task knowledge model to support the development of a triage decision-support system. Knowledge rather than experience is predominant in the triage decision making. Since the triage decision making knowledge is complex, we resort to knowledge modelling to help in the systematisation of the knowledge. The paper concentrates on the modelling of the task knowledge model to back triage decision-support. We adopted the CommonKADS methodology as a basis of modelling and engineering the knowledge. The top-down modelling approach availed general task structures that could be reused and adapted to engineer the triage decision-support task knowledge model. Consequently, the resulting task model informs the engineering of the triage decision-support domain knowledge model.

1 INTRODUCTION

The triage decision making that takes place in hospital Emergency Department (ED) involves clinical judgments that need to be made quickly under conditions of uncertainty. The decisions however, have a major impact on the mortality and mobility of the patients. A common characteristic in drawing the triage decision depends on the knowledge and experience of the triage officers (Considine et al., 2007). Then again, studies have shown that knowledge plays a more important role compared to experience in determining the triage decisions (LeVasseur et al., 2001).

In our previous work (Halim et al., 2011), we have noted that the benefits of implementing a uniform and more robust triage in Malaysian EDs that can ensure consistent triage decisions, which is possible with the aid of a Triage Decision-Support System (TDSS). The TDSS can help to determine the 'right' triage level of a patient by reasoning over the represented triage decision-support knowledge. However, the triage decision making knowledge is rather complex. It consists of factual and procedural knowledge gathered from decision rules and clinical practices and guidelines. Therefore there is a need to organise the supporting knowledge before we develop the TDSS. For that reason, we resorted to knowledge modelling to help in the systematisation of the triage decision-support knowledge.

Two crucial models of knowledge in a knowledge-based system are the task and the domain knowledge models (Annamalai, 2006). These models can help in the understanding of a knowledge intensive process, and lead the way to interact with them. The modelling of the domain knowledge is directed by its purposive mechanism (Annamalai and Sterling, 2003). In this regard, the triage decision-support task knowledge model informs the representation of the knowledge resources to support the task. Cognitive tasks involve inferences. Some knowledge modelling methods additionally advocate the use of an inference knowledge model to elaborate the inference knowledge (Tarta, 2004).

The paper is organised as follows. Section 2 analyses the top-down knowledge engineering and modelling methods. Section 3 describes the modelling of the triage decision-support task. Section 4 briefly discusses its validation. Finally, section 5 concludes the paper and points to future work.

A. Halim S., Annamalai M., Ahmad R. and Sharifuddin Ahmad M..
Task Knowledge Model for Triage Decision-Support.
DOI: 10.5220/0004546802610268
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2013), pages 261-268
ISBN: 978-989-8565-81-5
Copyright © 2013 SCITEPRESS (Science and Technology Publications, Lda.)

Table 1: Contemporary knowledge engineering methods.

Knowledge Engineering Method	Strength	Our requirement
Generic Task	Inspired by diagnostic and design tasks	×
	Include fixed PSM strategy which specify the inference steps	×
	Provides task specific vocabulary	√
Role-limiting	Views PSM as fixed	×
	Provides a set of predefined terminology	√
MIKE	Proposes informal and semi-formal specification techniques to describe knowledge	√
	Uses executable KARL	×
	Applies reversible process in system development	×
Protégé-II	Allows development of PSM independently from the knowledge base	√
	Decomposable of PSM into sub-method enable a configuration of generic problem-solver	√
	The domain layer is comprise in domain ontology	√
	Platform specific design and implementation	×
KADS CommonKADS	Has a broad view of the process during early stage of requirement and analysis	√
	The Model of Expertise provides four layers of knowledge: domain, inference, task and strategy layer	√
	Provides general knowledge-intensive task templates	√
	Emphasize on the internal control	√
	Platform independent design and implementation	√

2 KNOWLEDGE ENGINEERING AND MODELLING METHODS

Knowledge engineering extracts the concepts and relationships among them from the knowledge sources and resources, and defines them in knowledge models.

We propose to adopt a top-down knowledge engineering approach to the modelling of the triage decision-support knowledge. The top-down modelling approach avail general task structures that could be reused and adapted to engineer the knowledge models (Kingston, 2007). Knowledge acquisition is directed and focussed to knowledge that is relevant to the problem in hand. As a result, the time required for knowledge acquisition and analysis will be reduced.

Contemporary top-down knowledge modelling and engineering methods are Generic task (Chandrasekaran, 1986), Role-limiting (Marcus, 1988), MIKE (Angele et al., 1998), Protégé-II (Gennari et al., 2003), KADS (Schreiber et al., 1993) and CommonKADS (Schreiber et al., 2000). Table 1 highlights the strengths of each method and indicate the ones (with √), which we think are useful to support the development of the TDSS.

Inspired by diagnostic and design task, the Generic Task try to solve different types problems by creating a taxonomy or vocabulary which appropriate for a particular domain knowledge. However, the clarity of knowledge representation is weak because the fact that the languages to implement the expert system is not standardized across the tasks. On the other hand, the Role-limiting method separates the Problem Solving Method (PSM) from the domain knowledge where the object and their relation including the environment are fixed building block. It also provides a predefined terminology. The orientation of the terminology is a problem-solving-method-specific, and not domain-specific. This feature gives flexibility to knowledge engineer to accommodate a particular domain

MIKE proposes the informal and semi-formal specification techniques to describe the knowledge. MIKE uses KARL (Knowledge Acquisition and Representation Language) to describe the functionality of the knowledge precisely. Since it is an executable language, the specification will be developed based on prototyping approach and the functionality can be tested by a running prototype. Another special feature of MIKE is the ability to increment and reverse during system development process. However, the ability to develop the system is not required since we do not intend to develop the TDSS in this manner to test its functionality.

Protégé-II provides several PSMs which were developed separately from the knowledge base and those PSMs can be used to work with different knowledge bases and solve different real-world problems. Ever though this feature is not significant in the development of TDSS, the decomposable of generic PSM featuring in Protégé-II will help a lot. The domain layer in Model of Expertise in Protégé-II was captured in domain ontology and the other three layers were kept optional and can be used for any appropriate PSM. This separation however limits the system-level view of the process, particularly in triage process. The implementation is however within the Protégé knowledge acquisition environment.

KADS on the other hand provides all layers of knowledge in Model of Expertise. Therefore the

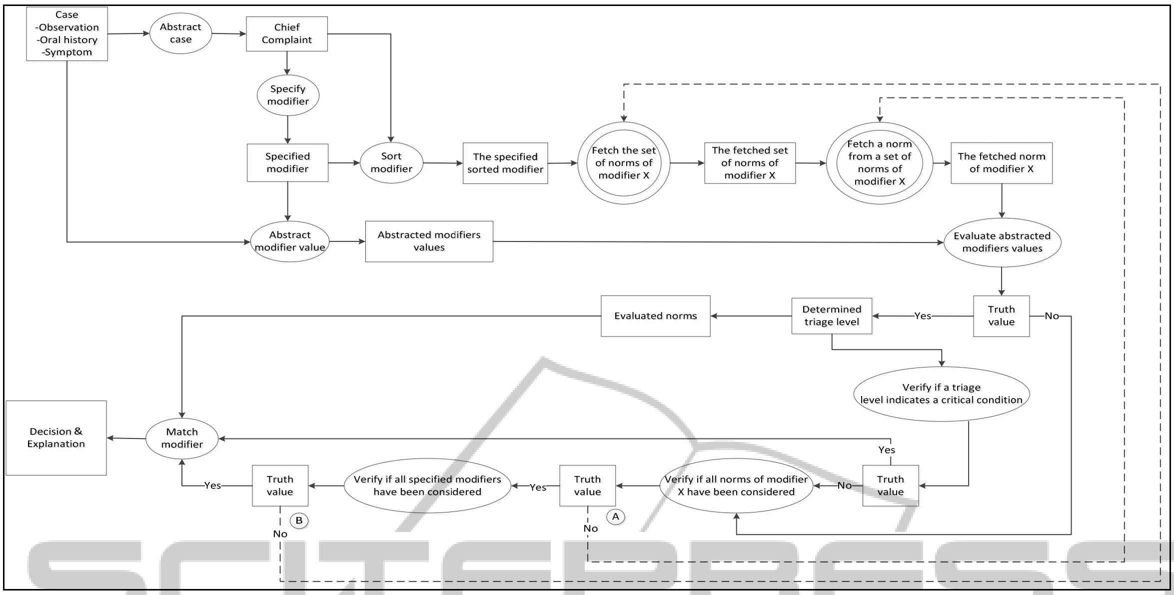

Figure 1: Triage Decision-support Task model.

domain, procedural, inferences needed and any election of tasks can be viewed. The KADS and the evolved CommonKADS methods support the modelling of knowledge-intensive tasks which divided into analytic and synthetic tasks. The task structures are captured generically as task templates. The template of each task type is flexible to addition and modification of its inference in order to fit a particular application task. Therefore, this method gives a flexibility to control the reasoning process. It allowed us to capture the expert reasoning strategies especially in sequencing the reasoning steps.

Among the task types, we found the Assessment task structure as a suitable to adapt for engineering the triage decision-support task knowledge. The goal of Assessment task is to determine a decision for a set of case (condition or event) with domain-specific norms as a rule. In the context of triage decision making, a set of case will be presented as a patient's condition while the norms consist of established triage guidelines or scale. CommonKADS has emerged as an industrial strength knowledge engineering method, and have been used in many cases (Lindow et al., 2013).

3 TRIAGE DECISION-SUPPORT TASK MODELLING

The explanation about this section will begin with structuring the triage decision-support task model and followed by inference models.

3.1 Triage Decision-Support Task Model

We engineered the triage decision-support task model by adapting the generic CommonKADS Assessment task. Figure 1 shows the resulting triage decision-support task model. In the process, we have made several modifications to the Assessment task structure. The Sort and Verify inferences are new. The Sort inference is introduce since the triage decision deals with a set of case which consist of more than one elements that need to be prioritized according to a particular preference. The Select inference is replaced by Fetch, which is a non-cognitive action. This action will fetch the element that has been prioritized by the Sort inference. The decision-support process flow, as a whole, has been revised and restructured to reflect the flow and processing of knowledge in triage decision making.

In Table 2, we describe the key knowledge or information resources utilised in the triage decision-support task.

The dotted flow line A in Figure 1 points to the iterative addition of a modifier under consideration. If there are no norms to be considered for that modifier, the evaluation continues using other modifiers (indicated by dotted flow line B).

As shown in Figure 1, there are eight inferences in our triage decision-support task model: *Specify modifier, Abstract modifier value*, *Sort modifier*, *Evaluate abstracted modifiers value*, *Verify if a*

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

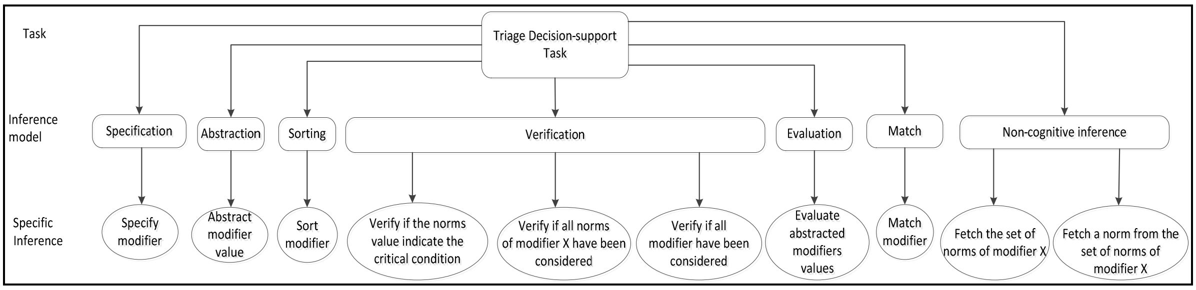

Figure 2: Triage Decision-support Task, Inference model and the Specific Inferences.

Table 2: The key knowledge/ information resources utilised in the Triage decision-support task.

Term	Description
Case	Knowledge or information gathered from the patient, which consists of oral history, vital signs and observation.
Chief Complaint (CC)	The most significant illness inferred from Case (based on Habboushe's guide (Habboushe, 2012))
Modifiers	A determinant to determine a triage level. Eleven modifiers are involved (Murray, Bullard and Grafstein, 2004): Glasgow Coma Score (GCS), Respiratory distress, Hemodynamic stability, Dedicated presenting complaint, Mental health, Bleeding severity, Hypertension, Temperature, Pain, Mechanism of injury and Blood glucose.
Modifier value	The value assign to a modifier. Ex. Modifier value is 38°C for modifier Temperature.
Norms of modifier / Modifier's norm	A conditional rule of a modifier. Each modifier has more than one rule. The following is example of a norm. IF level of distress is severe THEN triage level is I
Triage level	The outcome of a norm's evaluation. Triage levels consist of level one to level five in a robust triage scale (Gerdtz et al, 2009). The triage level indicates the severity of a patient's clinical condition.
Critical condition	The critical condition refers to triage level I and II, i.e., when the patient must be given immediate treatment.

triage value indicates a critical condition, *Verify if all norms of modifier X have been considered*, *Verify if all modifiers have been considered* and *Match modifier*. Another two non-cognitive actions are *Fetch the set of norms of modifier X* and *Fetch a norm from a set of norms of modifier X.* These inferences will be described in section 3.2.1, 3.2.2, 3.2.3, 3.2.4, 3.2.5, 3.2.6 and 3.2.7, respectively.

3.2 Triage Inference Models

Based on the triage decision-support task model, we have decomposed six intermediate inference subtasks, namely Specification, Abstraction, Sorting, Verification, Evaluation and Match. The decomposition of the task describes the control of sequence of task design and help in determining the inference models. Figure 2 shows the task decomposition of the triage decision-support task. Triage decision-support is the most general task.

The inference knowledge details the reasoning mechanism for the triage decision-support solution. This type of knowledge is described by specifying the performed function and their input and output. The six inference models that will guide the representation of the specific inference knowledge are: Specification model, Abstraction model, Selection model, Verification model, Evaluation model and Matching model. These models are described in the CommonKADS reference (Schreiber et al., 2000).

The following sub-sections will explain the inferences models.

3.2.1 Specification Inference

The Specification inference in the context of triage decision-support will identify CC as an input and determines a list of modifier as output. The output is produced by inferring over the Specify inferencing knowledge.

The list of modifiers consists of two groups: Specified and Optional modifier. A specified modifier is a modifier which has been identified by a domain expert. The selection of modifier is primarily determined by the degree of severity of the symptom. For example headache is a symptom and increase intra cranial pressure, migraine and stress are causes of headache. The level of triage is determined by the Pain modifier value not by the cause of headache and together with other modifiers: GCS, Hypertension, Presenting complaint, Bleeding and Mechanism of injury. The optional modifier is compliment handle for the triage officer, in case more modifiers are needed. The following pseudo code broadly describes the Specify inference knowledge. Note Variable X represents a particular CC.

```
READ CC from chief complaint list
IF CC is X
THEN DISPLAY list of Specified
   modifiers for X
```

3.2.2 Abstraction Inference

The Abstraction inference has case description as input and an abstracted case description as output. A case description can be a description about certain conditions, situations or any attributes that explain the entity. For example, a value of Respiratory modifier "cyanosis" is abstracted into severe level of distress.

In the triage decision-support task, out of the eleven modifiers, four modifier values are abstracted indirectly (computed) from the case, while other values can be extracted directly. The following pseudo code describes part of the Abstraction inference knowledge.

```
READ modifier value for GCS from
   list of Specified modifier for GCS
READ modifier value for Presenting
   Complaint from list of specified
   modifier for Presenting Complaint
...
...

IF modifier value for GCS is not
   null
THEN CALL function abstraction GCS
IF modifier value for Temperature is
   not null
THEN abstracted value Temperature =
   modifier value for Temperature
...
...
```

In the above example, the GCS modifier value has to be abstracted indirectly, while the Temperature can be extracted directly from the case. The following pseudo code explains the identified abstraction of the GCS modifier value.

```
READ eye opening response from list
   of specified modifier for GCS
READ verbal response from list of
   specified modifier for GCS
READ motor response from list of
   specified modifier for GCS

IF eye opening response is
   spontaneous
THEN eye point = 4
IF eye opening response is verbal
   stimuli
THEN eye point = 3
...
...
IF verbal response is oriented
THEN verbal point = 5
IF verbal response is confused
THEN verbal point = 4
...
...
IF motor response is obeys commands
THEN motor point = 5
IF motor response is withdraws in
   response to pain
THEN motor point = 4
...
...
COMPUTE abstracted value GCS = eye
   point + verbal point + motor point
```

3.2.3 Sorting Inference

A sorting inference has a set of elements as input and a sorted list which contains the same elements as an output. This inference decides the relative order of two or more elements. In the triage decision-support task, the Sorting inference prioritizes the specified modifiers (input) to be considered based on CC. The example of sorted modifier (output) for Abdominal pain CC is Pain, Hemodynamic, Respiratory, Hypertension and Presenting complaint.

The following pseudo code describes a piece of the Sorting inference knowledge which will show the order of priority Specified modifiers for a CC when their order is known.

```
READ CC from chief complaint list
READ list of specified modifier
IF CC is X
THEN OUTPUT list of specified sorted
   modifiers for X
```

However, in triage decision making, the modifiers of certain CC are not specific. In such situation, the principle of emergency care will be applied to sort the modifiers. The principle of emergency care highlights the element of the modifiers represented by the acronym of ABCD (A=airway; B=breathing, C= circulation; D=dysfunction of central nervous system) to relieve suffering and to prevent further deterioration of the illness. This knowledge has been used in practice to prioritise the modifiers.

3.2.4 Evaluation Inference

The inputs of Evaluation inference consist of two components: a set of data and a norm. Data is evaluated with a norm based on evaluation criteria. The truth value is derived, which indicates whether or not the data complies with the norm.

In the triage decision-support task, the Abstracted modifiers values and the Fetched norm of modifier X are data and norm, respectively. The evaluation knowledge consists of rules that examine whether the abstracted values comply with the norm in hand. A truth value of 1 indicates that the Fetched norm of modifier X fulfils the Abstracted modifiers values. In case of failure (0), the next norm of modifier X is considered. The following pseudo code describes a piece of the evaluation inference knowledge.

```
READ norm i of modifier X
READ abstracted value modifier X
IF (norm i of modifier X ==
   abstracted value modifier X) is
   true
THEN truth value = 1
   GET triage level value
   GET abstracted modifier X value
ELSE truth value = 0
...
...
```

SCIENCE AND TECHNOLOGY PUBLICATIONS

3.2.5 Verification Inference

The Verification inference is used to test a description of the system based on certain hypothesis. A system description represents a condition or event that has to be tested and the output for this task is a truth value, which indicates whether the system has passed the test. The violation is also an output.

The Verification inferences in triage decision-support verify three different events: Verify if a norm value indicates a critical condition, Verify if all norms of modifier X have been considered and Verify if all modifiers have been considered. The following paragraph discusses the first event.

This event is to verify whether the determined triage level indicates the critical condition which is triage level I and II. The truth value from this verification also determines whether there is a need to consider the other norms if the verification fails. If the verification succeeds, the following Match inference takes over. The following pseudo code describes part of the Verify critical condition inference knowledge.

```
READ triage level value
IF triage level value = 1
THEN GO TO Match modifier
ELSE GO TO Verify all norms
```

3.2.6 Matching Inference

The input for the Matching inference is an abstracted case description, which describes a particular event or entity and a set of norms represents the rules that indicate whether the description leads to decision.

The purpose of this inference in triage decision-support is to provide justified explanation for the determined triage level (decision). The explanation is based on norms that meet the lowest triage level. The following pseudo code broadly describes the matching inference knowledge.

```
FOR (norm i of modifier X ==
   abstracted value modifier X) is
   true
   READ all norm i of modifier X
   READ all norm i of modifier X
END FOR

DETERMINE lowest triage level value
READ norm i of modifier X with
   lowest triage level value
PRINT norm i of modifier X with
   lowest triage level value
PRINT triage level value
```

3.2.7 Fetch Action

Fetch is a non-cognitive action that appears in the triage decision-support task. This action fetches a set of norms for modifier X. The following pseudo code describes a piece of the Fetch inference knowledge.

```
READ list of specified sorted
   modifiers for CC

OBTAIN number of modifiers from
   specified sorted modifier for CC
...
...
IF specific sorted modifier i is X
THEN FETCH set norms of modifier X
```

4 VALIDATION OF THE TASK KNOWLEDGE MODEL

Once the understanding of the triage decision-support task is clear, the key resources utilised in the triage decision-support task will serve the basis of validation of the task knowledge model. Subsequently, the modelling of the purposive domain knowledge will be based on these identified resources (Annamalai, 2006; Annamalai and Sterling, 2003). In the ensuring paragraphs, we will discuss these knowledge resources. Due to page limitation, we only provide a brief description of the key knowledge resources stated in Table 2.

Case is a composition of Patient, Clinical judgment, History. Patient consists of patient identity such as his name, age and gender. Clinical judgment aggregates the objective and subjective observations and oral history taken from patient. The Case is abstracted to identify the Chief Complaint (CC). A list of Chief Complaint (CC) is provided in Habboushe's guide (2012).

Each CC has a set of Specified and Optional modifier associated with it. Table 3 shows the the modifiers for CC: *Altered mental status*.

Table 4 shows the modifier values and the abstracted modifier values for the modifier: *Respiratory*.

Table 3: The associated of the modifiers for CC: *Altered mental status*.

CC	Specified modifier	Optional modifier
Altered mental status	GCS Presenting complaint Bleeding Hypertension Pain Mechanism of injury	Hemodynamic Respiratory Temperature Mental health Blood glucose

Table 4: Modifier values for modifier: *Respiratory*.

Modifier	Modifier value	Abstracted modifier value
Respiratory	Cyanosis Single word speech Stridor Lethargic Confused	Severe
	Short of breath with mild exertion rest Speaking phrases Significant strider with airway protected	Moderate
	Mild short of breath Rapid breathing No obvious work of breathing Able to speak in sentences Stridor without obstruction	Mild

Table 5: Set of norms for modifiers *GCS*, *Respiratory* and *Hemodynamic*.

Modifier	Set of norms	
	Value	Level
GCS	3 to 9	I
	10 to 13	II
	14 to 15	III
Respiratory	Severe	I
	Moderate	II
	Mild	III
Hemodynamic	Severe end organ hypo-perfusion	I
	Borderline perfusion	II
	Upper and lower end VS	III
	Normal VS	IV

Table 5 shows the set of norms and the triggered triage levels for three example modifiers: *GCS*, *Respiratory* and *Hemodynamic*. The terms that feature in the modifier values such as Severe, Mild, Hypo-perfusion, Borderline perfusion and so on will be structured and defined in the domain knowledge model.

5 SUMMARY AND DISCUSSION

The paper discusses the construction of the knowledge model to support the development of a triage decision-support system. We adapted and extended the generic CommonKADS Assessment task to structure the triage decision-support task knowledge model, which consists of six inference models and one non-cognitive action. They are: Specification, Abstraction, Sorting, Evaluation, Verification, Matching and Fetching. The Sorting and Verification are new inferences that are introduced in this task, which do not exist in the generic Assessment task. The introduction of these two inferences is to support the nature of triage decision making.

Since we adapted a top-down modelling approach to engineer our task model, the validation of the task model involved checking with the domain experts on the concretisation of the abstract terms, the inference methods and the ordering, and their input and output data and/ or knowledge resources. In our future work, we will construct the triage purposive domain knowledge model informed by this task knowledge model.

ACKNOWLEDGEMENTS

This work was supported by Ministry of Education Exploratory Research Grant Scheme, Malaysia. We would like to thank Mohammad Hafidz bin Rahmat for his cooperation during our initial study.

REFERENCES

Angele. J., Fensel. D., Landes. D., Studer. R., 1998. Developing Knowledge-Based Systems with MIKE. *Automated Software Engineering,* October.5(4).

Annamalai, M., Sterling, L., 2003. *Guidelines for Constructing Reusable Domain Ontologies.* Proceeding of the AAMAS'03 Conference Workshop on Ontologies in Agent Systems, CEUR Series, Vol. 73.

Annamalai, 2006. *Modelling Knowledge for Scientific Collaboration on the Semantic Web.* Ph.D Thesis, University of Melbourne.

Chandrasekaran, B., 1986. Generic Tasks in Knowledge-based Reasoning: High-level Building Blocks for Expert System Design. *IEEE Expert 1,* Volume 3, pp. 23-30.

Considine, J., Botti, M., Thomas, S., 2007. Do Knowledge and Experince Have Specific Roles in Triage Decision-making?. *The Society for Academic Emergency Medicine,* pp. 722-726.

Gennari, J. H., Musen, M. A., Fergerson, R. W., grosso, W. E., Crubezy, M., Eriksson, H., Noy, N. F., Tu, S. T., 2003. The Evolution of Protégé: An Environment for Knowledge-based Systems Development. *International Journal of Human-Computer Studies,* January, 58(1), p. 89–123.

Gerdtz, M. F., Considine, J., Sands, N., Stewart, C. J., Crellin, D., Pollock, W. E., et al, 2009. *Emergency Triage Education Kit,* Canberra City.

Habboushe, J., 2012. *The Basics of Emergency Medicine, A Chief Complaint Based Guide.* s.l.:EMRA Publications.

Halim, S., Annamalai, M., Ahmad, M. S., Ahmad, R., 2011. A Conceptualisation of an Agent-Oriented Triage Decision Support System. In: *Knowledge Technology.* s.l.:Springer Berlin Heidelberg, pp.36-46.

Kingston, J., 2007. *Multi-perspective Modeling for Knowledge Management and Knowledge Engineering.* Ph. D Thesis. University of Edinburgh.

LeVasseur, S., Considine, J., Charles, A., Castle, C., Villanueva, E., 2001. *Consistency of Triage in Victoria's Emergency Departments: Consistency of Triage Report.* Monash Institute of Health Services Research.

Lindow, K., Heimann, O., Adolphy, S., Hayka, H., Stark, R., 2013. Decision-Making Support for Sustainable Product Development. In: M. S. R. Abramovici, ed. *Smart Product Engineering.* Bochum, Germany: Springer Berlin Heidelberg, pp. 979-988.

Marcus, S. e., 1988. *Automatic Knowledge Acquisition for Expert Systems.* Boston: Kluwer.

Murray, M., Bullard, M., Grafstein, E. , 2004. *Revisions to the Canadian Emergency Department Triage and Acuity Scale Implementation Guidelines.* CJEM.

Schreiber. A. T., Wielinga. B., Breuker. J., eds., 1993. *KADS: A Principled Approach to Knowledge-based System Development.* 11 ed. London: Academic Press.

Schreiber. G., Akkermans. H., Anjewierden. A., de Hoog. R., Shadbolt. N., de Velve. W. V., Wielinga. B., 2000. *Knowledg Engineering and Management: The CommonKADS Methodology.* Cambridge: The MIT Press.

Tarta, A. M., 2004. Task Modeling in Systems Design. *Informatica,* XLIX(2), pp. 37-44.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS