Discovering Expected Activities in Medical Context Scientific Databases

Daniela D'Auria and Fabio Persia
Dip. di Ingegneria Elettrica e Tecnologie dell'Informazione, University of Naples "Federico II", Naples, Italy

Keywords: Activity Detection, Scientific Databases, Cricothyrotomy Simulator, Medical Simulator.

Abstract: Reasoning with temporal data has attracted the attention of many researchers from different backgrounds including artificial intelligence, database management, computational linguistics and biomedical informatics. More specifically, activity detection is a very important problem in a wide variety of application domains such as video surveillance, cyber security, fault detection, but also clinical research. Thus, in this paper we present a prototype architecture designed and developed for activity detection in the medical context. In more detail, we first acquire data in real time from a *cricothyrotomy simulator*, when used by medical doctors, then we store the acquired data into a *scientific database* and finally we use an *Activity Detection Engine* for finding expected activities, corresponding to specific performances obtained by the medical doctors when using the simulator. Some preliminary experiments using real data show the approach efficiency and effectiveness. Eventually, we also received positive feedbacks by the medical personnel who used our prototype.

1 INTRODUCTION

Reasoning techniques are very essential in many application domains, such as video surveillance, cyber security, fault detection, fraud detection and in clinical domain, as well. In all cases, *temporal information* is crucial. For instance, for what the clinical research concerns, investigating disease progression is practical only by definition of a time line; otherwise, possible causes of a clinical condition have to be found by referring to a patient's past clinical history. In (Zhou and Hripcsak, 2007), the basic concepts of temporal representation in the medical domain have been described in order to include: category of time (natural, conventional, logical), structure of time (line, branch, circular, parallel), instant of time vs. interval, and, absolute time vs. relative time. Anyway, this is still a challenging and active subject of research. The main goal of (Safari and Patrick, 2013) consists in creating a special purpose query language for clinical data analytics (CliniDAL) to place in any clinical information system (CIS) and answer any answerable question from the CIS. In more detail, a category scheme of five classes of increasing complexity, including point-of-care retrieval queries, descriptive statistics, statistical hypothesis testing, complex hypotheses of scientific studies and semantic record retrieval have been designed to capture the scope encompassed by CliniDAL's objectives (Patrick and Cheng.Y., 2013). However, a review of temporal query languages reflects that the importance of time has led to the development of custom temporal management solutions, which are mostly built to extend relational database systems (for instance, T4SQL (Combi et al., 2007)). Many efforts in the relational database field have been conducted for developing expressive temporal query languages; nevertheless, they still suffer from two issues: firstly, they are only applicable to structural relational databases; secondly, it is difficult for hospital staff with poor IT skills to apply them. On the other hand, in most ontology based approaches composing queries can be difficult due to a complex underlying model representation and lack of expressivity.

In other contexts, such as video surveillance, cyber security and fault detection, the reasoning techniques using temporal information are broadly used for *activity detection*. Thus, several researchers have studied how to search for specifically defined patterns of normal/abnormal activities (Hongeng and Nevatia, 2001). Vaswani et al. (Vaswani et al., 2005) study how HMMs can be used to recognize complex activities, while Brand et al. (Brand et al., 1997) and Oliver et al. (Oliver et al., 2002) use coupled HMMs. Hamid et al. (Hamid et al., 2003) use Dynamic Bayesian networks (DBNs) to capture causal relationships between observations and hidden states. Albanese et al. (Albanese et al., 2007) developed a stochastic automaton-based language to detect activities in video, while Cuntoor et al. (Cuntoor et al.,

D'Auria D. and Persia F..
Discovering Expected Activities in Medical Context Scientific Databases.
DOI: 10.5220/0005146504460453
In *Proceedings of 3rd International Conference on Data Management Technologies and Applications* (KomIS-2014), pages 446-453
ISBN: 978-989-758-035-2

2008) presented an HMM-based algorithm. In contrast, (Albanese et al., 2014; Albanese et al., 2011) start with a set A of *activity models* (corresponding to innocuous/dangerous activities) and find observation sequences that are not sufficiently explained by the models in A. Such unexplained sequences reflect activity occurrences that differ from the application's expectations.

Other relevant works exploiting an *events sequence* definition are (Boselli et al., 2014a) and (Boselli et al., 2014b): in particular, (Boselli et al., 2014a) automatically identify cleansing activities, namely a sequence of actions able to cleanse a dirty dataset, which today are often developed manually by domain-experts, while (Boselli et al., 2014b) describe how a model based cleansing framework is extended to address integration activities as well.

In this paper, we present a prototype architecture designed and developed for activity detection in the medical context. The context of use is very concrete and important, as it is represented by a *cricothyrotomy simulator* built by the BioRobotics Laboratory of the University of Washington, Seattle (USA) (White et al., 2014; White et al., 2013; White et al., 2012a). Such a simulator is very relevant in the surgical robotic field, as it is very useful for helping medical doctors when they are performing a cricothyrotomy. Our main aim consists in making the medical doctors able to have a very fast feedback about their performances when using the simulator, that is very essential for this kind of applications. In order to do that, we first acquire data in real time from the simulator, when used by medical doctors, then we store the acquired data into a *scientific database* and finally we use an *Activity Detection Engine* for finding expected activities, corresponding to specific performances obtained by the medical doctors when using the simulator. Thus, we model the expected activities with *stochastic automata* (Albanese et al., 2014; Albanese et al., 2011) and exploit the activity detection algorithms presented by (Albanese et al., 2013), as temporal information is essential.

The paper is organized as in the following. Section 2 briefly describes the context of use of our prototype architecture, which is the *cricothyrotomy simulator* designed by the University of Washington, Seattle. Section 3 shows the model used for activity detection, while Section 4 describes the architecture of the developed prototype. Section 5 presents some preliminary experiments using real data. Eventually, Section 6 discusses some conclusions and future work.

2 CONTEXT OF USE: A CRICOTHYROTOMY SIMULATOR

Modern airway protocols involve many techniques to restore ventilation including bag-mask-ventilation, placement of a laryngeal mask airway, and intubation with or without videolaryngoscope. In cases where conservative measures fail or when contraindicated, the only methods remaining to re-establish ventilation may be surgical. In the developing world where devices such as the videolaryngoscope may not be available, accurate knowledge and training in the creation of a surgical airway may have a significant impact on patient outcomes.

A *cricothyrotomy* is a life-saving procedure performed when an airway cannot be established through less invasive techniques. Although performing such a procedure seems relatively straightforward, studies have shown that those performed in the pre-hospital setting were mostly unsuccessful (Wang et al., 2011). A review of 54 emergency cricothyrotomies found that the majority of the procedures performed in the field were unsuccessful or resulted in complications (King et al., 2012). A military team identified gap areas in the training of cricothyrotomy in emergency situations; these included lack of anatomical knowledge including *hands on* palpation exercises, poor anatomy in medical mannequins, and non-standard techniques (Bennett et al., 2011).

Most of the unsuccessful attempts were due to inaccurate placement, and incorrectly identifying anatomy. If the anatomy is not properly identified, it is unlikely that the procedure will be successful. Further, a large review of emergency airway cases found that emergency cricothyrotomies performed by anesthesiologists were successful in only 36% of instances (Cook et al., 2011). Although many reports suggest that the success rate of surgical airway placement is low, publications from advanced centers with extensive training for airway protocols including simulation show that pre-hospital cricothyrotomy success rates can be as high as 91% (Warner et al., 2009). Studies such as this suggest that with adequate training, the success rate of cricothyrotomy can be dramatically improved. Thus, an improved method of training needs to be provided for this rare, but life-saving procedure.

For such reasons, the BioRobotics Laboratory of the University of Washington, Seattle (USA) developed a low-cost cricothyrotomy simulator (White et al., 2014; White et al., 2013; White et al., 2012a) from readily available components that is equipped with inexpensive sensors. The simulator emphasizes the palpation and the correct identification of anterior cervical anatomy and has the ability to record the contact location of instruments on the trachea model during the full duration of the simulated procedure.

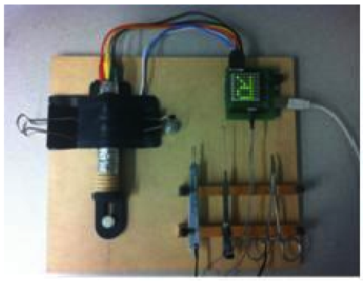
2.1 Simulator Design

The trachea model is disposable and is replaced after each procedure. To minimize costs, the trachea is made of cardboard with fixed size and dimension according to the trachea of an average adult. Foam strips were cut for the cartilaginous tracheal rings and were attached on the trachea with appropriate spacing. The *thyroid* and *cricoid* cartilages are permanent parts made of ABS plastic, using a 3D printer. These components are fixed onto a wooden base, and firmly support the trachea model. Conductive foils are used as low-cost sensors to detect six critical landmarks (identified by A-F letters) that providers might contact during the procedure. The conductive foils cover landmarks on the trachea model. Only one of these six landmarks, the cricothyroid membrane itself, is the correct area to contact and make an opening. Other landmarks like the posterior tracheal wall and lateral locations into the tracheoesphageal grooves should be avoided during the procedure.

The model is fitted with an *Arduino Uno* microcontroller board based on the Atmel Atmega 328 microprocessor with a mounted 8x8 LED matrix-based display for user interface capability. The microcontroller records the contact data of the instruments (scalpel, tracheal hook, and hemostat) onto the six conductive foils, as each instrument is wired. During the procedure, when a closed circuit is detected between the instrument and a patch of foil, the event is recorded and labeled with the time in milliseconds. Breaking contact is similarly recorded. Traditional matrix scanning techniques are used by the microcontroller to detect connections between the instruments and the foil patches. A minimum time (20 milliseconds) between contacts was used to debounce the inputs to the microcontroller. The resulting data was later low-pass filtered with a cutoff frequency of 10 Hz in accordance with general human reaction time.

The simulator's design is shown in Figure 1.

Moreover, the design was optimized for materials that are low-cost, widely available and simple to assemble. The total cost of the simulator was less than $50, which has a lower price compared to existing commercial simulators.



(a) *Trachea model is covered with inner bicycle tube as human skin.*

(b) *Six different landmarks represented by conductive foils and tools (scalpel, hook and forceps) are connected to microcontroller for data collection.*

Figure 1: Low-cost cricothyrotomy simulator.

2.2 How to Use the Simulator

Medical doctors who want to use the simulator are firstly forced to watch a video tutorial published by the New England Journal of Medicine (James and Pacheco-Fowler, 2008; White et al., 2012b). After watching the instructional materials, they are allowed to perform the procedure on the simulator following the instructions below:

- Step 1: Palpate the cricothyroid membrane. Immobilize the larynx with the non-dominant hand and perform the procedure with the dominant hand.
- Step 2: Incise the skin (bicycle inner tube) vertically after palpating the cricothyroid membrane.
- Step 3: Incise the cricothyroid membrane on trachea model horizontally (1 cm length).
- Step 4: Insert the tracheal hook into cricoid cartilage.
- Step 5: Insert the hemostat and to expand the airway opening vertically and horizontally.
- Step 6: Insert the endotracheal tube.

As these steps were performed, all the data (contact locations on trachea model, instruments information, contact duration and total time) were recorded by the microcontroller; procedures were also video-recorded for analysis.

Thus, after the description of the used *cricothyrotomy simulator*, the importance of temporal information is definitely clear. The following sections describe how our prototype has been designed and developed for helping medical doctors when they are using the simulator.

3 MODELING EXPECTED ACTIVITIES

This section describes the model that we have defined in order to derive a formal definition of *Expected Activity* for medical context. We use the temporal probabilistic graph proposed by (Albanese et al., 2014; Albanese et al., 2011), so that the elapsed time between observations also plays a role in defining whether a sequence of observations constitutes an activity, differently from what happens in other models, such as Hidden Markov Chains. We assume the existence of a finite set S of *action symbols*, corresponding to atomic events that can be detected by the *Arduino microcontroller board*, as described in Section 2.

3.1 Basic Definitions

An *Expected Activity* is a labeled directed graph A= (V ,E, δ, ρ) where: (i) V is a finite set of nodes labeled with action symbols from S; (ii) $E \subseteq V \times V$ is a set of edges; (iii) $\delta : E \to \mathbb{N}^+$ associates with each edge $\langle v_i, v_j \rangle$ an upper bound of time that can elapse between v_i and v_j; (iv) $\rho : E \to (0,1)$ is a function that associates a probability distribution with the outgoing edges of each node, i.e. $\forall v \in V \sum_{\langle v,v' \rangle \in E} \delta(\langle v, v' \rangle) = 1$; (v) there exists an initial node I in the activity definition, i.e. $\{v \in V \mid \nexists\, v' \in V \text{ s.t. } \langle v', v \rangle \in E\} \neq \emptyset$; (vi) there exists a final node F in the activity definition, i.e. $\{v \in V \mid \nexists\, v' \in V \text{ s.t. } \langle v, v' \rangle \in E\} \neq \emptyset$.

We assume the existence of a finite set S of *action symbols* representing particular interactions (for instance, *Pad C touched with SCALPEL, PAD C released*) between medical doctors and the simulator. Figure 2 shows an expected activity model representing simple interactions between a medical doctor and the cricothyrotomy simulator.

Then, we define an *instance* of an expected activity as a specific path in A from the initial node to the end node.

An *instance* of an *Expected Activity* (V, E, δ, ρ) is a finite sequence $\langle v_1, \ldots, v_m \rangle$ of nodes in V such that: (i) $\langle v_i, v_{i+1} \rangle \in E$ for $1 < i < m$; (ii) $\{v \mid \langle v, v_1 \rangle \in E\} = 0$, i.e. v_1 is the start node I; (iii) $\{v \mid \langle v_m, v \rangle \in E\} = 0$, i.e. v_m is the final node F. The probability of the instance is $\prod_{i=1}^{m-1} \rho(\langle v_i, v_{i+1} \rangle)$.

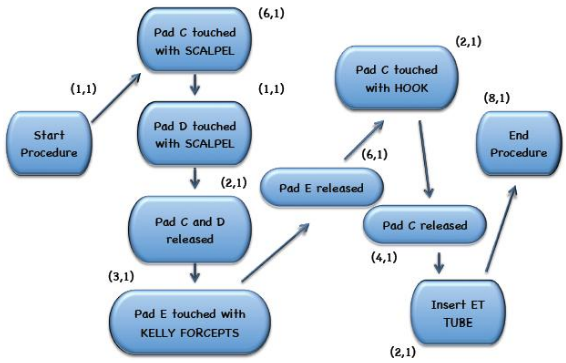


Figure 2: An example of Expected Activity Model.

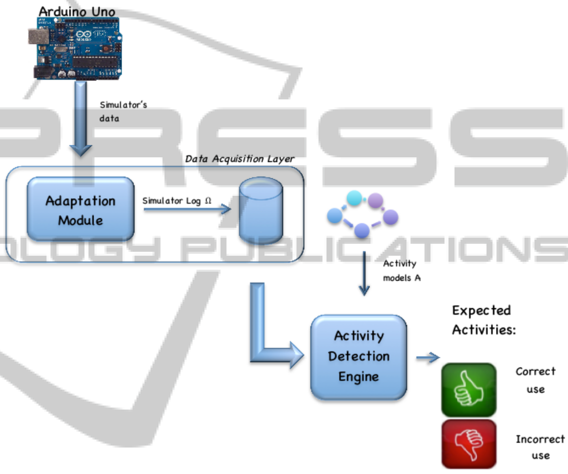


Figure 3: System Architecture.

We work with sequences of time-stamped events. Let us assume that the number of observable events in our domain is finite, each event can then be associated to a different action symbol in the set S. We define an *observed event* as a pair $\omega = (s, ts)$, where $\omega.s$ is the action symbol associated to the event and $\omega.ts$ is the time stamp at which s was observed.

We call a *Simulator Log* Ω a finite sequence of log entries ω_i.

Now, we are in the position of defining the concept of *Activity Occurrence*.

Let Ω be a *Simulator Log* and A=(V, E, δ, ρ) an Expected Activity. An *occurrence* of A in Ω is a sequence $\langle(\omega_1, v_1)\ldots(\omega_m, v_m)\rangle$ where: (i) $\langle \omega_1, \ldots, \omega_m \rangle$ is a subsequence of Ω such as $\omega_i = (\omega_i.ts, \omega_i.s)$, $\omega.s$ being an action symbol from S and $\omega.ts$ the associated time-stamp; (ii) $\langle v_1, \ldots, v_m \rangle$ is an instance of A; (iii) $v_i = \omega_i.s$ for $1 < \ i < \ m$[1]; (iv) $\omega_{i+1}.ts - \omega_i.ts \leq$

[1] v_i refers both to the node v_i in A and the action symbol s_i labeling it

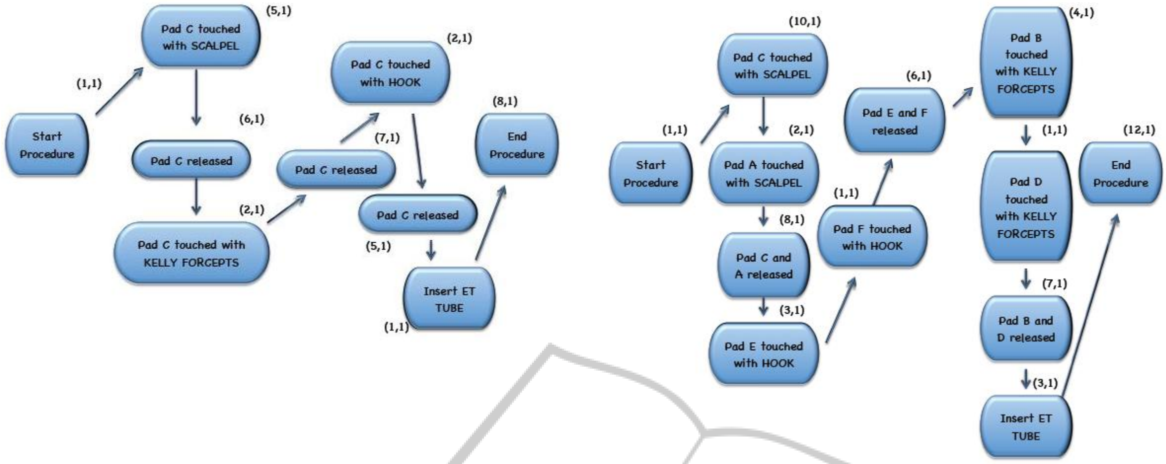


Figure 4: Expected Activity Models.

$\delta(\langle v_i, v_{i+1}\rangle)$ for $1 < i < m$.

The probability $p(o)$ of the occurrence o should be the probability of the instance $\langle v_1, \ldots, v_m\rangle$. Of course, shorter activities usually have higher probabilities. Therefore, since we compare occurrences of different activity models despite their different lengths, we introduce the relative probability $p^*(o) = p(o)/p(max)$. When computing $p^*(o)$ for a given occurrence o, we consider $p(max)$ as the highest probability of any instance of A when ignoring each instance's self-loops.

Thus, once we have given the previous formal definitions for defining our *expected activity model*, we can describe the proposed architecture for finding expected activities in medical context scientific databases in section 4.

4 THE PROPOSED ARCHITECTURE

The theoretical model has been exploited to develop a framework for the detection of expected activities in medical context scientific databases. The structure of the system is based on a modular architecture, as shown in Figure 3, which allows the medical doctors to get a very fast feedback about their performances when using the simulator.

The following subsections describe the single components of the overall system architecture.

4.1 The Arduino Microcontroller Board

As also mentioned in section 2, the *Arduino microcontroller board* allows us to capture in real time the contact data of the instruments (scalpel, tracheal hook, and hemostat) from six different landmarks of the simulator. In such a way, this component records the series of time-stamped events, corresponding to the medical doctors' interactions with the simulator. In more detail, events are defined as the start and end times of contacts between specific instruments and surfaces on the anatomical model. Other types of events are defined in terms of readings from different sensor types. Thus, events are represented by a series of symbols (ASCII characters). An excerpt of the captured data is shown in Figure 5. Data are encoded as follows:

- The first single digit number indicates the instrument (1 means *Scalpel*, 2 *Hemostat* and 3 *Tracheal Hook*).
- The character indicates which foil patch is touched: upper-case for making contact and lower-case for breaking contact. In more detail, A means *Posterior tracheal wall*, B the *Right lateral trachea and cricothyroid membrane*, C the *Midline cricothyroid membrane (correct placement of incision)*, D the *Left lateral trachea and cricothyroid membrane*, E the *Cricoid cartilage* and F the *Cartilaginous ring of lower tracheal wall*.
- The last number is the time in milliseconds.

Then, the data captured in this way represent the input of the *Data Acquisition* component.

4.2 The Data Acquisition Component

The *Data Acquisition* component includes an *Adaptation Module* that converts the data captured using the *Arduino* in a format suitable to the detection framework (i.e. the *Simulator Log*): it also saves them into a scientific database, which is able to store personal information about the medical doctors who are using the simulator as well.

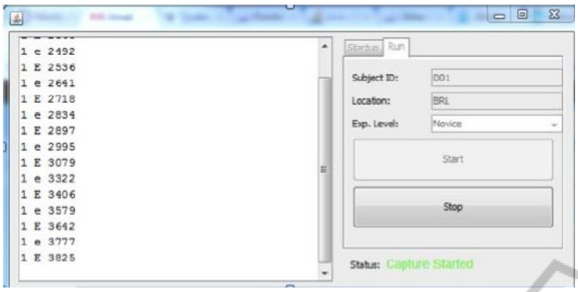


Figure 5: Data captured using the Arduino Microcontroller Board.

4.3 The Activity Detection Engine

The *Activity Detection Engine* takes as inputs time-stamped user data collected in the *Simulator Log* and a set of activity models to find the activity occurrences matching the known models. Such models have been previously defined by domain experts who have classified them in two different categories: the *good activities*, corresponding to a correct use of the simulator and the *bad activities*, corresponding to an incorrect use of the simulator. Figure 4 shows two model examples of a *good activity* (at left), corresponding to an excellent performance of the medical doctor and a *bad activity* (at right), corresponding to a very bad performance.

Expected activity occurrences in a data stream are efficiently detected using *tMagic* (Albanese et al., 2013), which allows to solve the problem of finding occurrences of high-level activity model in an observed data stream. As a matter of fact, they propose a data structure called *temporal multiactivity graph* to store multiple activities that need to be concurrently monitored, corresponding to our knowledge base of *good* and *bad* activities. They then define an index called *Temporal Multiactivity Graph Index Creation (tMAGIC)* that, based on this data structure, examines and links observations as they occur. Finally, they define an algorithm to solve the *evidence problem* that tries to find all occurrences of an activity (with probability over a threshold) within a given sequence of observations. In this way, we are able to find all the *Expected Activities* matching the activity models of our knowledge base in a certain sequence of observations, allowing medical doctors to get a very fast feedback about their performances when using the cricothyrotomy simulator.

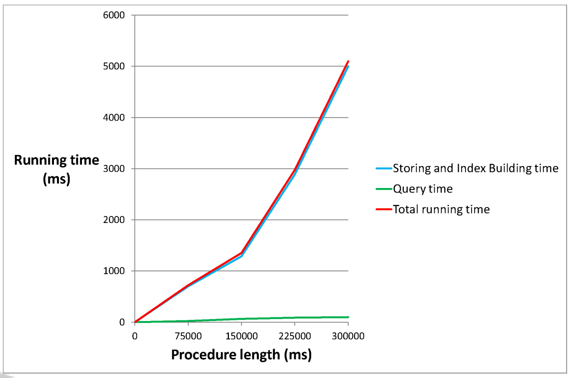


Figure 6: Framework Running Times.

5 PRELIMINARY EXPERIMENTAL RESULTS

This section shows a preliminary experimental evaluation of our framework. We present the experimental protocol for evaluating our framework in terms of *execution time scalability*, *detection accuracy* and *user satisfaction*.

5.1 Evaluating Execution Time

We decided to measure[2] the execution time of our framework for detecting *Expected Activities* in the worst case (at least one action symbol for each millisecond) when varying the length of the Simulator Log and using the previously defined set of known activity models. In more detail, the maximal length of Simulator Log considered has been 5 minutes, since a longer procedure would cause the death of the patient. The time for acquiring data using the *Arduino microcontroller* can be considered as negligible. Thus, the *Total Running Time* is given by the sum of the *Storing and Index Building Time* (higher value) and the *Query Time* (lower value), as shown in Figure 6. However, the obtained *Total Running Time* can be considered low even if we are considering the worst case.

5.2 Accuracy Results

100 medical doctors participated in a trial and used the simulator with our additional framework. The classic *Precision* and *Recall* metrics have been used

[2]All experiments presented in this Section were conducted on a machine running Mac OS X 10.9.1, and mounting a *2GHz Intel Core i7* processor with a *8 GB, 1600 MHz DDR3*.

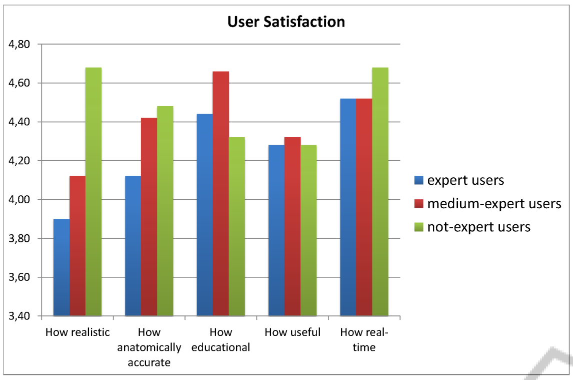


Figure 7: User Satisfaction.

to compute the accuracy (Albanese et al., 2014; Albanese et al., 2011), by comparing the Expected Activities discovered by our framework with a ground truth defined by experts who watched the recordings of the medical doctors'performances several times.

We use $\{A_i^a\}_{i\in[1,m]}$ to denote the Expected Activities returned by our framework and $\{A_j^h\}_{j\in[1,n]}$ to denote the activities flagged as expected by human annotators. Precision and recall were computed as follows:

$$P = \frac{|\{A_i^a | \exists A_j^h \ s.t. \ A_i^a \approx A_j^h\}|}{m} \quad (1)$$

and

$$R = \frac{|\{A_j^h | \exists A_i^a \ s.t. \ A_i^a \approx A_j^h\}|}{n} \quad (2)$$

We achieved an *average Precision* of *81%* and an *average Recall* of *98%*, that can be considered a very encouraging result.

5.3 User Satisfaction

After completing the procedure, the medical doctors, classified in three different categories (expert, medium-expert and not expert users), filled out a questionnaire to report their level of training, experience and their impressions of the simulator. Each subject was asked to answer 5 questions about the simulator (How realistic, How anatomically accurate, How educational, How useful, How real-time) using a 5-point Likert scale. As we can see in Figure 7, subjects (especially not-expert users) expressed positive opinions about their experiences with the simulator.

6 CONCLUSIONS AND FUTURE WORK

This work presented a framework for activity detection in the medical context. We started acquiring data from a *cricothyrotomy simulator*, when used by medical doctors and we then stored the captured data into a *scientific database*. Finally, we used some stable activity detection algorithms for discovering expected activities, corresponding to specific performances obtained by the medical doctors when using the simulator. Some preliminary experiments showed encouraging results concerning *efficiency*, *effectiveness* and *user satisfaction*.

Future work will be devoted to enlarge our experimentation and to plan to integrate the prototype in more complex and thorny applications by adding new functionalities and, if necessary, additional layers to the overall system architecture. For example, a potential application of this tool could consist in monitoring the performance of medical personnel in real time and detecting potential safety hazard *in advance*, for instance, using machine learning techniques and observations learned during the training of medical personnel. Moreover, data mining techniques could be used in an offline setting to analyze in detail the medical doctors'performances.

REFERENCES

Albanese, M., Molinaro, C., Persia, F., Picariello, A., and Subrahmanian, V. (2014). Discovering the top-k unexplained sequences in time-stamped observation data. *IEEE Transactions on Knowledge and Data Engineering*, 26(3):577–594.

Albanese, M., Molinaro, C., Persia, F., Picariello, A., and Subrahmanian, V. S. (2011). Finding "unexplained" activities in video. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1628–1634. AAAI Press.

Albanese, M., Moscato, V., Picariello, A., Subrahmanian, V. S., and Udrea, O. (2007). Detecting stochastically scheduled activities in video. In Veloso, M. M., editor, *IJCAI*, pages 1802–1807.

Albanese, M., Pugliese, A., and Subrahmanian, V. S. (2013). Fast activity detection: Indexing for temporal stochastic automaton-based activity models. *IEEE Trans. Knowl. Data Eng.*, 25(2):360–373.

Bennett, B., Cailteux-Zevallos, B., and Kotora, J. (2011). Cricothyroidotomy bottom-up training review: battlefield lessons learned. *Military Medicine*, 176(11):1311–9.

Boselli, R., Cesarini, M., Mercorio, F., and Mezzanzanica, M. (2014a). Planning meets data cleansing. In *Pro-*

ceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling.

Boselli, R., Cesarini, M., Mercorio, F., and Mezzanzanica, M. (2014b). A policy-based cleansing and integration framework for labour and healthcare data. In Holzinger, A. and Jurisica, I., editors, *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, volume 8401 of *Lecture Notes in Computer Science*, pages 141–168. Springer Berlin Heidelberg.

Brand, M., Oliver, N., and Pentland, A. (1997). Coupled hidden markov models for complex action recognition. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pages 994–999, Washington, DC, USA. IEEE Computer Society.

Combi, C., Montanari, A., and Pozzi, G. (2007). The t4sql temporal query language. In Silva, M. J., Laender, A. H. F., Baeza-Yates, R. A., McGuinness, D. L., Olstad, B., Olsen, . H., and Falco, A. O., editors, *CIKM*, pages 193–202. ACM.

Cook, T., Woodall, N., and Frerk, C. (2011). Major complications of airway management in the uk: results of the fourth national audit project of the royal college of anaesthetists and the difficult airway society. part 1: anaesthesia. *Fourth National Audit Project*, 106(5):617–31.

Cuntoor, N., Yegnanarayana, B., and Chellappa, R. (2008). Activity modeling using event probability sequences. *Image Processing, IEEE Transactions on*, 17(4):594–607.

Hamid, R., Huang, Y., and Essa, I. (2003). Argmode - activity recognition using graphical models. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW '03. Conference on*, volume 4, pages 38–38.

Hongeng, S. and Nevatia, R. (2001). Multi-agent event recognition. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 84–91 vol.2.

James, H. and Pacheco-Fowler, V. (2008). Cricothyrotomy. *N Engl J Med*.

King, D., Ogilvie, M., Michailidou, M., Velmahos, G., Alam, H., deMoya, M., and Fikry, K. (2012). Fifty-four emergent cricothyroidotomies: are surgeons reluctant teachers? *Scand J Surg*, 101(1):13–5.

Oliver, N., Horvitz, E., and Garg, A. (2002). Layered representations for human activity recognition. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 3–8.

Patrick, J.D. Safari, L. and Cheng.Y. (2013). Knowledge discovery and knowledge reuse in clinical information systems. *Proc. The 10th IASTED International Conference on Biomedical Engineering (BioMed 2013)*.

Safari, L. and Patrick, J. (2013). A temporal model for clinical data analytics language. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 3218–3221.

Vaswani, N., Roy-Chowdhury, A., and Chellappa, R. (2005). "shape activity": a continuous-state hmm for moving/deforming shapes with application to abnormal activity detection. *Image Processing, IEEE Transactions on*, 14(10):1603–1616.

Wang, H., Mann, N., Mears, G., Jacobson, K., and Yealy, D. (2011). Out-of-hospital airway management in the united states. *Resuscitation*, 82(4):378–385.

Warner, K., Sharar, S., Copass, M., and Bulger, E. (2009). Prehospital management of the difficult airway: a prospective cohort study. *J Emerg Med*, 36(3):257–65.

White, L., Bly, R., D'Auria, D., Aghdasi, N., Bartell, P., Cheng, L., and Hannaford, B. (2013). Cricothyrotomy simulator with computational skill assessment for procedural skill training in the developing world. In *AAO-HNSF Annual Meeting and OTO Expo*.

White, L., Bly, R., D'Auria, D., Aghdasi, N., Bartell, P., Cheng, L., and Hannaford, B. (2014). Cricothyrotomy simulator with computational skill assessment for procedural skill training in the developing world. *Journal of Otolaryngology - Head and Neck Surgery*.

White, L., Bly, R., D'Auria, D., Bly, R., Aghdasi, N., Bartell, P., Jones, C., and Hannaford, B. (2012a). Cricothyrotomy simulator training for the developing word. In *2012 IEEE Global Humanitarian Technology*.

White, L., Bly, R., D'Auria, D., Bly, R., Aghdasi, N., Bartell, P., Jones, C., and Hannaford, B. (2012b). Global simulation training in healthcare.

Zhou, L. and Hripcsak, G. (2007). Methodological review: Temporal reasoning with medical data-a review with emphasis on medical natural language processing. *J. of Biomedical Informatics*, 40(2):183–202.