2012 IEEE 24th International Conference on Tools with Artificial Intelligence

User-Oriented Evaluation of Event-Based Decision Support Systems

Jens Pottebaum*, Alexander Artikis†, Robin Marterer* and Georgios Paliouras†
*Computer Application and Integration in Design and Planning (C.I.K.), University of Paderborn, Paderborn, Germany
Email: {pottebaum, marterer}@cik.upb.de
†Institute of Informatics & Telecommunications, NCSR "Demokritos", Athens, Greece
Email: {a.artikis, palioug}@iit.demokritos.gr

***Abstract*—Artificial Intelligence-based event recognition systems carry high potential for organisations to utilise their structured and unstructured data. The application of these systems as a backbone of decision support systems allows for effective and efficient information management. To sufficiently evaluate such kind of integrated systems recognising events for the benefit of decision makers, a holistic methodology is necessary. We propose a new methodology which complements existing approaches for technology-oriented verification and validation by user-oriented evaluation (user experience analysis). We illustrate the proposed methodology by evaluating EP-IRM, an event processing system for intelligent resource management. This case study shows that our methodology offers invaluable information about the performance and acceptance of an event-based decision support system.**

***Keywords*-event recognition; event pattern matching; event processing; evaluation; user experience; decision support systems**

I. Introduction

Today's organisations collect data in various structured and unstructured digital formats, but they cannot fully utilise these data to support their resource management. It is evident that the analysis and interpretation of the collected data need to be automated, in order for large data volumes to be transformed into operational knowledge. Events are particularly important pieces of knowledge, as they indicate activities of special significance within an organisation. Therefore, the processing and, in particular, the recognition of events, is of utmost importance.

Systems for event recognition ('event pattern matching', in the terminology of [1]) accept as input a stream of time-stamped simple, derived events (SDE). A SDE (or 'low-level event' [1], 'short-term activity') is the result of applying a computational derivation process to some other event, such as an event detected by a sensor [2]. Event recognition systems identify composite events (CE) of interest — collections of events that satisfy some pattern. The 'definition' of a CE (or 'high-level event', 'long-term activity') imposes temporal and, possibly, atemporal constraints on its subevents (or 'members'), that is, SDE or other CE.

Event recognition systems are increasingly based on Artificial Intelligence (AI) techniques. They adopt logic-based languages with a formal, declarative semantics to express CE definitions. The Chronicle Recognition System [3], for example, is a highly-efficient temporal reasoning system that has been successfully applied to cardiac monitoring, intrusion detection and mobility management in computer networks, and distributed diagnosis of web services.

To sufficiently evaluate event recognition systems for the benefit of decision makers, we present a methodology allowing for user-oriented evaluation. There are several approaches in the literature evaluating AI-based systems — however, very few studies include 'extrinsic' evaluation approaches (cp. [4]) like usability testing and especially the question of user acceptance (cp. [5] focused on medical applications of AI-based systems). Lessons learned are partially available in the format of industry papers. These types of paper often do not follow any documented methodology (see Magid et al. [6] as an example). They add value to the field of evaluation research but do not allow conclusions to be drawn or comparisons between applications of the technology to be made.

The demand for evaluation methodologies for event processing applications has not been addressed sufficiently in the past. A well-known approach is that of Rabinovich et al. [7] where evaluation is performed in three steps. However, although these researchers explicitly mention the possibility of an extension by user-oriented evaluation, they do not elaborate on it. We build upon this work to present an evaluation methodology: we extend [7] by adding generic methodological foundations from the field of scenario-based evaluation (cp. [8], [9]) and data model validation (cp. [10], [11]).

We illustrate the proposed methodology by evaluating EP-IRM, an event processing system for intelligent resource management (IRM) developed in the context of the PRONTO project[1]. In EP-IRM, data is constantly acquired, synchronised and aggregated from various types of sensors that are installed in the infrastructure of the end user (for example, fire brigade), and from various modes of interaction between the actors of the application in hand (for instance, fire brigade officers). The aggregated data is

[1]http://www.ict-pronto.org/

1082-3409/12 $26.00 © 2012 IEEE
DOI 10.1109/ICTAI.2012.30

analysed, and enhanced with spatial information, in order to extract current, 'low-level' information in the form of SDE. Then, event recognition techniques are applied on the SDE streams in order to recognise, in real-time, CE. Given a SDE stream concerning the interactions of rescue workers and climate sensor data, for instance, the current status and criticality of a rescue operation are automatically detected for the benefit of the operation manager, responsible for resource management. A user-friendly IRM component is used to support the decision making process at that level.

To evaluate EP-IRM, we interviewed members of the end user organisations in order to determine the perceived impact of this system in real-world applications. We present the outcomes of, and our experiences from, the interviews. The evaluation of EP-IRM shows that our methodology offers invaluable information about the performance and acceptance of an event-based decision support system.

The remainder of this paper is organised as follows. Section II presents EP-IRM. Section III presents our evaluation methodology, while in Section IV we illustrate the methodology by evaluating EP-IRM. Section V summarises the presented work and outlines directions for further research.

II. The case study: EP-IRM

EP-IRM seamlessly integrates various types of novel event processing components for real-time CE recognition given multiple sources of information, including various types of sensor and modes of actor interaction. EP-IRM has been deployed, in the context of the PRONTO project, in two application domains: emergency rescue operations (ERO) and city transport management (CTM).

Concerning ERO, EP-IRM has been used for supporting the operations of the Fire Department of Dortmund, Germany. Input for CE recognition is gathered during regular daily business — using fire detection systems and weather information services — as well as in exceptional situations, that is, during an operation. An emergency and its evolution are observed by smoke and gas detectors. The emergency response is monitored by GPS, fuel and water sensors mounted on the vehicles used in the response. The SDE detectors operating on these sensors send data to control centres. Furthermore, rescue officers perform reconnaissance actions and communicate results to command posts — commanders enter information about the environment, the emergency and the response into management support systems. The communication channel and the user interaction with such systems are also used for SDE detection. The CE recognised on the SDE streams concern, among others, changes in the need for operations and the criticality of operations — such CE allow decision makers to perform goal-oriented improvisation and disposition of resources.

EP-IRM is based upon the principles of event-driven service-oriented architectures [12]. It is divided into subsystems containing one or more components — see Figure 1. All subsystems are connected through the Message-Oriented Middleware (MOM). Messages in the MOM represent events communicated between subsystems. EP-IRM includes components detecting SDE from audio, video, text, location, temperature, acceleration, and vehicle engine data. Following the publish-subscribe pattern, components may act as event producers or event consumers. For example, the SDE detection components consume raw events coming from sensors (microphones, cameras, GPS, etc) in order to produce SDE. The CE recognition component consumes SDE in order to produce CE. All events are logged in a semantic data store which is accessible via the MOM.

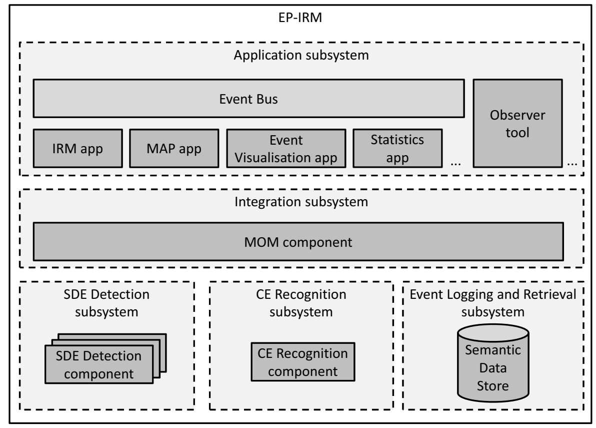

Figure 1. EP-IRM architecture.

The CE recognition component is a logic programming (Prolog) implementation of an Event Calculus dialect. The Event Calculus [13] is a logic programming language for representing and reasoning about events and their effects. The benefits of a logic programming approach to CE recognition are well-documented [14], [15]: such an approach has a formal, declarative semantics, is highly expressive, has direct routes to machine learning (see, for instance, [16]) for automatically constructing CE definitions, and has direct routes to reasoning under uncertainty (see, for example, [17]) for addressing the issues of noisy SDE streams and imprecise CE definitions. The use of the Event Calculus has additional advantages: the process of CE definition development is considerably facilitated, as the Event Calculus includes built-in rules for complex temporal representation and reasoning, including the formalisation of inertia. With the use of the Event Calculus one may develop intuitive, succinct CE definitions, facilitating the interaction between CE definition developer and domain expert, and allowing for code maintenance. A detailed account of the CE recognition component is given in [18].

User-oriented evaluation considerably depends on the visualisation capabilities of the event-based system. The representation of CE supports different user groups, use cases

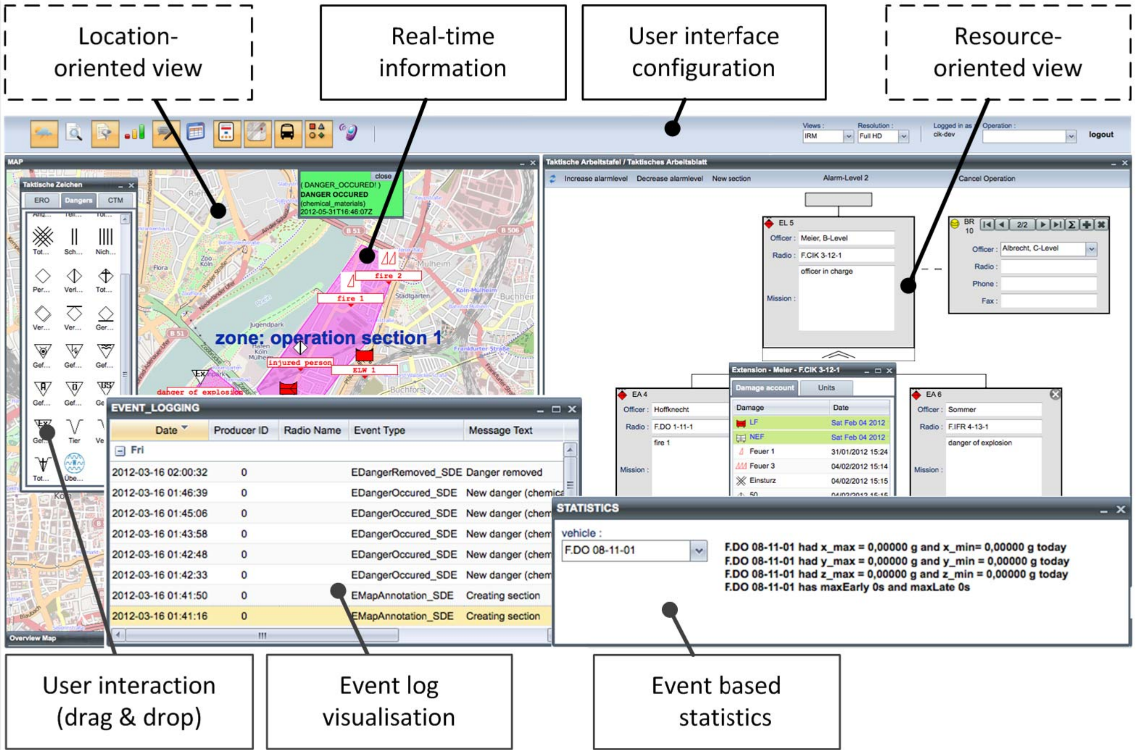

Figure 2. EP-IRM user interface for the real-time use cases.

and user interface requirements. EP-IRM addresses the concerned added value of event-based information throughout real-time decision support to post-operational analysis and debriefing based on event logs. The visualisation capabilities of EP-IRM are offered by the web applications (or 'apps')[2] that are part of the application subsystem. Each of the apps is represented by a window — see Figure 2. All windows can be resized and arranged arbitrarily. Assuming that the user interface is projected onto a wall, this realises a familiar handling for the end users, some of which work with paper sheets today. A user bar allows the configuration of apps and views. On the left hand side, each app can be switched on or off by a button. On the right hand side, different views for different user roles and operation context can be selected.

The Event Visualisation app, for example, enlists detected SDE and CE in real-time. The sorting and filtering functionality allows the adaptation of the visualisation to specific use cases. The Statistics app calculates and visualises event-based statistical information. The IRM app shows a logical view of the system's status. For example, in ERO the IRM app displays a tree view of the rescue operation command structure current at each time. Moreover, it shows the list of dangers of an operation, highlighting new ones in order to enable a commander to react to them. The MAP app presents a geo-based view of the system's status. For example, it displays positions of vehicles (buses and trams in CTM and fire engines in ERO) and other vehicle information or marked zones. Interactions between vehicles and spatial entities (such as a dangerous intersection in CTM or an emergency area in ERO) are prominently highlighted. All apps are connected to each other (via an event bus). For instance, when a new danger occurs and is shown in the IRM app, the user can click on it and its position as well as related information (such as a photo) will be presented by the MAP app. Apps act as event consumers — for instance, the Event Visualisation app consumes SDE and CE in order to display them in a time-line to the user. Moreover, apps act as event producers. Consider, for example, the case in which a commander creates an operational section using the IRM app in order to better manage the operation. In this case the event 'section created' will be produced and published in the MOM so that other components, such as the CE recognition component, may consume it. Similarly, when a commander drags & drops tactical symbols onto the map to denote a danger, the 'danger occurred' event is transmitted.

Functionality for use cases like debriefings and analysis is provided in EP-IRM by the 'Observer' component[3].

[2]For a clear distinction we use the term 'application' for applying event processing technology like EP-IRM to a context of use (such as ERO), and the term 'app' for a web application, that is, a HTTP-based software application which is accessible for users via web browsers.

[3]Noldus The Observer® XT with GeoViewer™ plugin.

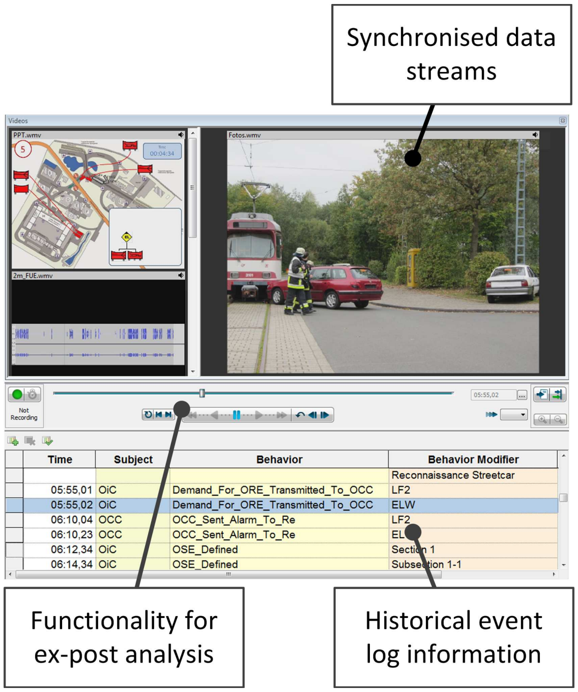

Figure 3. EP-IRM user interface for post-operation analysis.

This component presents historical event log information on a time-line — see Figure 3. Synchronised (sensor) data streams like audio, video and GPS signals are visualised. The log of recognised SDE and CE is used (a) to support the preparation and conduction of debriefing sessions after an operation, and (b) to enhance the authoring of scenario-based learning objects in learning management systems. More details on EP-IRM may be found at [19].

III. An Evaluation Methodology for Event-Based Decision Support Systems

The evaluation of event-based decision support systems targets technology-oriented and user-oriented questions. The core of these questions can be stated as 'does event recognition add significant value to existing IT-based support systems?' On the one hand, this question is related to quantitative characteristics like CE recognition times, as well as precision and recall. On the other hand, the added value of a system is defined by the impact which is perceived by users performing their regular tasks. These qualitative characteristics are related to questions of effectiveness (does the system fit to its intended purpose?), efficiency (does the system facilitate quick task conduction?) and user satisfaction (do users feel comfortable in using the system?) as the building blocks of usability [20]. Effectiveness is the foundation for efficiency — user satisfaction in general demands both effectiveness and efficiency.

Typically evaluation includes verification and validation (V&V) [21]. While verification correlates a technical system with its specification ('am I building the product right?'), validation concerns testing with respect to user needs and requirements ('am I building the right product?') [22]. The U.S. Department of Defense defines verification as the 'process of determining that a model implementation accurately represents the developer's conceptual description and specifications' and validation as the 'process of determining the degree to which a model is an accurate representation of the real-world from the perspective of the intended use of the model' [23].

The work of Rabinovich et al. [7] represents the most appropriate foundation for our research. We extend this methodology in order to sufficiently evaluate event-based decision support systems. We follow the steps below.

[Step 1] Verification with formal methods: analysis of the logical integrity of the artefacts of the event processing system. For example, with the use of answer-set programming techniques (such as [24]) one may prove various properties of a CE definition library expressed in the Event Calculus.

[Step 2] Static validation: design-time analysis of the modelled artefacts, such as the CE definition library. As an example, in ERO fire officers and EP-IRM developers collaboratively identified events in terms of SDE and constructed a CE definition library based on historical data and experiences. 20 CE types were defined based on 47 SDE types and corresponding data sources. Static validation checks whether (a) for each CE type all necessary subevents (SDE and CE) and event consumers are available, (b) consequences and (c) provenance are consistently modelled, and (d) cycles in the CE hierarchy are transparent for developers.

[Step 3] Dynamic validation: run-time observation of the event processing system. While static validation already tackles consequences and provenance between event types, dynamic validation checks the cascading trace from SDE to CE (that is, the forward trace), and vice versa from a CE to its members back to the initial SDE (backward trace).

When focusing on decision making in critical situations, the above procedure needs to be extended. Users (that is, decision makers and their staff) expect that the event processing system induces a perceptible impact on their task execution. The implementation of use cases has to support decision making processes and fulfil information quality requirements [25], [26]. The question whether this goal was reached cannot be answered by validation and verification. Therefore, we had to extend the evaluation methodology of [7] by adding the step below.

[Step 4] User-oriented evaluation: run-time analysis of the perceived impact of the event processing system on specific use cases and work processes.

According to several established approaches on requirements engineering, usability engineering and evaluation (for example, [27], [28], [29], [30]), we propose a scenario-driven approach for the user-oriented evaluation of event-based decision support systems. Scenarios are used as a

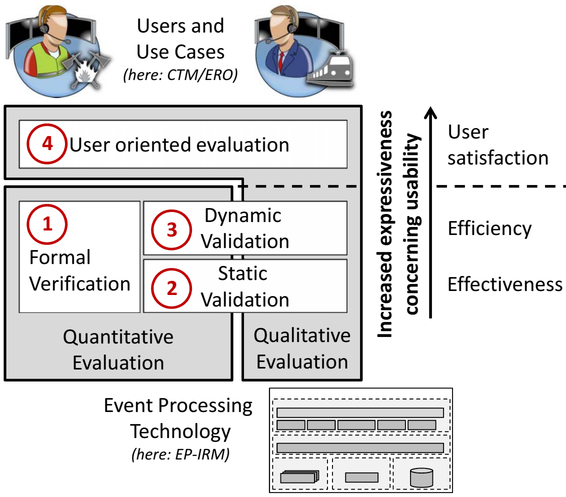

Figure 4. Evaluation methodology: integrating user- and technology-oriented methods.

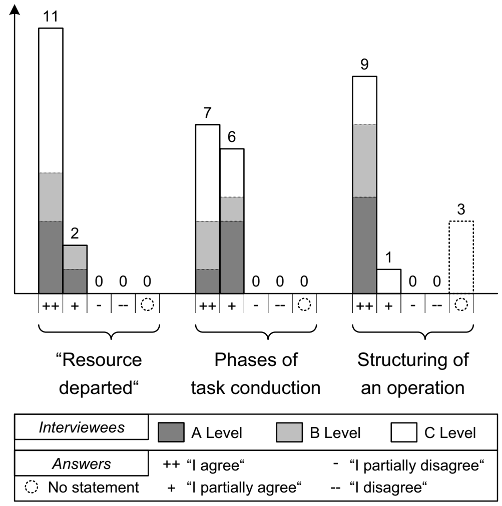

Figure 5. CE definition library validation. Agreement of fire officers to the statement: CE are modelled correctly.

reference to compare the behaviour of an event processing application with real event sequences and to correlate interviews with realistic contexts of use.

The extension of [7] leads to an integrated methodology for the evaluation of event-based decision support systems — see Figure 4. The focus of this paper is on the qualitative assessment of the impact of technology. In the following section, therefore, we illustrate the proposed extension of [7], that is, step 4, by presenting and discussing the results of the interviews on the user's perception of EP-IRM (see also [31]; for a quantitative assessment of EP-IRM see [18]).

IV. User-Oriented Evaluation of EP-IRM

For the definition of use scenarios and for the acquisition of corresponding data, 24 ERO training exercises were recorded and observed. We compared the real events within these exercises with our CE definition library. In each exercise, 22 fire fighters and officers responded to different types of emergency (car and tram accidents, fire in buildings, gas spreading in a train, and so on).

The event validation was extended by active practitioner involvement. Dedicated interviews were conducted with 13 fire officers individually; this group of interviewees was distinct from the event modelling group and the personnel that participated in training exercises. The fire officers command large scale operations (known as 'A level officers', 'Gold level officers' or 'strategic commanders'), medium size operations ('B level', 'Silver level' or 'tactical commanders') or small operations ('C level', 'Bronze level' or 'operational commanders'). After some introductory questions, we started every interview with a basic example. We identified an event type which (a) represents a CE, (b) is recognised as an 'event' by interviewees from a terminological perspective, and (c) is easy to understand for interviewees. These requirements were met by the CE 'resource departed': semantically, it states that the resource acknowledged an alarm, carries personnel and material, and leaves its current location to an emergency site. This CE already facilitates the discussion of the event-driven approach, but it is not a crucial factor for making new decisions. We call 'decisiveness' this attribute of a CE. While some of the CE directly represent decisions (for example, a task is assigned to a specific resource), others only hint at the need for a decision (for example, a new danger occurred), and a third category represents information which is used for decision making (for instance, a vehicle changing its location). In order to draw conclusions for different levels of decisiveness, two groups of CE were discussed:

- One concerning the phases of task conduction, that is, CE representing the workflow of an operation, focusing on a specific resource, and
- one concerning the logical structure of an operation, that is, the specification, change and deletion of structural, spatial and task-oriented relationships between resources.

The results of the validation of the CE definition library are shown in Figure 5 using four-level Likert scaling from full agreement (that is, the user expects an added value by using the system) to full disagreement. All CE were accepted by interviewees. These results confirm that the event types of the CE definition library support the *effectiveness* of an operation. Fire officers:

- Agree on the initial CE example 'resource departed' as described above (see the left chart of Figure 5).
- Miss CE representing the communication itself as part of the task conduction (see the middle chart of Figure

5). Consider, for example, a report acknowledging a task sent from the field to a command post via audio communication. Fire officers recognised the report as an event, but not its content. To define CE concerning task conduction, we had to represent the content of the report — the acknowledgement — and not the act of sending the report.
- Agree on the relevance of events concerning the management of the operation structure (see the right chart of Figure 5). Nevertheless, concerning the distinction of structural, spatial and task-oriented CE, nearly all fire officers preferred to use the recognised CE for explaining decisions after an operation, as opposed to using these CE for managing an on-going operation.

Furthermore, fire officers:

- Agree that the defined CE sufficiently represent the expected flow of an operation, as this is defined by the fire brigade guidelines and directives.
- Highlight the importance of CE representing delays and demands (that is, abnormal events with respect to resource availability).

The event types carry the innovation of EP-IRM on top of traditional decision support systems that do not allow for event processing[4].

One of the challenges of the the evaluation of EP-IRM had to do with the understanding and use of the term 'event'. As an example, nearly 70% of ERO interviewees reduced the scope of 'event' to an emergency (for example, dangers happening in a specific environment) — they did not correlate this term with the emergency *response* (for instance, forces starting fire fighting) which is prominent in the ERO CE definitions. We found it impossible to introduce a shared understanding of 'event', and accepted the terminology divergence in our interviews with end users. The researchers conducting the interviews had to ensure that statements made by end users were appropriately interpreted.

Targeting the contribution of EP-IRM to the *efficiency* of work processes, we associated each event type with the acceptable delay between event occurrence and its visualisation in EP-IRM (categorised as 'less than 10 sec', 'less than 60 sec' and 'more than 60 sec'). Additionally, for each event type we estimated the impact of recognition accuracy — precision and recall — on operations and the acceptance of EP-IRM. Figure 6 displays the outcome of 13 interviews with ERO end users with respect to these issues. Concerning recognition efficiency (see the left chart of Figure 6), practitioners correlate the acceptable CE recognition delay with:

- the processes in which the recognised CE are needed,
- available comparable technology — they compare, for example, CE recognition by EP-IRM to being informed about a CE by means analogue radio.

[4]As an example, see the white paper by Oracle presenting a case study on Emergency Response Resource Proximity/Location Tracking [32].

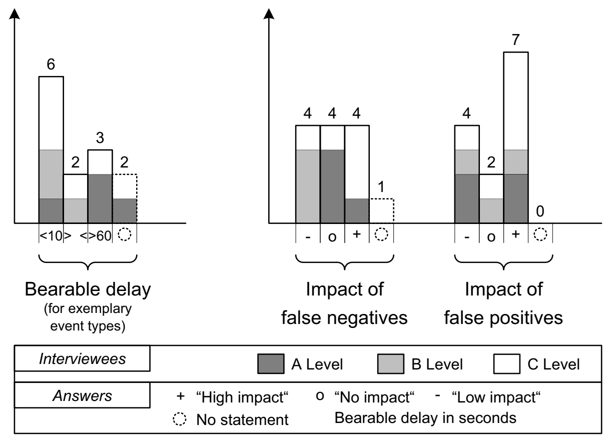

Figure 6. Acceptable delay and accuracy of CE recognition.

End users do not always think of milliseconds — they sometimes think about the processes before which a CE should be recognised. For example, the ERO 'demand for additional resources' CE should be recognised any time before the following reconnaissance. One aspect that we did not anticipate concerns the fact that users do not always have high performance requirements. For example, some rescue officers accepted delays up to one minute concerning the recognition of some CE as they would not be able to use this information (recognised CE) earlier. This is supported by the fact that briefings, debriefings and operating reports are highlighted as major use cases for EP-IRM in addition to real-time event recognition and visualisation.

Concerning recognition accuracy, our initial assumption of 100% precision and recall was challenged by users. A 95% accuracy is acceptable by most users. The middle and right charts of Figure 6 show the impact of recognition errors expected by users:

- The impact of false negatives is diverse. In some cases, such as when a 'resource departed' in order to participate in an emergency operation but this CE was not recognised, false negatives lead to an overestimation of the necessary resources, but have no negative influence on an emergency.
- The impact of false positives is much more critical up to 'not acceptable'. Only a few officers deviate from this judgement who would double-check information provided by an event processing system.

The main goal of the qualitative evaluation is to determine the *user satisfaction*. We correlate user satisfaction with the added value perceived by practitioners when using an event-based system. All fire officers answered questions targeting the added value offered by EP-IRM. For real-time decision support, as well as debriefing and training sessions, users assessed the innovation with respect to known

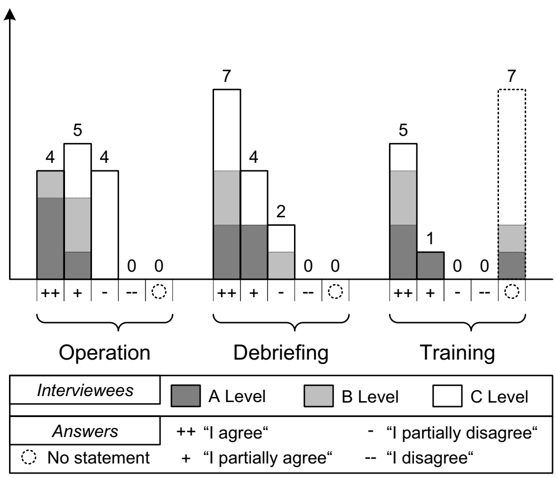

Figure 7. User satisfaction. Agreement of fire officers to the statement: EP-IRM adds value to an operation, debriefings, and trainings.

information management solutions. The results are shown in Figure 7. All interviewees acknowledged the potential of CE recognition in large-scale operations (see the left chart of Figure 7). These operations are characterised by a high number of influencing elements such as danger events, resources performing actions and communication, with complex relations. Interviewees stated that the recognised CE add significant value to domain-specific, state-of-the-art information management tools. Not surprisingly, higher-level officers saw increased added value than lower-level officers.

As expected (cp. [33]), another agreement was visible with respect to the usability of CE and SDE logs for debriefing and training purposes (see the middle and right charts of Figure 7). Interviewees used and positively rated the visualisation of CE and SDE on a time-line and the utilisation of CE as anchors to access specific parts of the CE logs and underlying data streams. Additionally, EP-IRM allows the utilisation of event logs for authoring of learning content. Concerning this use case, we interviewed 6 out of 13 fire officers. These officers brought in their experience in training exercises and courses. The event logs of EP-IRM allow the definition of training scenarios, authoring of learning objects and generation of scenarios for simulation platforms. Fire officers clearly confirmed the added value for this use case and actively added thoughts on further utilisation of event log information.

With respect to visualisation, end users benefit from, and often demand, explanation facilities from the EP-IRM. When various recognised CE were presented to the users, an explanation concerning CE recognition was required (‘drill down’) — what are the occurrences of the subevents of the CE that lead to the CE recognition? Such a feature is deemed necessary both for run-time and off-line use of the system.

V. Summary & Further Work

Existing methodologies for the evaluation of (AI-based) event recognition systems are limited to verification and validation, generic software qualities and specific requirements. To identify the added value of embedded technologies like event processing for the targeted work processes and tasks, additional work is necessary to assess human concerns and user acceptance. We presented a methodology that extends existing approaches on verification and validation of generic software systems, specific event processing systems, data model validation and usability testing. We argue that four steps are necessary for an evaluation: verification with formal methods, static and dynamic validation, and user-oriented evaluation.

We introduced a case study by applying an event recognition system to different application domains. We focused on user-oriented evaluation in order to highlight the benefits of this evaluation step in contrast to existing methodological approaches. Results like the perceived bearable delay of event processing, acceptance barriers and the added value for real work processes can only be measured by ‘extrinsic’ user-oriented methods. User acceptance needs to be measured to estimate the value of an integrated IT system: run-time analysis of the perceived impact adds significant value to this quality.

The proposed methodology for event-based decision support systems needs to be validated in additional case studies. EP-IRM is capable of serving other use cases and application domains. This facilitates comparative research broadening the results presented in this paper.

Acknowledgments

This work has been partially funded by EU, in the context of the PRONTO project (FP7-ICT 231738).

References

[1] D. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley, 2002.

[2] D. Luckham and R. Schulte, “Event processing glossary,” Event Processing Technical Society, 2008.

[3] C. Dousson and P. L. Maigat, “Chronicle recognition improvement using temporal focusing and hierarchisation,” in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2007, pp. 324–329.

[4] P. Chen, W. Ding, C. Bowes, and D. Brown, “Large-scale dependency knowledge acquisition and its extrinsic evaluation through word sense disambiguation,” in *Proceedings of ICTAI ’09*, 2009, pp. 314 –318.

[5] B. Kaplan, “Evaluating informatics applications clinical decision support systems literature review,” *International Journal of Medical Informatics*, vol. 64, no. 1, pp. 15 – 37, 2001.

[6] Y. Magid, G. Sharon, S. Arcushin, I. Ben-Harrush, and E. Rabinovich, "Industry experience with the ibm active middleware technology (amit) complex event processing engine," in *Proceedings of DEBS'10*, Cambridge, UK, 2010.

[7] E. Rabinovich, O. Etzion, S. Archushin, and S. Ruah, "Analyzing the behavior of event processing applications," in *Proceedings of DEBS'10*, J. Bacon, P. R. Pietzuch, J. Sventek, and U. Çetintemel, Eds. ACM, 2010, pp. 223–234.

[8] K. Weidenhaupt, K. Pohl, M. Jarke, and P. Haumer, "Scenarios in system development: Current practice," *IEEE Software*, vol. 15, no. 2, pp. 34–45, 1998.

[9] K. Go and J. M. Carroll, "The blind men and the elephant: views of scenario-based system design," *interactions*, vol. 11, no. 6, pp. 44–53, nov 2004.

[10] R. G. Sargent, "Verification and validation of simulation models," in *Proceedings of Winter Simulation Conference (WSC '05)*, 2005, pp. 130–143.

[11] J. Banks, J. p. Carson, and B. L. Nelson, *Discrete event system simulation*, 2nd ed., ser. Prentice-Hall international series in industrial and systems engineering. Upper Saddle River, NJ: Prentice-Hall, 1996.

[12] G. Mühl, L. Fiege, and P. R. Pietzuch, *Distributed event-based systems*. Springer, 2006.

[13] R. Kowalski and M. Sergot, "A logic-based calculus of events," *New Generation Computing*, vol. 4, no. 1, pp. 67–96, 1986.

[14] A. Paschke and M. Bichler, "Knowledge representation concepts for automated SLA management," *Decision Support Systems*, vol. 46, no. 1, pp. 187–205, 2008.

[15] D. Anicic, S. Rudolph, P. Fodor, and N. Stojanovic, "Real-time complex event recognition and reasoning — a logic programming approach," *Applied Artificial Intelligence*, vol. 26, no. 1–2, pp. 6–57, 2012.

[16] O. Ray, "Nonmonotonic abductive inductive learning," *Journal of Applied Logic*, vol. 7, no. 3, pp. 329–340, 2009.

[17] A. Kimmig, B. Demoen, L. D. Raedt, V. S. Costa, and R. Rocha, "On the implementation of the probabilistic logic programming language ProbLog," *Theory and Practice of Logic Programming*, vol. 11, pp. 235–262, 2011.

[18] A. Artikis, M. Sergot, and G. Paliouras, "Run-time composite event recognition," in *Proceedings of International Conference on Distributed Event-Based Systems (DEBS)*. ACM, 2012, pp. 69–80.

[19] A. Artikis, R. Marterer, J. Pottebaum, and G. Paliouras, "Event processing for intelligent resource management," in *Proceedings of European Conference on Artificial Intelligence (ECAI)*. IOS Press, 2012, pp. 943–948.

[20] ISO, "9241-11: Ergonomic requirements for office work with visual display terminals (vdts) – part 11: Guidance on usability," 1998.

[21] IEEE, "Std 1012-2004: Ieee standard for software verification and validation (revision of ieee std 1012-1998)," New York, NY, 2004.

[22] H. Partsch, *Requirements-Engineering systematisch - Modellbildung für softwaregestützte Systeme*. Berlin, Heidelberg, New York: Springer Verlag, 1998.

[23] U. D. of Defense, "Dod modeling and simulation (m&s) management: Directive," 2007.

[24] M. Gebser, O. Sabuncu, and T. Schaub, "Finite model computation via answer set programming," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2011, pp. 2626–2631.

[25] R. Gonzalez and N. Bharosa, "A framework linking information quality dimensions and coordination challenges during interagency crisis response," in *Proceedings of HICSS '09*, jan. 2009, pp. 1 –10.

[26] T. Friberg, S. Prödel, and R. Koch, "Analysis of information quality criteria in a crisis situation as a characteristic of complex situations," in *Proceedings of ICIQ'10*, Little Rock, November 2010.

[27] J. Goguen, "Requirements engineering as the reconciliation of technical and social issues," in *Requirements Engineering: Social and Technical Issues*. Academic Press, 1994, pp. 165–199.

[28] M. Rosson and J. Carroll, *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. San Francisco: Morgan Kaufmann, 2002.

[29] A. Sutcliffe, *User-Centered Requirements Engineering: Theory and Practice*. London: Springer, 2002.

[30] W. Dzida and R. Freitag, "Making use of scenarios for validating analysis and design," *IEEE Trans. Software Eng.*, vol. 24, no. 12, pp. 1182–1196, 1998.

[31] J. Pottebaum, *Optimierung des einsatzbezogenen Lernens durch Wissensidentifikation*, ser. Dissertation, University of Paderborn. Münster: MV-Verlag, 2012.

[32] O. P. Management and D. Teams, "Oracle complex event processing: Lightweight modular application event stream processing in the real world," Oracle Corporation, Oracle Parkway, USA, Tech. Rep., June 2009, an Oracle White Paper.

[33] M. Turoff, M. Chumer, B. Van de Walle, and X. Yao, "The design of a dynamic emergency response management information system (dermis)," *Journal of Information Technology Theory and Application (JITTA)*, vol. 5, no. 4, pp. 1–36, 2004.