# Event Definition for the Application of Event Processing to Intelligent Resource Management

**Jens Pottebaum**
University of Paderborn, C.I.K.
pottebaum@cik.upb.de

**Alexander Artikis**
NCSR Demokritos
a.artikis@iit.demokritos.gr

**Robin Marterer**
University of Paderborn, C.I.K.
marterer@cik.upb.de

**Georgios Paliouras**
NCSR Demokritos
paliourg@iit.demokritos.gr

**Rainer Koch**
University of Paderborn, C.I.K.
r.koch@cik.upb.de

## ABSTRACT

The application of event processing methods and systems carries high potential for the domain of crisis management and emergency response for different use cases and architectural aspects. This hypothesis is based on the general event based characteristics of the domain as well as former research approaches. Resource management represents a complex task for decision makers; therefore it is taken as a basic use case for this work. It builds up on foundations of resource management (use case and demand side) and event processing (technology and supply side). Methods and results are presented for the identification, definition and validation of events that happen in reality and corresponding event objects which are processed by information systems.

### Keywords

Event, event definition, event recognition, event processing, resource management, emergency response.

## INTRODUCTION

The domain of crisis management and emergency response has foundations in event driven task accomplishment. Disposition and improvisation as initial activities, but also continuous response and management efforts are highly driven by the occurrence of events. Therefore command and control procedures are focused on observation as a starting point for cyclic decision making processes (cp. OODA loop, Boyd, 1996). Observations are focused on things that happen in a(n emergency) situation. Actually this is the definition of an "event" used in this paper: It can be seen as a single point in time when something happens (cp. Allen and Hayes, 1985). As one very clear example for the impact of events, many directives for fire protection claim that as soon as human beings get endangered (this is an "event") rescue tasks have to get prioritized to the highest level (this is a reaction to that "event" in terms of resource management).

"Events" can be seen from many different perspectives. At first a distinction has to be made between an event (real world) and an event object (representation of an event in an IT system). The event "human beings endangered" might be communicated via radio from the field to the command post. Assuming a speech recognition system identifies the embedded event, it can store a corresponding "event object" in a database. Following this example, an event object is "an object that represents, encodes, or records an event, generally for the purpose of computer processing." (Luckham and Schulte, 2008). Such an event object could be sent as an electronic message to decision makers or it could be visualized by a GUI of an IT system. This distinction is very important for research dedicated to the promising transfer from real world analysis to event processing. The objectives are on the one hand a deeper understanding of the domain with the opportunity to use a natural way of describing this information, and on the other hand the preparation of further research on event driven systems

**Reviewing Statement:** This full paper has been fully double-blind peer reviewed for clarity, relevance, significance, validity and originality.

within this domain (Birkhäuser et al., 2010). While the community in the area of event processing is growing (cp. Etzion and Niblett, 2010), only a few results are visible in the specific domain of interest. Examples are presented by ter Mors et al. (2005), focusing on the event based optimisation of disaster plans, Wickler et al. (2007), addressing training support by flexible event based scenario generation, and Gonzalez (2009), combining distributed event simulations and multi-agent systems. All of them have in common that the challenge of correlating events and tasks is a main requirement for event driven information systems.

Another important basis is the temporal dimension of events: In some fields (cp. Luckham and Schulte, 2008) "events" are not restricted to a single point in time but might be durative. Although on the one hand this reduces clarity in the distinction of events (something happens) and states (something is true), on the other hand it allows to include research results that propose the term "event" as a synonym for incidents or public activities. This becomes significant for event processing as soon as the same semantics are applied to digital data, e. g., when data about an incident should be extracted from social media (cp. Edmonds et al., 2010). In addition events are at the same time either simple (something that can be observed) or complex (something that is based on more than one event). Event objects are either raw (directly detected) or derived (result of event processing) (simplified terminology based on Luckham and Schulte, 2008, Etzion and Niblett, 2010).

The application of event processing methods and systems carries high potential for the domain of crisis management and emergency response. Such applications have to implement a clear distinction between events and their representation as event objects. Literature research and observations in the field of fire protection showed that such a distinction is essential. Nevertheless only a few approaches exist that focus on this specific research topic. This paper presents results with a clear focus on event definition for the purpose of event processing. It builds upon foundations of resource management (use cases resp. demand side) and event processing (technology resp. supply side). Methods and results are presented for the identification, definition and validation of events and corresponding event object types. The presented results concern an event-driven service-oriented system (SOA) for crisis management and emergency response. Based on these results we draw conclusions for both event definition research and implementation of event processing components.

## FOUNDATIONS OF EVENT PROCESSING AND INTELLIGENT RESOURCE MANAGEMENT

The definition of a reasonable set of event types (representing a class of events correlated in terms of specific parameters, see Luckham and Schulte, 2008) is based on the assumption that it is not possible to model all possible events that might happen in an operation. This implies at first a research question including (a) demands and requirements based on the use case as well as (b) technological capabilities to be found in the event processing area: What kind of events should be detected by event processing systems to optimize information provision for supporting intelligent resource management?

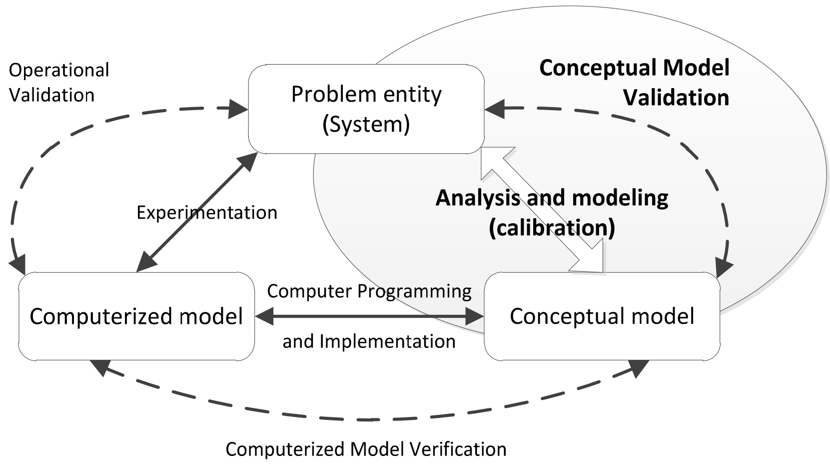


**Figure 1. Highlighting the scope of this paper within a simplified model of the dependencies between modeling, verification and validation of event models related to Sargent (2005) and Banks et al. (1996)**

The research presented in this paper answers this research question by providing a 'valid' subset of events. Based on literature on distributed event based simulation systems as one fundament of current event processing development the definition of valid models is an iterative process approaching the challenge from all given directions (cp. "calibration" in Figure 1 resp. Banks, Carson and Nelson, 1996). Validation in this context means "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" having strong interdependencies with verification as "The process of determining that a model implementation accurately represents the developer's conceptual description and specifications." (U.S. Department of Defense, 2007; cp. IEEE Std 1012-2004, Banks et al., 1996, Sargent, 2005, and others).

Besides taking into account demand and supply side it is important to involve future applications of the results. Event processing and, even more, event recognition depends on sensors providing data. Thus the question of sensor availability has to be taken into account early. This leads to a combination of "bottom-up" (Initial question: What simple event can be recognized by available sensors?) and "top-down" (What complex event would lead to a decision in the use case?) approaches.

### Intelligent Resource Management

The core of emergency management is to coordinate resources for effective and quick danger averting to rescue or protect entities (i. e., human beings, animals, buildings, etc.) within the environment. The availability of resources is essential for decision making processes (cp. Birkhäuser, Pottebaum and Koch, 2009). Thereby the term "resource" summarizes everything that is related to emergency response and management, e. g., personnel, vehicles and material as well as pre-defined units. The challenge for commanders is to define adequate tasks, to build up effective tactical units from resources and to bring them into an operation by handing over appropriate tasks within a spatial area of responsibility. Tasks are directly related to dangers or support danger-averting tasks; both for dangers and task execution the spatial dimension is important. All of these elements define the areas of interests for event definition (see figure 2): By active or passive participation in an emergency or in emergency response they generate events or are affected by events Therefore they have to be modeled for defining the basis of event recognition (cp. sub-ontologies defined in Pottebaum et al., 2010).

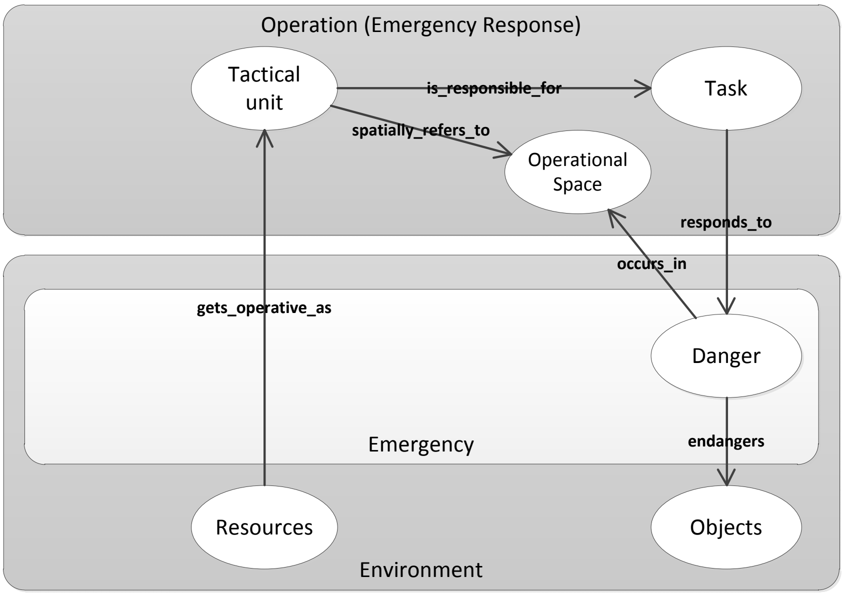


**Figure 2. Areas of interests for event definition**

The demand for knowledge about events arises within the decision-making processes of emergency management. Commanders follow a cyclic process for this task[1]: Iteratively the situation has to be "observed", based on the gathered facts and hints commanders have to "orient" and "decide" followed by "acting" in terms of sending commands to resources. Thereby the complete operational picture has to be taken into account: Both the "own situation" (emergency response and management) as well as the "foreign situation" (emergency and environment) develop dynamically having potentially an impact on decisions. Intelligent Resource Management (IRM) is based on events representing actual or possible changes of dangers as well as events

[1] The work presented here is focused on German fire protection processes. Within this domain the FWDV100 defines a cyclic process of command consisting of "observe/control", "plan (orient/decide)" and "command" phases. Therefore it is closely related to the presented OODA loop which is more familiar in other countries.

describing status changes of resources and steps within task execution. This builds up a demand for early and accurate event recognition.

### Event Processing

Event-orientation increases agility, reaction time and real-time capabilities of business applications (Bruns and Dunkel, 2010). These features are important for IT systems for dynamic and time-critical emergency operations. The Event-Driven Architecture (EDA) with Complex Event Processing (CEP) is a modern architectural pattern that can be combined with the Service-Oriented Architecture (SOA) architectural pattern. The resulting Event-Driven SOA architectural pattern (Taylor et al., 2009) follows the same idea as the cyclic process for emergency management. In detail three steps can be identified (the terminology of Event Processing Networks (EPN, see Etzion and Niblett, 2010) is used): (1) Sensors recognize real-world events, which are translated into event objects by event producer components. (2) Event processing components/agents apply several calculations (filter, transform, detect pattern) on top of the incoming raw event objects and produce new derived event objects. (3) Event objects are provided to event consumers. All components within the EPN (event producers, event processing agents and event consumers) are connected to each other by one or more event channels.

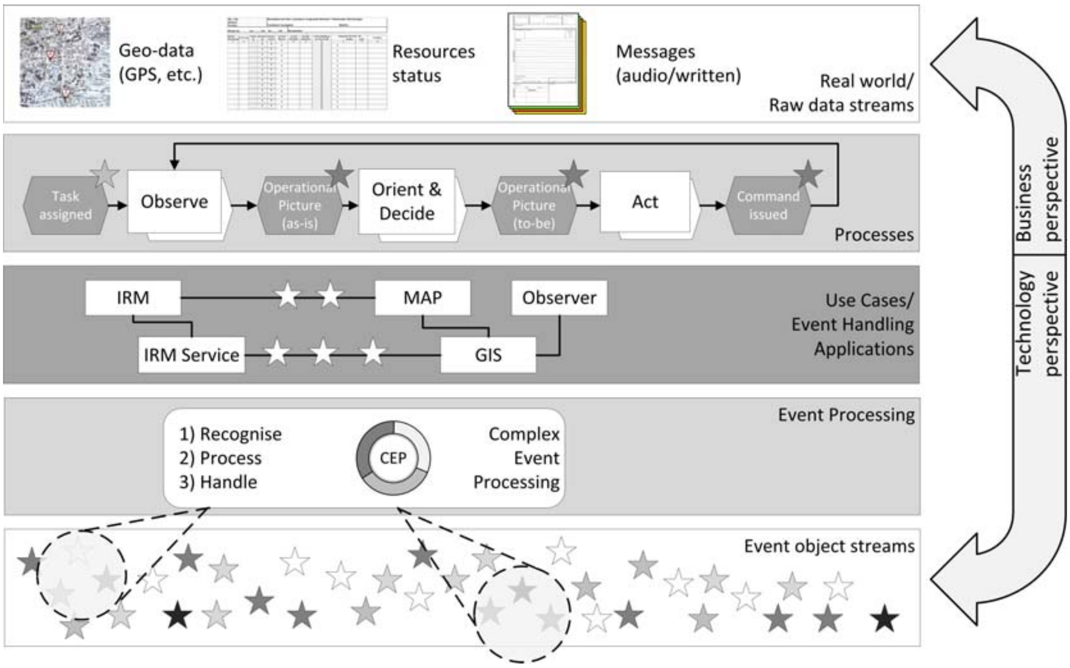


**Figure 3. Event Processing for Intelligent Resource Management (adopted from Bruns and Dunkel, 2010)**

The event-driven SOA based system which is referred to in this paper contains one central rule-based event processing agent (EPA), which calculates complex events as output. In order to facilitate the validation of the event definitions and, furthermore, perform event recognition, the approach for this EPA expresses the definitions in a knowledge representation language from the field of Artificial Intelligence. More precisely, a dialect of the Event Calculus was developed which is specifically tailored to the task of event recognition. The Event Calculus (EC, see Kowalski and Sergot, 1986) was introduced as a logic programming framework for representing and reasoning about events and their effects. The new EC dialect (Artikis, Sergot and Paliouras, 2010) extends the original EC by, among others, including built-in constructs, such as temporal interval manipulation predicates, facilitating the development of event definitions. In brief[2], it supports:

- Recognition of complex events. The full power of logic programming on which our EC dialect allows for the representation of complex temporal as well as atemporal constraints.
- Verification and traceability of results. This facility is due to the formal logic programming semantics of our EC dialect.

[2] Due to space limitations it is not possible to give here a detailed account of our EC dialect. Such an account, including an extensive library of event definitions for emergency rescue operations, is given by Artikis, Sergot and Paliouras, 2010.

- Compact representation of event definitions, facilitating code maintenance and end-user interaction.
- Efficient complex event detection even in the presence of thousands simple events.
- Automated construction and refinement of event definitions, in order to minimize the involvement of humans in the event recognition process and continuously improve recognition accuracy. The implementation of this feature is based on inductive logic programming techniques.

All other event processing agents in our system architecture calculate simple events, which serve as input for complex event processing (see Figure 3). The event channel is implemented as a Message-Oriented Middleware (MOM) with a publish-subscribe mechanism. GPS devices, audio and text communication recognizers as well as the Operation Control Center serve as the main event producers. Several applications use the event-driven backend sub-system as a live-data source. The IRM component integrates geo information (map view) and command structure information (IRM view). The Observer component shows an aggregated and synchronized real-time view of the event streams for three user groups with the common goal to increase their awareness: researchers (understanding dependencies between events), developers (understanding the behavior of the system) and stakeholders from the firefighting domain (understanding the situation within an operation).

## METHODOLOGY

As emphasized before, the foundation for achieving added value from event processing is the definition of an event model. Events have to be identified in real world processes which are (a) relevant for intelligent resource management and (b) recognizable by IT components. In case checking a specific event type for the latter criterion fails – technologically the recognition of this type of events is not possible – the corresponding event type is dismissed from the model. Alternatively, an IT system may provide the option to manually define and send an event based on observation results. For the identification and definition of event types we use a goal-oriented and scenario-based technique to enable open requirements analysis sessions. Focusing on goals instead of requirements allows a solution-independent discussion (cp. Pohl, 2007). In addition several definitions of the term "scenario" already imply their relevance to event identification. E. g., Sutcliffe (1998) interprets a scenario as "facts describing an existing system and its environment including the behavior of agents and sufficient context information to allow discovery and validation of system requirements". We build such scenarios by discussing real (training) operations with experienced fire officers. By incorporating real operations, an early focus on describing event objects instead of (real world) events is avoided. These scenarios are transformed to sequence diagrams (UML plus highlighting of event occurrences) in which domain experts (researchers) identify event types to be incorporated in the event model (cp. Sutcliffe, 2002, McGraw and Harbison, 1997, and Robertson and Robertson, 2006, concerning the analysis approach). (Friberg et al., 2010)

The event model is built by the description of event types. Each event type is defined by a set of header, payload and open content attributes. For event type elicitation a reduced list of parameters can be applied carrying all necessary information for later formalization (see Etzion and Niblett, 2010, for details):

- **Event Name**: Identifier for the event.
- **Event description**: Textual description of the event.
- **Event structure**: An event structure or definitions is provided by means of various operators, such the logical operators 'and' and 'or, and interval manipulation operators. Example event structure will be given shortly.
- **Duration characteristics:** Information whether the event is characterized as instantaneous or durative. For durative events indicators for starting and ending points are defined. In most cases this is implemented by "true" and "false" types for the concerned events.
- **Level of decisiveness:** For domain aspects the impact of events for decisions is important. Events might: (1) represent decisions, (2) be relevant for decisions, (3) be simple pieces of information or (4) be irrelevant for situational awareness and decision making. This distinction is useful at least in the evaluation of the Event Processing system when the impact of the system must be assessed.
- **Common reaction to event**: The common reaction to the event which might depend on user or role.
- **Reason for definition**: Reason for defining the event as a relevant event for the event set.
- **Example**: An example for the occurrence of the event.
- **Representation requirements**: Reference to use cases (and corresponding users) and user interface requirements (visualization and concerned functionality).

- **Recognition requirements**: Bearable delay between real occurrence and visualization within a GUI, an assessment of the impact of false negatives resp. positives for the operation and an assessment of the impact of false negatives resp. positives on the acceptance of the event driven system.
- **Recognition capability:** Decision whether the event driven system is capable of recognizing the event.
- **Relevant data sets**: Description of data sets are relevant to the definition/validation of the event.

As mentioned before the validation of the event definitions represents the major research challenge presented in this paper. Besides assessment of validity increased credibility and acceptance by users are the core goals of this research (Banks et al., 1996). Therefore a methodology is needed to assess the correlation between real world and event model (conceptual validation) with respect defined from the perspective of the use case. At first, data-based, quantitative measures have to be taken into account; secondly, experts and users have to be involved by means of qualitative measures (Sargent, 2005). For data-based methods, the essential boundary condition is set by available data in correspondence to event data sources specified within the event definition. For the requirement of complete coverage of the event types defined (cp. Rabinovich et al., 2010) it is obligatory that synchronized data is available whenever complex events are based on simple events relying on different types of data streams. Within the application domain availability of data represents a known problem; nevertheless focused validation is possible by combining complementary approaches (cp. Sargent, 2005, and references literature). Specific methods for validation in the area of event processing are hardly published. One example is the work of Rabinovich et al. which is compliant with the terminological backgrounds used in this work. At first, they divide validation into three phases: Static validation (at development-time, observing event types), dynamic validation (at run-time, observing event instances) and formal verification. These steps are embedded in an eight-step procedure where face validity is pre-condition for the mentioned ones. Based on these recommendations, our event model validation is conducted in four steps:

1) Collection of real data from 24 operations conducted during training courses of fire brigade officers, each of them lasting about 15 to 30 minutes including the most critical initialization phase of an operation. Thereby the requirement stated by Banks et al. (1996) is fulfilled: "[…] if one data set has been used to develop and calibrate the model, it is recommended that a separate data set be used as the final validation test." Besides event data streams, debriefings were observed to manually annotate the event sequences and solve inconsistencies within the data caused by the lack of observations (photos, tracks of vehicles, incident characteristics) or direct data recordings (communication channels).
2) Based on recordings, detailed scenarios (event sequences) were generated for each training exercise representing the results of event recognition assuming 100% accuracy. Simple events were implicitly included, and complex events were manually annotated. These streams were discussed with domain experts. For specific event types the approaches of Turing tests and sensitivity analysis were applied.
3) Specific resource management use cases for the static validation were defined for each complex event type. Static validation is performed as in Rabinovich et al. (2010).
4) The documentation of the recorded training exercises was transferred to simulation scenarios in order to prepare for dynamic validation. These scenarios contain event sequences and context restrictions to check the event model for identified issues and inconsistencies with respect to temporal dependencies.

Formal verification is out of scope for this part. It will be conducted within research on the computerized model.

## IDENTIFICATION AND DEFINITION OF EVENTS

In our case, events represent changes concerning the environment, the emergency or the operation itself (cp. Figure 2). For resource management, the most important aspect is the structure of operation which is managed throughout an operation. User-interaction with this structure and corresponding coordination activities represent events within the operation on different structural levels.

### Specification of event types

Concerning IRM, the "operation changed" event carries decisive information. It is an abstract event type subsuming all specialized events which represent a change to the emergency response activities. Both for in-situ assessment of the situation as well as for post-operational analysis these changes are important as they hint at decisions or actions taken during iterations of the cycle of command. Within this set of events a distinction can be made between structural, spatial and task-related changes. Figure 4 presents an example concerning the last category: The event "Unit got effective within the operation" (here it is interpreted in as a durative event) is based on a combination of "resource started fighting danger" preceding "danger decreased" correlated to the same danger occurrence within the emergency. An example of implementing this event type definition is to have a detection of fire within a thermal video image to recognize danger (i. e., fire) related events and to recognize task execution within a corresponding radio message by reporting fire fighters.[3]

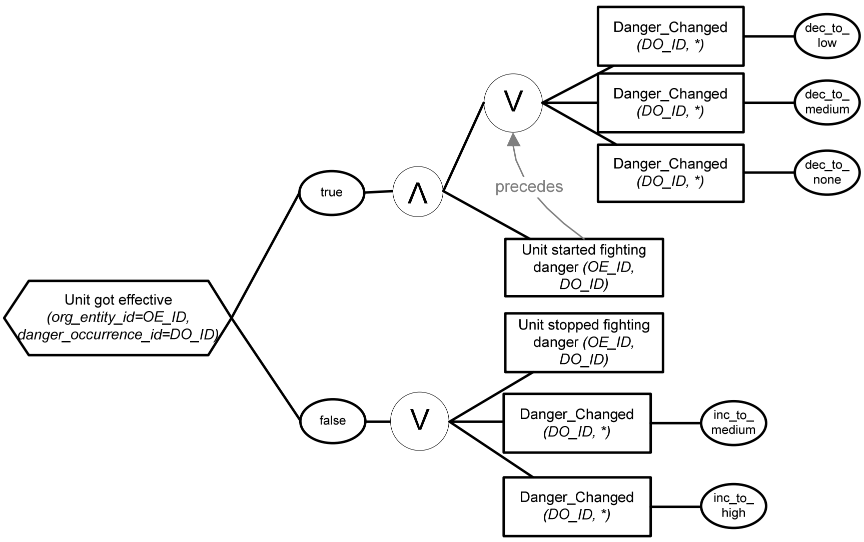


**Figure 4. Example of a complex event: "Unit got effective within the operation"**

Besides resource-driven decisions, fire officers have to find solutions for "critical situations". These represent situations in which a decision has to be taken; it becomes critical as more than one decision exists and the decision has to be taken within a certain interval (see Badke-Schaub; Buerschaper and Hofinger, 1999). A critical situation can be based on a "change of dangers" related to the operation. For command and control, an increasing danger as well as a new danger occurrence primarily represents something to react to. Furthermore, it might hint at mistakes done throughout emergency response and/or management. The impact of "changes within the environment" differs from simple context information to decisive events. Consider, for example, changes of weather conditions that may be meaningless for some tasks (e. g., normal temperature change), but meaningful for other tasks within the operation (e. g., rain or snow fall affecting rescue of people). Besides these generic classifications, some specific ones for resource management can be identified: "resource performed action" without affecting the operation directly (e. g., the acknowledgement of an alarm only gets important through the event "acknowledgement delayed"), "resource changed status" carrying important information for coordinating resources (e. g. resource reports to be available at its home base) and "information was transmitted" providing the opportunity to check events for consistency.

Based on the fact that events were identified within very different stages of scenarios, it is not possible to argue for a generic requirement for event recognition accuracy without reflecting it to every individual event type. The corresponding research goal is to identify deviations from such a generic requirement reaching for a realistic one. Therefore the impact of false positives (What happens if an event is recognized but did not occur in reality?) and false negatives (What if an event occurring in reality is not recognized?) is assessed during the requirements analysis phase. In addition a subjective assumption is given about the effect on acceptance of an event driven system in case of failure. As a result it can be stated that the high accuracy requirement is tested by this analysis without being able to state any clear number. Within the same research activities the question for a "bearable delay" between real occurrence of an event and its reporting to the highest command level is focused on. The result is that both for SE and for CE types the average means that reporting for the next decision or for the next status report is important. Status meetings that require preparation in the form of status reports are organized situation-dependent. This can mean the same kind of short-term requests for event data like in case of event data needed for the next decision Therefore the requirements for an event driven system should incorporate the need for information immediately for the next decision. Some deviations have to be mentioned as reporting with "approximately zero delay" is required: For SEs "Operation control center initiated a new operation" and "Vehicle availability changed" this is already the case within current processes, therefore no decrease of quality is acceptable; CEs "Danger related to operation increased" and "Demand for operation" imply high impact on the operation and therefore need immediate reporting.

[3] The combination of sensor types represents an exemplary demonstration case.

### Validation of the event specification

Validation, as well as verification, of an event model definition is needed before the transfer to a run-time model. Rabinovich et al. (2010) propose a methodology which is highly focused on verification. Nevertheless we adopt their basic methodology as it strongly emphasizes relationships between event types. Also for our validation these relationships are core research objects. The main difference is the focus on relationships between events on the one hand and event objects on the other hand.

Thus for static validation criteria are transferred to the event model to identify:

- Disconnected event types: Event types which are neither used as members of complex events nor carry relevant information for “event consumers”, i. e., for decision making processes.
- Event type consequences: Every path between an event type to consequent complex event types is identified to check the event model for consistency mainly caused by changes.
- Event type provenance: For each event type a path to an event producer has to be available.

For the purpose of validation all of these criteria are applied to each event type and its structure like the one presented in figure 4. Therefore a hierarchy of event types is used: Level 0 events are directly based on data sources (simple events), level 1 events are characterized by level 0 events as members, level 2 events include level 1 events as members and so on. Validation starts with level 0 heading forward to higher levels; for event type provenance backwards. Each issue raised by one of the test procedures is discussed with end users to either dismiss the concerned event type from the model or refine the event model. Validation steps have to be repeated for the concerned event types. In the specific event model which is researched on cycles do not have to be addressed as higher level events (e. g., level 4) may not be input for lower level (e. g., level 3) events.

The event model definition already incorporates temporal restrictions. E. g., to recognize the event “Unit got effective” it is important that at first this unit started fighting a danger and the danger decreased (e. g., a fire was extinguished). For validation of temporal restrictions – both the existence as well as the dimension – dynamic aspects are important. Rabinovich et al. (2010) propose event instance forward and backward traces as well as application coverage by scenario execution.

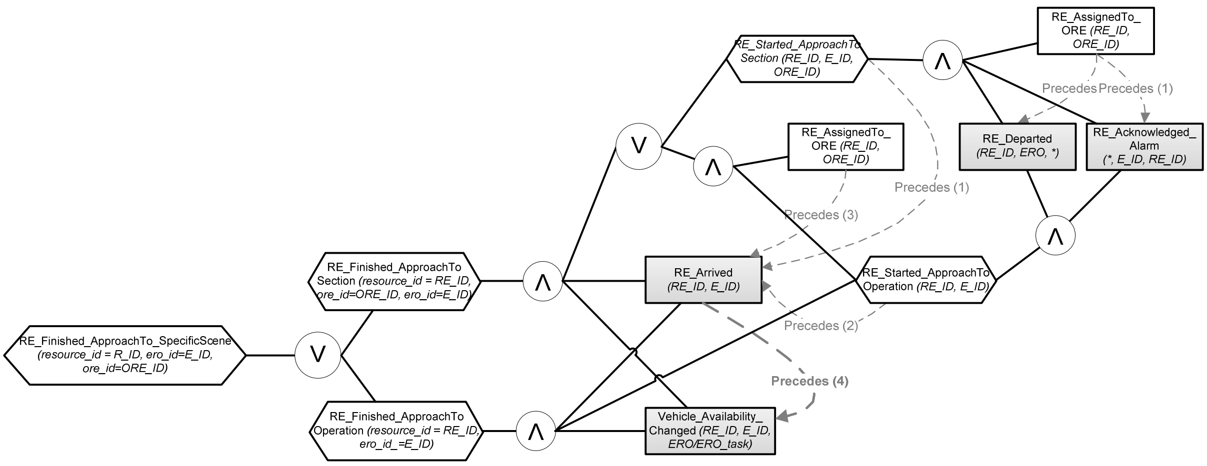


**Figure 5. Extract from a scenario: Overlay of paths for the event “Resource finished approach to specific scene”**

For validation purposes these procedures are adopted by applying the collected data of training exercises:

- Backward trace: Instead of simulating event instances, event sequences, i. e., training exercise logs, are analyzed per temporal restriction. For the event type example given in figure 4 this means that for every real event “resource got effective” the event sequence before this specific event occurred is checked for an occurrence of “Unit started fighting danger” and “Danger decreased”. Having identified these events the temporal relation between them is compared to the definition within the event model.
- Forward trace: For each event the possible forward paths to higher levels are identified. Based on that the log is checked for event occurrences for which the first event is a member. This is done step by step reaching for higher levels. Thereby it is tested whether the expected effects of events are proven.

- Event model coverage by scenario execution: An overlay of forward and backward traces highlights all event types which are relevant for a specific scenario. Based on that the relevance of event types can be assessed. An extract from an example is provided in figure 5: The finalization of the approach of vehicles to the site of an incident can be modeled in different ways. This event is represented on the left side; OR and AND operators are used to visualize different ways of building this complex events; “precedes” restrictions mark temporal relationships between events. Visualizing all paths used in one scenario in one graph the relevance of the departure, the acknowledgement of the alarm, the arrival and the confirmed availability change as event types which are always applicable (marked in yellow).

## CONCLUSION

Event Processing for crisis management and emergency response offers high potential for information systems. For this field Event Processing technology – while growing in terms of significance for many business areas – has to be adopted starting with real events: An incident, a crisis or a disaster actually is an “event” happening in an environment, i. e., a specific “context”. Emergency response efforts add events to this “context” building up situations. Decision makers have to react to “critical situations” – therefore the situational awareness they need is built up both from danger events as wells as emergency response events. Following this argumentation observable real world objects and activities have to be starting point for event model definitions. The strongest requirements for definition of event types and their relationships is their accuracy concerning decisive situations. Thus, significant efforts have to be spent for validation to prepare for building computerized models and corresponding verification activities. This paper proposes to extend the approach of Rabinovich et al. (2010) by validating activities: Based on both real world data as well as domain expert knowledge, static and dynamic validation should be performed using methods which are later applied for verification aspects of the computerized model. Cross-links can be expected as useful side-effects.

These results will be utilized by transferring the event model to an event object model resp. a computerized version. This will be on the one hand verified to provide new knowledge about the feasibility of event recognition within this field, and on the other hand event processing results will be fed to an IT system visualizing the operational picture (for details see http://www.ict-pronto.org).

## ACKNOWLEDGMENTS

This work has been partially funded by the EC, in the context of the PRONTO Project (FP7-ICT 231738). We would like to thank the State fire service institute North Rhine Westfalia (Germany) and the participants from different fire brigades for supporting this study by providing the opportunity to observe training exercises. The authors themselves, however, are solely responsible for any misunderstanding about the utilization of that data.

## REFERENCES

1. Allen, James F.; Hayes, Patrick J. (1985): A common-sense theory of time. In: Proceedings of the 9th international joint conference on Artificial intelligence - Volume 1. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., p. 528-531.
2. Artikis A., Sergot M., and Paliouras G. (2010) A logic programming approach to activity recognition. In Proceedings of ACM Workshop on Events in Multimedia.
3. Badke-Schaub, Petra; Buerschaper, Cornelius; Hofinger, Gesine (14.-15.10.1999): Merkmale kritischer Situationen: Paper für den Workshop "Kritische Situationen" der Plattform "Menschen in komplexen Arbeitsumwelten e.V.". Radebeul.
4. Banks, Jerry; Carson, John p.; Nelson, Barry L. (1996): Discrete event system simulation. 2. ed. Upper Saddle River, NJ: Prentice-Hall (Prentice-Hall international series in industrial and systems engineering).
5. Birkhäuser, Benedikt; Pottebaum, Jens; Koch, Rainer (2009): Unterstützung von Einsatzentscheidungen der Feuerwehr auf Basis IT-unterstützter Kräftekoordination. In: Fischer, Stefan (Eds.): Informatik 2009. Bonn: Ges. für Informatik (GI-EditionProceedings), Bd. 154.

6. Birkhäuser, Benedikt; Schwenniger, Jochen; Loke, Ben; Niittylä, Laura; Karampiperis, Pythagoras; Marterer, Robin (2010): Entwicklung eines ereignisgetriebenen Systems für intelligentes Ressourcenmanagement - 'Work-in-Progress'-Report zum Stand des ersten Demonstrators. In: Klaus-Peter Fähnrich; Bogdan Franczyk (Eds.): Informatik 2010: Service Science - Neue Perspektiven für die Informatik, Beiträge der 40. Jahrestagung der Gesellschaft für Informatik e.V. (GI), Band 2, 27.09. - 1.10.2010, Leipzig: GI (LNI), Bd. 176, p. 143–149.
7. Boyd, John R. (1996): The Essence of Winning and Loosing. URL http://www.projectwhitehorse.com/pdfs/boyd/The%20Essence%20of%20Winning%20and%20Losing.pdf.
8. Edmonds, John; Raschid, Louiqa; Sayyadi, Hassan; Wu, Shanchan (2010): Exploiting Social Media to Provide Humanitarian Users with Event Search and Recommendations. In: French, Simon; Tomaszewski, B.; Zobel C. (Eds.): Proceedings of the 7th International Conference on Information Systems for Crisis Response and Management ISCRAM2010. Seattle, USA.
9. Etzion, Opher; Niblett, Peter (2011): Event processing in action. Stamford, Conn.: Manning.
10. Friberg, Therese; Birkhäuser, Benedikt; Pottebaum, Jens; Koch, Rainer (2010): Using Scenarios for the Identification of Real-World Events in an Event-Based System. In: French, Simon; Tomaszewski, B.; Zobel C. (Eds.): Proceedings of the 7th International Conference on Information Systems for Crisis Response and Management ISCRAM2010. Seattle, USA.
11. Gonzalez, Rafael A. (2009): Crisis response simulation combining discrete-event and agent-based modeling. In: Proceedings of the 6th International ISCRAM Conference. Gothenburg .
12. IEEE Std 1012-2004 IEEE standard for software verification and validation (revision of IEEE Std 1012-1998) (2005). New York, NY: IEEE (IEEE standards).
13. Kowalski R.; Sergot M. (1986). A logic-based calculus of events. New Generation Computing 4(1):67–96.
14. Luckham, David; Schulte, Roy (Eds.) (2008): Event Processing Glossary: Version 1.1. URL http://www.ep-ts.com/component/option,com_docman/task,doc_download/gid,66/Itemid,84/, last visit 24.06.2010.
15. McGraw, Karen L.; Harbison, Karan (1997): User-centered requirements: The scenario-based engineering process. Mahwah, NJ: Lawrence Erlbaum Associates.
16. Pohl, Klaus (2007): Requirements engineering: Grundlagen, Prinzipien, Techniken. 1. Aufl. Heidelberg: dpunkt Verlag.
17. Pottebaum, Jens; Japs, Anna Maria; Prödel, Stephan; Koch, Rainer (2010): Design and modelling of a domain ontology for fire protection. In: French, Simon; Tomaszewski, B.; Zobel C. (Eds.): Proceedings of the 7th International Conference on Information Systems for Crisis Response and Management ISCRAM2010. Seattle, USA.
18. Rabinovich, Ella; Opher Etzion; Sitvanit Ruah; Sarit Archushin (2010): Analyzing the behavior of event processing applications. In: Jean Bacon; Peter R. Pietzuch; Joe Sventek; Ugur Çetintemel (Eds.): Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems, DEBS 2010, Cambridge, United Kingdom, July 12-15, 2010: ACM, p. 223–234.
19. Ralf Bruns; Jürgen Dunkel (2010): Event-Driven Architecture: Softwarearchitektur für ereignisgesteuerte Geschäftsprozesse. Berlin: Springer.
20. Robertson, Suzanne; Robertson, James (2006): Mastering the requirements process. 2nd ed. Upper Saddle River, NJ: Addison-Wesley.
21. Sargent, Robert G. (2005): Verification and validation of simulation models. In: Proceedings of the 37th conference on Winter simulation: Winter Simulation Conference (WSC '05), p. 130-143.
22. Sutcliffe, Alistair (1998): Scenario-based requirements analysis. In: Requirements Engineering, Jg. 3, H. 1, p. 48–65.
23. Sutcliffe, Alistair (2002): User-centred requirements engineering. London: Springer.
24. Taylor, Hugh; Yochem, Angela; Phillips, Les; Martinez, Frank (2009): Event-Driven Architecture: How SOA Enables the Real-Time Enterprise. Boston: Addison-Wesley Professional.
25. Taylor, Hugh; Yochem, Angela; Phillips, Les; Martinez, Frank (2009): Event-Driven Architecture: How SOA Enables the Real-Time Enterprise. Boston: Addison-Wesley Professional.

26. ter Mors; Valk; Witteveen (2005): An Event-Based Task Framework for Disaster Planning and Decision Support. In: Proceedings of the 3rd International ISCRAM conference. Brussels .
27. U.S. Department of Defense (2007): DoD Modeling and Simulation (M&S) Management: Directive. (5000.59). URL http://www.dtic.mil/whs/directives/corres/pdf/500059p.pdf, last visit 13.11.2010.
28. Wickler, Gerhard; Tate, Austin; Potter, Stephen (2007): Integrating Discrete Event and Process-Level Simulation for Flexible Training in the I-X Framework. In: Proceedings of the 5th International ISCRAM conference. Delft.