Event Modelling and Reasoning with Uncertain Information for Distributed Sensor Networks

Jianbing Ma, Weiru Liu, and Paul Miller

School of Electronics, Electrical Engineering and Computer Science,
Queen's University Belfast, Belfast BT7 1NN, UK
{jma03,w.liu}@qub.ac.uk, p.miller@ecit.qub.ac.uk

Abstract. CCTV and sensor based surveillance systems are part of our daily lives now in this modern society due to the advances in telecommunications technology and the demand for better security. The analysis of sensor data produces semantic rich events describing activities and behaviours of objects being monitored. Three issues usually are associated with events descriptions. First, data could be collected from multiple sources (e.g., sensors, CCTVs, speedometers, etc). Second, descriptions about these data can be poor, inaccurate or uncertain when they are gathered from unreliable sensors or generated by analysis non-perfect algorithms. Third, in such systems, there is a need to incorporate domain specific knowledge, e.g., criminal statistics about certain areas or patterns, when making inferences. However, in the literature, these three phenomena are seldom considered in CCTV-based event composition models. To overcome these weaknesses, in this paper, we propose a general event modelling and reasoning model which can represent and reason with events from multiple sources including domain knowledge, integrating the Dempster-Shafer theory for dealing with uncertainty and incompleteness. We introduce a notion called event cluster to represent uncertain and incomplete events induced from an observation. Event clusters are then used in the merging and inference process. Furthermore, we provide a method to calculate the mass values of events which use evidential mapping techniques.

Keywords: Bus Surveillance; Active System; Event Composition; Event Reasoning; Inference.

1 Introduction

CCTV-based[1] surveillance is an inseparable part of our society now – everywhere we go we see CCTV cameras (e.g. [2, 11, 5, 13], etc). The role of such systems has shifted from purely passively recording information for forensics to proactively providing analytical information about potential threats/dangers in real-time fashion. This shift poses

[1] This paper is an extended version of [9] in which we have included a set of running examples, a method (summarized by an algorithm) to calculating mass values of events which uses evidential mapping techniques, and the newly introduced notion *rule clusters*. Furthermore, we also demonstrate in this paper how to interpret and use the background knowledge, or domain knowledge, which was only preliminarily introduced in [9].

some dramatic challenges on how information collected in such a network shall be exchanged, correlated, reasoned with and ultimately be used to provide significantly valuable predictions for threats or actions that may lead to devastating consequences.

Central to this is the ability to deal with a large collection of meaningful events derived from sensor/camera data analysis algorithms. An event can be understood as something that happened somewhere at a certain time (or time interval). Typically, a life cycle of event includes detection, storage, reasoning, mining, exploration and actions.

In this paper, we focus on a real-time event modelling and reasoning framework for supporting the instant recognition of emergent events based on uncertain or imperfect information from multiple sources. This framework has many potential uses in various applications, e.g., active databases, smart home projects, bus/airport surveillance, and stock trading systems, etc.

Various event reasoning systems have been proposed in the literature, e.g., an event language based on *event expressions* for active database systems in [6], the Semantic Web Rule Language (SWRL) for semantic web applications and the situation manager rule language [1] for general purposes, etc. These systems provide both event representation and deterministic event inference in the form of rules. However, these systems do not take into account uncertainties which are usually associated with real-world events. To remedy this weakness, in [15–17], an event composition model was proposed with uncertainties represented by probability measures.

However, this model cannot deal with the problem of incomplete information in event reasoning. For example, in the case of monitoring a person entering an building, the person may be classified as male with a certainty of 85% by an event detection algorithm (an event here is to identify a person's gender). However, the remainder does not imply that the person is female with a 15% certainty, rather, it is unknown. That is, we do not know how the remaining 15% shall be distributed on alternatives $\{\text{male}\}$ or $\{\text{female}\}$. Hence with probability theory, this information can only be represented as $p(male) \geq 0.85$ and $p(female) \leq 0.15$ which is difficult for subsequent reasoning (e.g., a Bayesian network).

In distributed sensor networks, events are more often gathered from multiple heterogeneous sources, e.g., the same event can be obtained from video or audio data analysis, or from speedometers. We assume that each source channels its information via event descriptions, hence a practical event model should consider combining information about the same event from multiple sources. As different sources may provide possibly conflicting descriptions on the same event, the event composition model should also be able to deal with such conflict between multiple sources. Unfortunately, to the best of our knowledge, this issue is hardly mentioned in the literature on event composition models. In [1], although events can come from multiple sources, a particular event can only be from one source, so this model cannot deal with multiple events from different sources relating to the same situation (scenario).

Furthermore, when an event reasoning system receives events descriptions from multiple sources, it also needs to consider the reliabilities of these information sources. For instance, in surveillance applications, sensors/cameras, etc, are frequently used. However, since sensors/cameras can be malfunctioning such as a camera may have been tampered with, illumination could be poor, or the battery is low, etc, they may give

imprecise information which cannot be simply represented by probability measures, either.

Dempster-Shafer (DS) Theory [4, 12] is a popular framework to deal with uncertain or incomplete information from multiple sources. This theory is capable of modelling incomplete information through ignorance as well as considering the reliabilities of sources by using the discounting function. In this paper, we propose an event model integrating DS theory that can represent and reason with possibly conflicting information (recorded as events) from multiple sources which may be uncertain or incomplete. We also deploy the *discounting* function [8] to resolve imprecise information due to unreliable sources.

Furthermore, it is also a key requirement for an event model to have the ability to represent and manage domain knowledge [14]. Because domain knowledge does not fit into the usual definitions of events in the literature, it is not surprising that it is generally ignored by the existing event models, e.g., [1, 15–17], etc. In our event model, however, domain knowledge is treated as a special kind of event and is managed the same way as other types of events.

To summarize, the main contributions of our event composition model are

1. a general model for representing uncertain and incomplete information (events),
2. a combination framework for dealing with events from multiple sources,
3. utilization of domain knowledge for assisting inferences,
4. using evidential mapping technique to calculate the event mass.

The framework has been implemented and tested with a set of events acquired from the Intelligent Sensor Information System (ISIS) project which aims at developing a state-of-the-art surveillance sensor network concept demonstrator for public transport. A set of domain specific rules are constructed with the help of criminologist working on the project.

The rest of the paper is organized as follows. In Section 2, we provide the preliminaries on Dempster-Shafer theory. In Section 3, formal definitions of event model are given including the definitions of events, multi-source events combination, event flow and event inference. We then provide an algorithm for calculating mass values of events in Section 4. Finally, we discuss related work and conclude the paper in Section 5 and Section 6 respectively.

2 Dempster-Shafer Theory

For convenience, we recall some basic concepts of Dempster-Shafer's theory of evidence (DS theory). Let Ω be a finite, non-empty set called the frame of discernment, denoted as, $\Omega = \{w_1, \cdots, w_n\}$.

Definition 1 *A mass function is a mapping* $m : 2^{\Omega} \rightarrow [0, 1]$ *such that* $m(\emptyset) = 0$ *and* $\sum_{A \subseteq \Omega} m(A) = 1$.

If $m(A) > 0$, then A is called a focal element of m. Let $\mathscr{F}_m$ denote the set of focal elements of m. From a mass function, m, *belief* function (Bel) and *plausibility* function

(Pl) can be defined to represent the lower and upper bounds of the beliefs implied by m as follows.

$$Bel(A) \;=\; \sum_{B \subseteq A} m(B) \;\text{ and }\; Pl(A) \;=\; \sum_{C \cap A \neq \emptyset} m(C). \tag{1}$$

One advantage of DS theory is that its has the ability to accumulate and combine evidence from multiple sources by using *Dempster's rule of combination*. Let m_1 and m_2 be two mass functions from two distinct sources over Ω. Combining m_1 and m_2 gives a new mass function m as follows:

$$m(C) \;=\; (m_1 \oplus m_2)(C) \;=\; \frac{\sum_{A \cap B = C} m_1(A) m_2(B)}{1 - \sum_{A \cap B = \emptyset} m_1(A) m_2(B)} \tag{2}$$

In practice, sources may not be completely reliable, to reflect this, in [12], a *discount rate* was introduced by which the mass function may be discounted in order to reflect the reliability of a source. Let r $(0 \leq r \leq 1)$ be a discount rate, a discounted mass function using r is represented as:

$$m^r(A) = \begin{cases} (1-r)m(A) & A \subset \Omega \\ r + (1-r)m(\Omega) & A = \Omega \end{cases} \tag{3}$$

When $r = 0$ the source is absolutely reliable and when $r = 1$ the source is completely unreliable. After discounting, the source is treated as totally reliable.

In our event composition and inference model, we use a set of rules (with degrees of certainty) to describe which collection of events could imply what other events to a particular degree. A simplified form of a rule of this kind[2] is as *if E then H_1 with degree of belief f_1, ..., H_n with degree of belief f_n*. These rules are called heuristic in [7], in which a modelling and propagation approach was proposed to represent a set of heuristic rules and to propagate degrees of beliefs along these rules, through the notion *evidential mapping Γ^**.

An evidential mapping is to establish relationships between two frames of discernment Ω_E, Ω_H such that $\Gamma^* : 2^{\Omega_E} \rightarrow 2^{2^{\Omega_H \times [0,1]}}$ assigning a subset $E_i \subseteq \Omega_E$ to a set of subset-mass pairs in the following way:

$$\Gamma^*(E_i) = \big((H_{ij},\; f(E_i \rightarrow H_{ij})),\; ...,\; (H_{it},\; f(E_i \rightarrow H_{it}))\big) \tag{4}$$

where $H_{ij} \subseteq \Omega_H$, $i = 1,\; ...,\; n$, $j = 1,\; ...,\; t$, and $f : 2^{\Omega_E} \times 2^{2^{\Omega_H}} \rightarrow [0,1]$ satisfying[3]

(a) $H_{ij} \neq \emptyset,\; j = 1,\; ...,\; t$;
(b) $f(e_i \rightarrow H_{ij}) \geq 0,\; j = 1,\; ...,\; t$;
(c) $\sum_{j=1}^{n} f(e_i \rightarrow H_{ij}) = 1$;
(d) $\Gamma^*(\Omega_E) = \big((\Omega_H, 1)\big)$;

A piece of evidence on Ω_E can then be propagated to Ω_H through evidential mapping Γ^* as follows:

$$m_{\Omega_H}(H_j) = \sum_i m_{\Omega_E}(E_i) f(E_i \rightarrow H_{ij}). \tag{5}$$

[2] The definition of rules is given in Section 3.

[3] For the sake of clear illustration, instead of writing $f(E_i, H_{ij})$, we write $f(E_i \rightarrow H_{ij})$.

To calculate the mass values of inferred events based on the premise events of inference rules, we integrate *evidential mapping* Γ^* technique [7] into our event composition and reasoning model, which is detailed in Section 4.

3 A general Framework for Event Modelling

3.1 Event Definition

For an event model, the first issue we should address is the definition of events. The definition of an event should be expressive enough to deliver all the information of interest for an application and also be as simple and clear as possible.

Definitions of an event from different research fields are very diverse and tend to reflect the content of the designated application. For instance, in text topic detection and track, an event is something that happened somewhere at a certain time; in pattern recognition, an event is defined as a pattern that can be matched with a certain class of pattern types, and in signal processing, an event is triggered by a status change in the signal, etc.

In this paper, to make our framework more general, we define the events as follows: an event is *an occurrence that is instantaneous (event duration is* 0*, i.e., takes place at a specific point of time)*[4] *and atomic (it happens or not)*. The atomic requirement of an event does not exclude uncertainty. For instance, when there is a person boarding a bus and this person can be a male or a female (suppose we only focus on the gender), then whether it is a male/female that boards the bus is an example of uncertainty. But *a male (resp. a female) is boarding the bus* is an atomic event which either occurs completely or does not occur at all. To represent uncertainty encountered during event detection, in the following, we distinguish an *observation* (with uncertainty) from *possible events associated with the observation* (because of the uncertainty). This can be illustrated by the above example: an observation is that *a person is boarding the bus* and the possible events are *a male is boarding the bus* and *a female is boarding the bus*. An observation says that something happened, but the *entity* being observed is not completely certain yet, so we have multiple events listing what that *entity* might be.

This definition of events is particularly suitable for surveillance problems, where the objects being monitored are not complete clear to the observer.

In the literature, there are two types of events, one type contains *external events* [1] or *explicit events* [15, 16] and the other consists of *inferred events*. External events are events directly gathered from external sources (within the application) while inferred events are the results of the inference rules of an event model. In addition, to make use of domain knowledge, we introduce the third type of events, *domain events*, which are usually extracted from experts's opinions or background knowledge about this application the domain. Intuitively, domain knowledge is not from observed facts while external events are. Examples of these events can be seen in the next subsection.

[4] Domain events introduced later in this subsection may have a nonzero duration. A domain event can be seen as a series of instantaneous events.

3.2 Event Representation

Intuitively, a concrete event definition is determined by the application domain which contains all the information of interest for the application (including data relevant to the application and some auxiliary data). But there are some common attributes that every event shall possess, such as

1. $EType$: describing the type of an event, such as, *Person Boarding Vehicle* abbreviated as PBV.
2. $occT$: the point in time that an event occurred.
3. ID: the ID of a source from which an event is detected.
4. rb: the degree of reliability of a source.
5. sig: the degree of significance of an event.

Formally, we define an event e as follows.

$$e = (EType, occT, ID, rb, sig, v_1, \cdots, v_n)$$

where v_is are any additional attributes required to define event e based on the application. Attribute v_i can either have a single or a set of elements as its value, e.g., for attribute *gender*, its value can be *male*, or *female*, or $\{$*male, female*$\}$ (however, it is not possible to tell the gender of a person when their face is obscured, so we introduce a value *obscured* as an unknown[5] value for gender). Any two events with the same event type, source ID and time of occurrence (Typically the occurrence time is like $21:05:31pm12/2/09$, for simplicity we only use the hours) are from the set of possible events related to a single observation. For example, $\mathsf{e}_1 = (\mathsf{PBV}, \mathsf{20pm}, 1, 0.8, 0.7, \mathsf{male}, \cdots)$ and $\mathsf{e}_2 = (\mathsf{PBV}, \mathsf{20pm}, 1, 0.8, 0.7, \{\mathsf{male}, \mathsf{female}\}, \cdots)$ are two events with v_1 for *gender* (we have omitted other attributes for simplicity).

Events of the same type have the same set of attributes.

Example 1 *Suppose we are monitoring passengers boarding a bus through the front door. Then we have an event type* PBV *(Person Boarding Vehicle) which may include related attributes such as source ID, occurrence time, reliability, significance, person gender, person ID, person age, front/back door, vehicle type, vehicle ID, bus route (we omit the bus position for simplicity). An instance of* PBV *is* (PBV, 21pm, 1, 0.9, 0.7, male, 3283, young, fDoor, double decker bus, Bus1248, 45).

An event is always attached with a mass value. Semantically, for a particular event type with each of its event represented as $(EType, occT, ID, rb, sig, v_1, \cdots, v_n)$, we use Dom_i to denote the domain of v_i, and $\mathscr{Y} = \prod_{i=1}^{n} Dom_i$ to denote the frame of discernment (domain of tuple $(v_1, \cdots, v_n)$), and m to denote a mass function over $2^{\mathscr{Y}}$.

To represent an observed fact with uncertainty, we introduce concept *event cluster*. An *event cluster* EC is a set of events which have the same event type ($EType$), occurrence time ($occT$) and source ID (ID) , but with different $v_1, \cdots, v_n$ values. Events e_1

[5] Note that *obscured* is not the same as $\{$male, female$\}$ since it may indicate some malicious event. In fact, here the attribute *gender* can be seen as an output from a face recognition program in which *obscured* means that information about face recognition is not available. In this sense, *gender* is an abbreviation of *gender recognition result*.

and e_2 above form an event cluster for the observed fact *someone is boarding the bus*. Note that as the reliability is based on the source, events in a specified event cluster EC will have the same reliability.

For an event e in event cluster EC, we use $e.EType$ (resp. $e.occT$, etc) to denote the event type (resp. time of occurrence, etc) of e, $e.v$ to denote $(v_1, \cdots, v_n)$, and $e.m$ to denote the value $m(e.v)$. By abuse of notations, we also write $EC.EType$ (resp. $EC.ID$, $EC.occT$, $EC.rb$) to denote the event type (resp. source ID, time of occurrence, reliability) of any event in EC since all the events in EC have the same values for these attributes.

It should be noted that within a particular application, the degree of significance of an event is self-evident (i.e., a function over $e.v$). For example, in bus surveillance, the event *a young man boards a bus around 10pm in an area with high crime statistics* is more significant than the event *a middle-aged woman boards a bus around 6pm in an area of low-crime*. However, due to space limitation, we will not discuss it further.

A mass function m over $\mathscr{V}$ for event cluster EC should satisfy the normalization condition: $\sum_{e \in EC} e.m = 1$. That is, EC does contain an event that really occurred. For example, for the two events, e_1 and e_2, introduced above, a mass function m can be defined as $m(\mathsf{male}, \cdots) = 0.85$ and $m(\{\mathsf{male}, \mathsf{female}\}, \cdots) = 0.15$.

An event cluster hence gives a full description of an observed fact with uncertainty from the perspective of one source.

Example 2 *(Example 1 continued) Consider a camera overlooking the front entrance to a bus. There are several possible events relating to a camera recording: a male boards the bus, a female boards the bus, a person that we cannot distinguish it is a man or woman boards the bus, a person boards the bus but its face is obscured. Among these the last is the most significant as someone who boards the bus with its face obscured is likely to be up to no good. Most vandals and criminals will take steps to ensure their faces are not caught by the cameras. Therefore, we have the following event cluster with a set of events as (we omitted other details for simplicity)*
$\{(\mathsf{PBV}, 21:05:31, 1, 0.9, 0.7, \mathsf{male}, 3283)$, $(\mathsf{PBV}, 21:05:31, 1, 0.9, 0.4, \mathsf{female}, 3283)$,
$(\mathsf{PBV}, 21:05:31, 1, 0.9, 0.7, \{\mathsf{male}, \mathsf{female}\}, 3283)$,
$(\mathsf{PBV}, 21:05:31, 1, 0.9, 1, \mathsf{obscured}, 3283)\}$.

A sample mass function assigning mass values to focal elements from frame $\Omega = \{male, female, obscurred\}$ *can be* $m(\mathsf{male}, 3283) = 0.4$ $m(\mathsf{female}, 3283) = 0.3$, $m(\{\mathsf{male}, \mathsf{female}\}, 3283) = 0.2$, *and* $m(\mathsf{obscured}, 3283) = 0.1$.

Observe that if some E^* in ET s.t., $E^*.m = 1$, then the event cluster ET simply reduces to a single event, i.e., E^* (other events with mass values 0 are ignored).

Domain knowledge[6] can be represented as a special event cluster in which an event (called a *domain event*) is in the same form of the external/inferred events except that the time of occurrence can be an interval.

Example 3 *(Example 2 con't) After a survey, we obtained a distribution (with reliability* 0.8*) on person boarding bus 1248 at route 45 between* $20:00$ *and* $22:00$ *as* $\mathsf{male} : \mathsf{female} : \mathsf{obscured} = 5 : 4 : 1$. *Then this piece of domain knowledge produces*

[6] In our event model, we reserve source 0 for domain knowledge.

the following event cluster with events
(PBV, [20 : 00 : 00, 22 : 00 : 00], 0, 0.8, 0.7, male,double decker bus, Bus 1248, 45),
(PBV, [20 : 00 : 00, 22 : 00 : 00], 0, 0.8, 0.4, female, double decker bus, Bus 1248, 45), *and*
(PBV, [20 : 00 : 00, 22 : 00 : 00], 0, 0.8, 1, obscured,double decker bus, Bus 1248, 45).

The mass values given by the domain knowledge are (we omitted other details here) $m(\mathsf{male}, \cdots) = 0.5$, $m(\mathsf{female}, \cdots) = 0.4$, *and* $m(\mathsf{obscured}, \cdots) = 0.1$.

A domain event has a *series interpretation* such that it contains a series of external events, the occurrence time of each such external event is one time point in the time interval of the domain event while other attributes are unchanged. For instance, event (bus details omitted) $(\mathsf{PBV}, [20:00:00, 22:00:00], 0, 0.8, 0.7, \mathsf{male}, \cdots)$ can be seen as a series of events
$(\mathsf{PBV}, 20:00:00, 0, 0.8, 0.7, \mathsf{male}, \cdots)$, $(\mathsf{PBV}, 20:00:01, 0, 0.8, 0.7, \mathsf{male}, \cdots)$, $\cdots$,
$(\mathsf{PBV}, 22:00:00, 0, 0.8, 0.7, \mathsf{male}, \cdots)$.

With the series interpretation of domain events, intuitively we do not allow two domain events having the same event attribute values except that their time intervals are overlapped. To illustrate, if one domain event gives $e_3 = (\mathsf{PBV}, [20:00:00, 22:00:00], 0, 0.8, 0.7, \mathsf{male}, \cdots)$ with $e_3.m = 0.5$ (i.e., $m(\mathsf{male}, \cdots) = 0.5$) and another domain event provides $e_4 = (\mathsf{PBV}, [21:00:00, 23:00:00], 0, 0.8, 0.7, \mathsf{male}, \cdots)$ with $e_4.m' = 0.9$ (i.e., $m(\mathsf{male}, \cdots) = 0.9$), then they contradict each other during the time interval $[21:00:00, 22:00:00]$. If the second domain event gives the same mass value (i.e., $m'(\mathsf{male}, \cdots) = 0.5$), then in fact these two event clusters can be merged with a time interval $[20:00:00, 23:00:00]$. Similarly, two domain event clusters with their events pairwise having the same attributes values except overlapping time intervals either contradict each other or can be merged into one domain event cluster. Hence hereafter we assume that there does not exist two domain event clusters having the same event type and with overlapping time intervals.

3.3 Event Combination

When a set of event clusters have the same event type and time of occurrence but different source IDs, we call them *concurrent* event clusters[7]. This means that multi-model sensors may have been used to monitor the situation. Therefore, we need to combine these event clusters since they refer to the same observed fact from different perspectives. The combined result is a new event cluster with the same event type and time of occurrence, but the source ID of the combined event will be the union of the original sources. The combination of event clusters is realized by applying Dempster's combination rule on discounted mass functions. That is, the mass function of an event cluster is discounted with the discount rate defined as the reliability of a source.

Definition 2 *Let* $EC_1, \cdots, EC_k$ *be a set of concurrent event clusters, and* $m_1^r, \cdots, m_k^r$ *be the corresponding discounted mass functions over* $2^{\mathscr{V}}$, m *be the mass function obtained by combining* $m_1^r, \cdots, m_k^r$ *using the Dempster's combination rule, then we get*

[7] Due to the series interpretation of domain event clusters, a domain event cluster and an external event cluster are called concurrent iff the time of occurrence of the latter is within the time interval of the former.

the combined event cluster $EC = \oplus_{j=1}^{k} EC_j$ *such that* $\forall e \in EC$*, we have* $e.v \in \mathscr{F}_m$*,* $e.EType = EC_1.EType$*,* $e.occT = EC_1.occT$*,* $e.ID = \{EC_1.ID, \cdots, EC_k.ID\}$*,* $e.rb = 1$*, and* $e.m = m(e.v)$*. Conversely, for each focal element* A *in* $\mathscr{F}_m$*, there exists a unique* $e \in EC$*, s.t.,* $e.v = A$*.*

As stated earlier, $e.sig$ (event significance) is a function on $e.v$.

Example 4 *(Example 3 continued) Let* EC_0 *be the event cluster given in Example 2, and* EC_1 *be the event cluster given in Example 3, then the combined event cluster* $EC = EC_0 \oplus EC_1$ *is* { (PBV, 21 : 05 : 31, {0, 1}, 1, 0.7, male, 3283), (PBV, 21 : 05 : 31, {0, 1}, 1, 0.4, female, 3283), (PBV, 21 : 05 : 31, {0, 1}, 1, 0.7, {male, female}, 3283), (PBV, 21 : 05 : 31, {0, 1}, 1, 1, obscured, 3283), (PBV, 21 : 05 : 31, {0, 1}, 1, 1, {male, female, obscured}, 3283) }*, and the corresponding mass values are* $m(\mathsf{male}, 3283) = 0.478$*,* $m(\mathsf{female}, 3283) = 0.376$*,* $m(\{\mathsf{male}, \mathsf{female}\}, 3283) = 0.059$*,* $m(\mathsf{obscured}, 3283) = 0.054$*, and* $m(\{\mathsf{male}, \mathsf{female}, \mathsf{obscured}\}, 3283) = 0.033$*.*

3.4 Event Flow

Event models usually use the concept *Event History* (EH) to describe the set of all events whose occurrences fall between a certain period of time. However, in our framework, given a set of event clusters, we first carry out events combination, and then retain only the combined event clusters. So what we have is not a *history*, because of this, we call it an *event flow* and denote it as EF. We use $EF_{t_1}^{t_2}$ to represent a set of combined event clusters whose occurrences fall between t_1 and t_2. Since an event flow contains the combined events, to some extent, we have already considered the *opinions* (of the original events) from different sources.

Example 5 *(Example 4 continued) Let* EC_0 *and* EC_1 *be the event clusters given in Example 2 and Example 3, respectively,* EC *be the combined event cluster of* EC_0 *and* EC_1 *in Example 4. Let* EC_2 *be the event cluster for describing a person loitering in a bus (for simplicity we also omitted other details) given by source 2 as* (PL, 21 : 05 : 37, 2, 1, 1,
3283, DriveCabin), (PL, 21 : 05 : 37, 2, 1, 0.7, 3283, StairWay)*, and* (PL, 21 : 05 : 37, 2, 1, 0.3, 3283, Seated) *and the mass values given by source 2 are* $m_2(3283, \mathsf{DriveCabin}) = 0.2$*,* $m_2(3283, \mathsf{Stairway}) = 0.1$*, and* $m_2(3283, \mathsf{Seated}) = 0.7$*. Then the event flow is* $[EC, EC_2]$*.*

3.5 Event Inference

Event inferences are expressed as a set of inference rules which are used to represent the relationships between events. In the literature of event models, most rules were defined in a deterministic manner without uncertainty except [15], where rules are defined in a probabilistic way. Simply speaking, rules in [15] are defined as follows: if some conditions of a rule are satisfied, then a certain event E occurs with a probability p, and does not occur with a probability $1 - p$. This type of inference rules is an uncertainty-based extension to the Event-Condition-Action (ECA) paradigm proposed in active databases.

However, this approach ignores situations where a set of events can be inferred due to uncertainty or incompleteness[8].

In this paper, we define our event inference rules which can resolve uncertainty and incompleteness. An inference rule R is defined as a tuple $(LS, EType, Premise, Condition, m_{IEC})$ where:

LS, abbreviated for Life Span, is used to determine the temporal aspect of a rule R [3, 1, 16]. LS is an interval determined by a starting point and an end point, or an initiator and a terminator, respectively. The starting point and the end point are two points in time which can be determined by the event flow that is known at the time a rule is executed. For instance, a starting time point may refer to the occurrence time of a specific event, a prior given time, etc, and an end time point can be the occurrence time of another event, a prior given time, or a time period plus the starting point, etc.

$EType$ is the event type of the inferred event cluster. For example, SAD standing for Shout At Driver is an inferred event type.

$Premise$ is a set of $EType$s that a set of events of such types are used by the rule as prerequisites. For example, to induce an SAD event, we need to have the corresponding PBV, PL (Person Loiter), PS (Person Shout) events[9], hence $Premise = \{PBV, PL, PS\}$. $Premise$ is used to select the premise events for a rule.

$Condition$ is a conjunction of a set of conditions used to select appropriate events from the event flow to infer other events. The conditions in $Condition$ can be any type of assertions w.r.t the attributes of events. For example, let e_1 and e_2 both denote a male loitering event and e_3 denote a person shouting event, then
"$e_1.pID = e_2.pID \wedge e_1.gender = male \wedge e_1.location = e_2.location = DriverCabin$
$\wedge e_2.occT - e_1.occT \geq 10s \wedge e_1.occT \leq e_3.occT \leq e_2.occT \wedge e_3.volume = shouting$"
is a valid $Condition$. Note that for each inference rule, we only select events in the event flow within the lifespan LS (denoted by $LS(EF^t_{t'})$). In addition, the types of events used in the $Condition$ belong to $Premise$. Let the events used in $Condition$ be denoted as $Evn(Condition)$, then $Evn(Condition)$ is an instantiation of $Premise$.

m_{IEC} is the mass function for the inferred event cluster and it is in the form of $(< v^1_1, \cdots, v^1_n, mv_1 >, < v^2_1, \cdots, v^2_n, mv_2 >, \cdots, < v^k_1, \cdots, v^k_n, mv_k >)$ where each mv_i is a mass value and $\sum_{i=1}^{k} mv_i = 1$. We will explain this in detail when discussing rule semantics next.

To differentiate inferred events from other events, we use -1 to denote the source ID of an inferred event cluster and the occurrence time is set as the point in time an inference rule is executed. Moreover, the reliability is set to 1 as we assume that the inference rules are correct.

The semantics of using an inference rule R is interpreted as follows. Given an event flow $EF^t_{t'}$, if $Condition$ of any rule R is true at some time point $t^* > t'$, then an event cluster is inferred from rule R with mass function m_{IEC}. Otherwise, no events are inferred.

[8] In fact, the event model in [15] could be extended so that more than one target event can be inferred, see [17]. However, rules like this still cannot infer events with incomplete information due to the different expressive power between probability theory and DS theory.

[9] Simply speaking, to get a SAD event, we need to check that a person X entered the bus, went to the driver's cabin and then a shout at the cabin was detected.

Formally, for any vector $< v_1^i, \cdots, v_n^i, mv_i >$, if $Condition(LS(EF_{t'}^t)) = true$, we in fact generate an event E_i whose event type is $EType$, source ID is -1, occurrence time is the time of rule execution, reliability is 1, $E_i.v = (v_1^i, \cdots, v_n^i)$ (and $E.sig$ is a function over $E_i.v$), and

$$m_{IEC}(E_i.v) = mv_i, 1 \leq i \leq k$$

For any two rules having the same $Premise$, we consider them from a single *rule cluster*. Intuitively, rules in a rule cluster describe inferences based on the same observations, hence these rules have the same lifespan and the same inferred event type but with different $Condition$ and m_{IEC} values due to the different premise events induced from the observations. In addition, in this framework, if two rules do not have the same $Premise$, then they will not infer the same type of events.

Example 6 *An inference rule R_1 which reports an obscured person loiters at the driver's cabin can be defined as $(LS, EType, Premise, Condition, m_{IEC})$ where*
$LS = [0, E.occT]$ *where E is an event of PL,*
$EType$ *is PD abbreviated for Passenger-Driver,*
$Premise$ *is $\{PBV, PL, PL\}$,*
$Condition$ *is "$e_1.gender = obscured \wedge e_1.pID = e_2.pID = e_3.pID \wedge e_2.location = e_3.location = DriverCabin \wedge e_3.occT \geq e_2.occT + 10s$",*
and m_{IEC} *is* $< \{Stand, Talk\}, 0.7 >, < Leaving, 0.2 >, < hasThreat, 0.1 >$.
A similar inference rule R_2 can be used to report a male loiters at the driver's cabin with $R_2 = (LS', EType', Premise', Condition', m'_{IEC})$ where
$LS' = LS$, $EType' = EType$, $Premise' = Premise$,
$Condition'$ *is "$e_1.gender = male \wedge e_1.pID = e_2.pID = e_3.pID \wedge e_2.location = e_3.location = DriverCabin \wedge e_3.occT \geq e_2.occT + 10s$",*
and m'_{IEC} *is* $< \{Stand, Talk\}, 0.7 >, < Leaving, 0.22 >, < hasThreat, 0.08 >$.
Hence R_1, R_2 are in a rule cluster.

4 Calculation of Event Mass Values

Since events in $Evn(Condition)$ are themselves uncertain, to get the mass value of an inferred event, we need to consider both m_{IEC} and the mass values of events in $Evn(Condtion)$. Here the mass value can be seen as a joint degree of certainty of all events involved (similar to the joint probability in Bayesian networks).

To proceed, first, it is necessary to ensure that the execution of the event model in an application is guaranteed to terminate. That is, in a finite time period, there would be only finite external events, finite domain events, finite inference rules to be triggered (hence finite inferred events). Second, it is also necessary that the execution of the event model is guaranteed to be deterministic. That is, with the same time period, same input events and same set of rules, the resultant event flow (after applying all rules) should be unique. These two issues are discussed in [10] where they can be solved by avoiding cycles in rule definitions and by ranking the rules, respectively.

Now assume that there are no cycles in rules and the rules are ranked. Typically, for a specific inferred event, it can be inferred from more than one rule in a rule cluster

(e.g., a PD event with value $hasThreat$ in Example 6). Hence the mass value should consider all these rules in that rule cluster RC. Since each rule in RC has the same $Premise$, let $Premise = \{ET_1, \ldots, ET_t\}$ be a set of event types, and let $\mathscr{V}_i$ be the corresponding frames of discernment of ET_i. Let $\Omega_{RCE} = \prod_{i=1}^{t} \mathscr{V}_i$ be a joint frame of discernment for the premise event types and Ω_{RCH} be the frame of discernment of the inferred event type. Then we can use evidential mapping to get the mass value of an inferred event. Formally, for each rule R in RC, we set $\Gamma_R^*(e_1^R.v, \cdots, e_t^R.v) = ((v_1^1, \cdots, v_n^1), mv_1), \cdots, ((v_1^k, \cdots, v_n^k), mv_k)$ where $(e_1^R, \cdots, e_t^R) = Evn^R(Condition^R)$ and $((v_1^i, \cdots, v_n^i), mv_i) \in m_{IEC}^R$. For other $(e'_1, \cdots, e'_t)$ where $(e'_1, \cdots, e'_t)$ is an instantiation of $Premise$ but there does not exist a rule R in RC, s.t., $(e_1^R, \cdots, e_t^R) = (e'_1, \cdots, e'_t)$, we set $\Gamma_R^*(e'_1.v, \cdots, e'_t.v) = (\emptyset, 1)$. Therefore, the mass value of inferred event can be obtained using Equation 5.

We can also use Bel and Pl functions to get a plausible interval of inferred events.

Based on the above, here we give an algorithm to calculate event mass in **Algorithm 1**. Note that this algorithm executes when a new observation (hence a set of event clusters are gathered from multiple sources, and a combination is carried out) is obtained.

Algorithm 1 Event Mass Calculation

Input: An event flow EF^t, most recent combined event cluster $EC^{t'}$, all rule clusters RCs in a pre-specified order.

Output: Output a mass value of each inferred event when some rules are triggered.

1: $EF^{t'} \leftarrow EF^t \cup \{EC^{t'}\}$;
2: **for** each rule cluster RC **do**
3: calculate $LS^{RC}(EF^{t'})$;
4: select event clusters in $LS^{RC}(EF^{t'})$ according to $Premise^{RC}$;
5: **for** each selected event clusters $EC_1, \cdots, EC_t$ **do**
6: construct the frames of discernment Ω_{RCE} and Ω_{RCH};
7: **for** each rule R in RC **do**
8: **for** each list of events $e_1, \cdots, e_t$, s.t., $e_i \in EC_i$ **do**
9: **if** $Condition^R$ is satisfied **then**
10: add the contents of m_{IEC}^R to $\Gamma_R^*(e_1, \cdots, e_t)$;
11: **else**
12: set $\Gamma_R^*(e_1, \cdots, e_t)$ to $(\emptyset, 1)$;
13: **end if**
14: **end for**
15: **end for**
16: calculate the mass values of focal elements in Ω_{RCH} using Equation 5;
17: set the mass value of a focal element to the mass value of an inferred event whose $e.v$ is the focal element;
18: **end for**
19: **end for**
20: **return** the mass values for the inferred events.

5 Related Work

Our event definition is similar to that considered in [1, 15, 16] where events are considered significant (w.r.t the specified domain of the application), instantaneous and atomic. The reason why we do not require the events to be significant is that in real applications, we also need to model insignificant events (otherwise we may lose information). For instance, in surveillance applications, up to 99% of the events are just trivial events. Hence, instead of defining events as significant, we introduce a built-in significance value in the representation of events to facilitate subsequent processing.

For the inference rules, in [15], a rule is defined as $(sel^n, pattern^n, eventType, mappingExpressions, prob)$ where sel^n is used to get n events, $pattern^n$ is a conjunction of a set of conditions. However, conditions for $pattern^n$ can only be an equality form as $e.attr_i = e'.attr_j$ or temporal conditions of the forms, $a \leq e.occT \leq b$ or $e.occT < e'.occT$ or $e.occT \leq e'.occT \leq e.occT + c$. Obviously, it can not express conditions like $E.gender = obscured$, $E_1.speed < E_2.speed$, etc, while our $Condition$ can. In addition, a rule in [15] can only provide a single inferred event with a probability $prob$ whilst a rule in our model can provide a set of possibilities with mass values.

Classical deterministic rules are special cases of our rule definition with the inferred event cluster having only one event with a mass value 1 and probabilistic rules are also special cases of our rule definition.

Furthermore, in our model, the notion of event history or event flow is also different from those used in [6, 1, 15, 16] such that our event history/flow takes embedded uncertainty (in fact it contains observations (event clusters) which consist of multiple possible events) while in those models an event history itself is considered deterministic and the uncertainty on event history is expressed as there can be multiple possible event histories. Due to this difference, the rule semantics is totally different from the conditional representation in [15].

Finally, when our event model reduces to the situation considered in [15], it is easy to find that our calculation of event mass reduces to the probability calculation by Bayesian networks in [15].

Proposition 1 *If we consider only one-source, one-inference-target probabilistic case as in [15], then the mass value of inferred event is equivalent to the joint probability obtained by the Bayesian network approach in [15].*

6 Conclusion

In this paper, we proposed an event model which can represent and reason with events from multiple sources, events from domain knowledge, and have the ability to represent and deal with uncertainty and incompleteness. For events obtained from multiple sources, we combined them using Dempster-Shafer theory. We introduced a notion called event cluster to represent events induced from an uncertain observation. In addition, in our model, inference rules can also be uncertain. Furthermore, we discussed how to calculate the mass values of a set of events. This framework has been implemented and evaluated by a bus surveillance case study.

Since in real-world applications, information is frequently gathered from multiple sources, and uncertainties can appear in any part of the applications, our event model can serve as an important foundation for these applications. Domain knowledge is very useful in many active systems, however, it is somehow ignored in the existing event reasoning systems. Our model can also represent and deal with it.

For future work, we want to extend this event model to include temporal aspects of events. First, in some active systems, there is no accurate occurrence time attached with events. Second, some behaviours associated with a time interval such as *a person is holding a knife* is hard to be represented in this event model. In fact, as the instantaneous nature of events in this event model, we can only tell at a certain time point (or a set of successive time points), the person is holding a knife.

Acknowledgement This research work is partially sponsored by the EPSRC projects EP/D070864/1 and EP/E028640/1 (the ISIS project).

References

1. A. Adi and O. Etzion. Amit - the situation manger. *VLDB J.*, 13(2):177–203, 2004.
2. Bsia. Florida school bus surveillance. In *http://www.bsia.co.uk/LY8VIM18989_action;displaystudy_sectorid;LYCQYL79312_caseid;NFLEN064798*.
3. S. Chakravarthy-S and D. Mishra. Snoop: an expressive event specification language for active databases. *Data and Knowledge Engineering*, 14(1):1–26, 1994.
4. A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *The Annals of Statistics*, 28:325–339, 1967.
5. B. Abreu et al. Video-Based Multi-Agent Traffic Surveillance System. In *Proc. IEEE Intel. Vehi. Symp.*, Lecture Notes in Computer Science, pages 457–462. SPIE, 2000.
6. N. H. Gehani, H. V. Jagadish, and O. Shmueli. Compostite event specification in active databases: Model & implementation. In *Proc. of VLDB*, pages 327–338, 1992.
7. W. Liu, J. G. Hughes, and M. F. McTear. Representating heuristic knowledge in d-s theory. In *Proc. of UAI*, pages 182–190, 1992.
8. J. D. Lowrance, T. D. Garvey, and T. M. Strat. A framework for evidential reasoning systems. In *Proc. of 5th AAAI*, pages 896–903, 1986.
9. J. Ma, W. Liu, P. Miller, and W. Yan. Event composition with imperfect information for bus surveillance. In *Procs. of 6th IEEE Inter. Conf. on Advanced Video and Signal Based Surveillance (AVSS'09)*, pages 382–387. IEEE Press, 2009.
10. N. W. Patton. *Active Rules in Database Systems*. Springer-Verlag, 1998.
11. Gardiner Security. Glasgow transforms bus security with ip video surveillance. In *http://www.ipusergroup.com/doc-upload/Gardiner-Glasgowbuses.pdf*.
12. G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
13. C. F. Shu, A. Hampapur, M. Lu, L. Brown, J. Connell, A. Senior, and Y. Tian. Ibm smart surveillance system (s3): a open and extensible framework for event based surveillance. In *Proc. of IEEE Conference on AVSS*, pages 318–323, 2005.
14. L. Snidaro, M. Belluz, and G. L. Foresti. Domain knowledge for surveillance applications. In *Proc. of 10th Intern. Conf. on Information Fusion*, 2007.
15. S. Wasserkrug, A. Gal, and O. Etzion. A model for reasoning with uncertain rules in event composition. In *Proc. of UAI*, pages 599–608, 2005.
16. S. Wasserkrug, A. Gal, and O. Etzion. Inference of security hazards from event composition based on incomplete or uncertain information. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1111–1114, 2008.
17. S. Wasserkrug, A. Gal, O. Etzion, and Y. Turchin. Complex event processing over uncertain data. In *Proc. of DEBS*, pages 253–264, 2008.