DOI 10.1007/s11055-023-01360-5
Neuroscience and Behavioral Physiology, Vol. 52, No. 8, October, 2022

A Conceptual Model of Sensor System Ontology with an Event-Based Information Processing Method

E. O. Cherskikh

UDC 004.042

Translated from Sensornye Sistemy, Vol. 36, No. 2, pp. 124–135, April–June, 2022. Original article submitted October 7, 2021. Accepted November 9, 2021.

The aim of the present work was to analyze existing methods of event-based processing of information both at the level of the sensors of sensor systems and at the level of the system as a whole. To achieve this goal, sensors with an event-based principle of operation are considered, the most widely used of these being cameras and dynamic sound sensors. For other types of sensors with continuous data transmission, event processing methods using ontologies that work with homogeneous and heterogeneous sensor systems are considered. Methods for separating events from the general flow of data coming from sensors and methods for creating complex events are identified. The most popular way to isolate an event from a stream of data coming from sensors is to match the data received from sensors with a sample. To create complex events, most of the studies addressed here use templates and specialized systems for processing complex events. Drawbacks of these methods are highlighted and an approach to eliminating them is proposed, based on developing an editable ontology for the sensor system able to take account of the consequences of adding or removing sensor nodes.

Keywords: sensor systems, events, event sensors, information processing.

Introduction. Sensor systems used in robotic devices (RD) to interact with the environment, as well as those used in vision systems and audio information processing systems, often use sensors whose data are fed to a central or distributed control device in a continuous stream. One of the main problems of such systems is the large amount of data received. This results in the needs for significant time, computational, and energy resources for data transmission and processing. This problem can be solved by using sensors with an event-based operating principle or methods of event-based processing the information coming from sensors transmitting continuous datastreams.

The aims of the present work were to analyze existing solutions for event-based data processing in RD sensor systems and to create our own conceptual ontological model for event-based extraction of data from the nodes of a sensor system. Sensors with an event-based operating principle will be considered, along with methods for processing streams of information coming from the sensors and identifying simple events formed directly from the data.

Event processing methods allow significant events to be extracted from the overall datastream, while complex events can be constructed from multiple simple events, which can be useful for systems including different types of sensors. The basis for these methods is an ontology, i.e., a conceptual scheme consisting of a data structure. The ontology contains classes, objects, and their properties, as well as object relationships and constraints. Event-based methods based on an ontology, can not only extract events, but can also provide the user with the opportunity to interact with the sensor system by searching for the required device on request, generating a user event, or notifying the user about the occurrence of events.

Systems with Sensors with an Event-Based Operating Principle. Sensors with an event-based operating principle are used in technical vision systems in RD for tracking static and dynamic objects and their recognition, for locating the

St. Petersburg Federal Research Center, Russian Academy of Sciences; St. Petersburg Institute for Informatics and Automation, Russian Academy of Sciences, St. Petersburg, Russia;
e-mail: cherskikh.e@iias.spb.su.

RD and mapping the environment. Sensors of this type are also used in audio information processing systems and to detect hazardous contaminants in air. In the field of machine vision, video cameras are used as sensors with an event-based operating principle. Conventional cameras capture changes at regular intervals, polling all the pixels and recording the light intensity readings taken over a certain period. Event-based cameras respond to changes in the brightness of individual continuously operating pixels. Information of importance in the use of these cameras are the moving edges of the objects being tracked. Knowing the features of this type of information can help reduce the computational resources for its processing. We will consider some existing solutions in this area.

For example, a sensor such as the DAVIS (Dynamic and Active Pixel Vision Sensor) dynamic pixel sensor [Mueggler et al., 2017] includes a conventional global shutter camera and an event-based sensor. The output of the sensor consists of a stream of asynchronous brightness changes, i.e., events and synchronous grayscale camera frames. Events are timestamped and dispatched asynchronously with respect to the time at which they occur. Each event e is a tuple (x, y, t, p), where × and y are the pixel coordinates of the event, t is the timestamp of the event, and $p = \pm 1$ is the polarity of the event, which is the sign of the change in brightness. The sensor has a resolution of 240 × 180 pixels, does not require calibration, can be used both indoors and outdoors, and can be used as a vision system for mobile PCs.

Tracking a moving object with a DAVIS sensor [Liu et al., 2016] is performed in three steps: first, regions of interest are generated, likely target locations are detected using a convolutional neural network and classified as foreground or background, and a multi-particle filter infers the target location of the tracked object from the regions of interest. In the experiment presented, a wheeled robot bearing a sensor follows another manually controlled wheeled robot.

A solution similar in functionality is a system based on an event camera [Glover and Bartolozzi, 2017] used on the iCub anthropomorphic robot to track moving objects. The authors propose using a multi-particle filter to provide resistance to the temporal fluctuations that occur when the camera and target move at different speeds, which can lead to loss of visual information.

Ghosh et al. [2014] implemented not only object tracking, but also real-time identification using an event-based camera and a convolutional neural network. The sensor reacted only to moving objects, ignoring static ones. Preprocessing is performed by a noise filter, after which the space-time interval of the region of interest is determined, and the different bursts contained in this area are then converted into a static classified image. The system tracks and distinguishes between cars, bicycles and pedestrians on the road, and is also capable of detecting and identifying household items and their orientations relative to the camera.

With the aim of recognizing moving objects, Ceolini et al. [2020] also proposed combining multiple sensors performing the functions of reading an electromyographic signal and visual information. To overcome computational limitations, the authors proposed using neuromorphic technologies that allow real-time data processing: an event camera and two neuromorphic platforms, Loihi and ODIN + MorphIC. Worn directly on a person's forearm, the electrodes of the Myo electromyography sensor detect signals from forearm muscle activity, and the data acquired are sent to an external electronic device. The data represent of a set of five hand gestures recorded using two sensory modalities: muscle activity from Myo and the visual camera input. Data from the sensors are combined and gestures recognized using a neural network, and as this has a limited number of neurons, the camera input was limited to 40 × 40.

An event-based camera can also be used to recognize human gait [Sokolova and Konushin, 2019]. The proposed algorithm for processing data received from the camera consists of four steps: visualization of the flow of events; detection of a human figure; optical flow estimation; evaluation of the person's posture. The stream of events generated is virtualized to allow processing analogous to that used with standard video frames. An event-based image is produced by creating a time window of defined length and summing all the events occurring in each individual pixel during a given time interval. Various detection methods are applied to the images acquired and the influence of the presence of various body parts on the recognition probability is evaluated. The applicability of event-based recognition methods and accuracy levels of over 98% have been reported.

The use of an event camera for scene reconstruction in a non-deterministic environment was implemented using a 3D reconstruction method [Kim et al., 2016] based on the data obtained from the camera, using three independent probabilistic filters, each of which evaluates camera movement, the intensity gradient of the logarithmic scene, and the inverse scene depth relative to the keyframe. The authors indicate that this method allows motion tracking simultaneously with reconstruction of an arbitrary scene based on a video stream without the use of additional sensors.

Event processing of audio information can be carried out by means of dynamic audio sensors (DAS). The sensor is a binaural system of silicon cochleas designed for spatial listening and analysis of auditory scenes. DAS report the outputs of active nodes only through asynchronous digital events.

The authors of this study [Li et al., 2012] used neuromorphic silicon cochleas with 64 frequency channels and 512 output neurons to identify people speaking in real time. Auditory features are extracted from the output signal of the cochlea and consist of fading histograms of intersignal intervals and channel activity distributions. The feature vectors are then classified by a linear support vector machine and then the speaker is identified. The authors presented two methods – in the first, features were calculated for each 100-msec time interval only if the events in this interval exceeded a predetermined threshold, while in the second method, the feature vector was calculated, regardless of the duration of the time interval, when the number of events exceeded the set threshold value. Both methods require finding a compromise between the size of the time interval and the delay in making a decision for optimal performance of the system as a whole.

A similar sensor was used by Anumula et al. [2018], who developed a probabilistic model for sound localization. Each cochlea has two separate 64-stop cascaded filter banks driven by two microphones spaced close to each other. The frequency selectivity of the 64 channels is in the range 100 Hz to 10 kHz. Each channel has four neurons. The output from the microphones is fed into a cascaded filter bank that simulates the basilar membrane, inner hair cells, and spiral ganglion cells. A sound event is localized by calculating the time difference between the signals from two sensors, which is estimated by calculating the time difference between the nearest event of one sensor and the nearest event in the same frequency channel but from another sensor.

The nodes of a sensor system for detecting hazardous impurities in air [Somov et al., 2011] are boards bearing gas sensors. The event-based operating principle is implemented in software using pulse width modulation (PWM). Each network node contains a homogeneous semiconductor gas sensor, a microcontroller, a ZigBee module, and a battery pack (three AA 1.5 V, 3000 mA·h batteries). The current consumed by the node is 80 mA. The measurement cycle lasts about 1 sec, excluding the time spent on data transfer if an emergency occurs. The sensitive layer of the sensor heats up to about 500°C when a measurement is required. The resulting value is compared with two preset thresholds set for the sensor node. If the second threshold is exceeded, an event signal is transmitted over the data channel.

Sensor systems containing sensors operating on an event-based operating principle are used in machine vision systems, for processing audio information, and for detecting impurities hazardous to humans in air. Neural networks, multi-particle filters, and noise filters are used to track objects, recognize objects, map the environment, and localize RD in stationary and dynamic environments. Audio information processing systems use dynamic sound sensors using support vector machines and time difference calculations to localize sound events. The event-based operating principle of the sensor system nodes for detecting hazardous impurities in air is implemented using PWM.

Various algorithms are used to resolve problems in the recognition and tracking of objects using data obtained directly from event sensors. For example, Belbachir et al. [2007] presented an algorithm for processing data received from an event camera in real time. Processing of incoming data included object detection, noise removal, normalization of received data, and an object recognition function. Experimental confirmation was obtained that, using the circle selection and orientation assessment included in the object recognition function, the proposed method allows recognition of objects of regular shapes: cube, spheres, hexagons. The complexity of the proposed algorithm was proportional to the number of events.

Algorithms are also used to track objects [Mueggler et al., 2014; Ramesh et al., 2018]. Both of these algorithms work on the basis of data received from a DAVIS sensor. The algorithm described by Mueggler et al. [2014] tracks segments defining the edges of an unmanned aerial vehicle to track and evaluate its position in the air based on known patterns, and identifies an event when a similarity to a predefined pattern is found. The system described by Ramesh et al. [2018] uses a distinctive representation of objects being tracked, along with online learning, and also detects and retracks an object when it returns to the camera's field of view. The system uses local sliding window technology to ensure reliable operation in scenes with complex backgrounds.

Rebecq et al. [2017] presented a visual odometry algorithm, for mapping the environment to calculate the position and orientation of the cameras. Observation and detection of the edges of objects are performed from two points: from the first point, a reference position is tracked (a three-dimensional map of the scene obtained by combining a small number of events into a map of boundaries), while the image from the second point consists of a projected semi-dense 3D scene map consistent with the known position of the dynamic pixel sensor, a DAVIS [Dynamic and Active Pixel Vision Sensor] device. The algorithm presented allows the position and orientation of the camera to be calculated and a semi-dense 3D map of the environment to be produced, even though the edges parallel to the movement of the event camera are not recorded.

Power consumption is minimized in systems with large numbers of system nodes with resultant increases in system operation time. Cao et al. [2005] proposed a protocol for scheduling the operation mode of nodes when detecting events in real time. The system is optimized for rare event detection and allows for a trade-off between event detection latency and autonomous operating duration. The ability to identify incorrect sensor readings, identify correlations between the incoming readings, and calculate the characteristics of the events that have occurred are implemented. The system is built on the principles of semantic description, including determination of the relative importance of intermediate events. An event is considered critical if the failure rate is high. This parameter is also used to distinguish between event occurrences and false triggerings.

Event-based cameras reduce the impact of the problem of processing a continuous stream of data by addressing only pixel changes in successive events that can be observed with high temporal resolution. Due to the low latency and high temporal resolution, event-based sensors are promising for high-speed mobile PCs. When there is a large number of sensors in the sensor node, it becomes necessary to find a compromise between the number of missed events, response time, and power consumption in order to find the optimal configuration of the sensor system. In such cases, processing of information coming from event sensors can be performed using algorithms whose input data are already events. The use of event sensors in robotic systems benefits from lower power consumption, higher time resolution, and reduced computational load compared to sensors that transmit data continuously. Application of this operating principle requires the architecture of the sensor system and specialized data processing and filtering algorithms to be developed but provides the ability to respond only to certain events with high temporal resolution and energy-efficient operation, as compared with the nodes of sensor systems that transmit data continuously. In addition to the methods considered for implementing the event-based operating principle directly in the nodes of sensor systems, this principle can be used at the stage of processing data coming from sensors.

Next, we will consider existing algorithms and methods for processing data from sensors and sensor systems in general using ontologies.

Event-Based Information Processing Methods. This section addresses ontological methods for processing data received continuously from various types of sensors. Methods of this type are used to recognize events in continuous datastreams. Ontological methods are used to unload the central computing device of the sensor system or when using a distributed method processing the data from the system as a whole. Events can be classified and presented in separate parts to form other, more complex events. Thus, Dunkel [2009] proposed a network architecture that providing for analysis and processing of event streams in real time to identify significant events in the datastream coming from the nodes of a sensor network. The network contained several types of agents that perform the functions of analyzing unprocessed events, diagnosing and generating state events, and planning actions. This approach is based on the use of ontologies that allow structural properties of event types and constraints between them to be represented. Events are divided into state events and action events. Each event contains an identifier, timestamp, ID, and sensor data. Pattern matching and event processing are performed by event processing nodes that monitor event streams. These nodes filter, separate, and create more complex events from multiple simple ones. As the amount of incoming data is large, each event has an expiration date, after which it is removed from the system.

As in the work previous discussed in this review, a method of processing incoming events based on sensor system ontology [Taylor and Leidinger, 2011] is used to recognize complex events consisting of several simple user-defined events. Each of the composite events contains an observation (here the term is used to describe five different kinds of composite observations). An atomic observation is a description of a simple event within the definition of a complex one; it contains information for programming a selected sensor and the definition of an event trigger. A complex event is built using logical operations, groupings of simple events, complex event processing (CEP), and datastream management systems (DSMS). When a user requests information about an event of interest through the interface, the middleware processes the request and generates commands for the CEP server. The server monitors the selected datastreams and generates alerts that are delivered to the specified users when an event occurs. Certain events are stored in an ontology whose data are used to recognize incoming events in the future. The description of each event consists of two main parts: the warning that will be activated if the event was recognized and the definition of the event itself. In this system, an ontology is used to define events that include multiple sensor datastreams, and fragments of it can be replaced with a different command language if adaptation to a different event handler is required.

A sensor system consisting of a set of wireless sensors, actuators, and a computer-based controller, also operates in a similar scenario [Mazo and Tabuada, 2011]. However, unlike the situation in the reports discussed above, every time a certain event occurs, the system terminates the work cycle to reduce the frequency of controller updates. Conditions should be set depending on the information received at each of the nodes of the system. When any of these conditions is violated in a node, it informs the computing device. After receiving such an event, the computing device requests new measurements, updates the control signals, and sends new commands to the actuation nodes.

Processing of a continuous stream of events is described by Bhargavi et al. [2010]. One of the pyroelectric infrared motion sensors of the sensor system in this study is used as an event initializer. When motion is detected, the camera is triggered to capture an image. The captured images are sent to the server via a wired network. Incoming datastreams are processed by CEP according to predefined rules. Events are defined using the ESPER event engine, which continuously sends predefined requests to continuous event streams. Once an event is detected by a node, it generates a packet containing the sensor ID, RFID tag, pyroelectric infrared sensor readings, time, and other useful information. Since the number of nodes is large and the events are random and multiple, the dataset generated is large. Data arriving at the server are cleaned, after which relationships between existing events are discovered to build more complex events. The architecture of the proposed sensor network has several levels: the data sources level, the data collection level, and the data filtering level. Building events from incoming data occurs by matching with an existing sample. Comparison of output data with existing or statistical data occurs in real time.

Data matching to implement the event-based principle of system operation was also used by Kasi et al. [2021]. Processing of events generated by a heterogeneous sensor network was implemented by means of an ontological knowledge base in each of the nodes of the system. The ontology fragments in each sensor node identify the data routed through the sensor network. Unlike previous work, the matching algorithm used here was capable of handling a changing knowledge base. The node distinguishes between three types of incoming events: sensed, shared, or forwarded. Typing was used because different operations are performed on each event type. The actions specified by the rule engine could be: discard, share or forward event. When the incoming fact does not match the rules then the event is discarded, i.e., event filtration occurs. When a fact fully matches any of the available rules, the event is forwarded to the gateway node. However, if a partial match is found then the event is shared with the relevant sensor node for further processing.

Some of the methods are based on existing ontologies such as SSN [Lefort et al., 2011], Event [Yves and Samer, 2007], FOAF [Brickley and Miller, 2014], Time Ontology [Hobbs and Pan, 2017], Geo Ontology [Brickley, 2003], and MA-Ont [Thierry, 2012], which are used to describe sensors, events, temporal properties, and resources, and to combine various descriptions of media resources.

These ontologies may not have a sufficient set of properties to perform any task but can be used as the basis of other ontologies. Rinne et al. [2013] present an event processing system based on SSN, DUL, and Event-F ontologies. Events are identified in the datastream coming from sensors by using patterns and timestamps, which are described by a set of individual properties. The sample of sensor-generated data detected by the event processing system triggers the creation of an event object, which in turn describes the actual event that occurred. The main event has several component objects, i.e., sub-events. SPARQL events can be queried against the most common four-event query patterns to compose complex events. The proposed structure does not require OWL reasoning per se, but gives the opportunity to reason over the structures, using transitivity and inverse properties.

Ontology also allows data received from various devices in a system to be structured. Kuznetsov and Buzunova [2018] presented an ontology of a lighting system containing several classes and instances, some of which were used to describe the sensor devices connected to the system. The paper describes a basic ontology that needs to be refined on the basis of existing ontologies such as OntoSensor [Shaukat et al., 2017] and SSN. The main purpose of applying the ontology in this system was for an agent to determine information exchange in a system of participants.

The ontology used in a museum [Khaidarova et al., 2019] includes an element containing data from sensors for temperature, humidity, and room light. Some of the entities in the ontology are borrowed from FOAF. The ontology presented allows consultative and reference tasks to be resolved, along with tasks consisting of monitoring and regulating microclimate parameters. An event is a deviation in microclimate parameters, whose standard values are stored in the ontology; the system sends control signals to the microclimate control device or employees of the institution. Information processing is carried out in pseudo-real time.

Extension of the SSN and MA-Ont ontologies [Lee et al., 2012] and the recording of events from the sensors of a multimedia sensor system [Angsuchotmetee et al., 2020] were performed using the MSSN-Onto ontology. One of the authors' goals was to ensure syntactic and semantic compatibility to facilitate the process of event detection. The authors present the results of a simulation in which the sensor system has up to 500 multimedia sensors and centralized control and is used to monitor conference participants located in the same room. Processing and indexing of incoming datastreams is done by a separate module to map them to MSSN-Onto. Each of the streams is decoded and indexed according to low-level features (visual, audio, or motion descriptors), and thus various types of data (audio, video, images, scalar values) are indexed using the MSSN-Onto data model. Event detection and handling of user requests are performed by the event handling module. In total, ten complex events from the scenario are proposed: the beginning of the meeting; presentation of the daily schedule; presentation of reports; use of smart boards; slide changes; events of simultaneous discussion by several participants; participants' arrival and departure times; report of the results of the meeting; departure of all participants from the room. The system has a significant disadvantage: the lack of flexibility makes it impossible to use it in a room with a different infrastructure, since it is impossible to add new or remove old sensors without reconfiguring the entire system. The possibility of adding heterogeneous sensors without reconfiguring the system exists and is implemented in the A3ME ontology [Herzog et al., 2008]. This ontology is a basic concept hierarchy for classification, self-description, and device discovery, but there is no event handling in A3ME. MSSN-Onto event detection has the limitation of being able to recognize new events if the relevant knowledge and user events are provided in the same framework.

Extensions of the existing SNN, Event, Time Ontology, FOAF and Geo Ontology ontologies has also been carried out by Belkaroui et al. [2018]. The ontology presented for events for Wine Cloud, implemented in Protégé 8, is used to extract events from data generated by the sensors of a heterogeneous distributed sensor system used in vineyards. An event is defined as a tuple of six values, meaning: that the action set in the event occurs; the period of time during which the event lasted; event location; conditions that caused the event; the combination of elements that characterize the event; the main participants in the event. The events that can happen are predetermined. The authors divided them into four groups: vine diseases, the presence of pests, physiological risks, and climatic risks. The event detection system consists of two main components: the event information service detects relevant information, identifies its nature, and gets values for their properties; The Data Mart API serializes objects using the Wine Cloud ontology dictionary, extracts knowledge, and sends it to the central system component used to store knowledge. It is also possible for the user to ex-

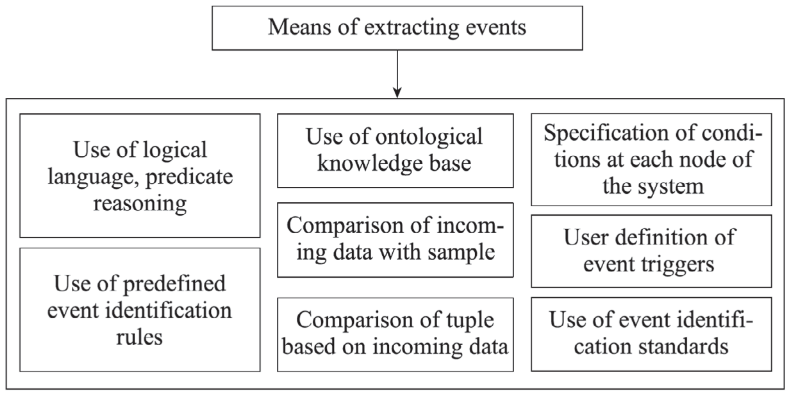

Fig. 1. Means of extracting events from an incoming data sensor stream.

tract knowledge via SPARQL queries, such as: searching for events that occur in the same period, extracting the factors of an event, or searching for events that have one specific cause. The scope of the proposed ontology is currently limited to events that can occur during the life cycle of a grapevine.

In contrast to the previous works reviewed here, which focused on extracting events from datastreams, Nawaz et al. [2019] proposed predicting future events, as well as modeling complex events using CEP and time-varying actions performed through computation and complex event processing. The proposed structure uses two types of knowledge bases: main and actions. The first of these contains rules for defining and recording complex event patterns, while the second defines all alternative actions that can be taken at a given point in time to avoid the predicted unwanted event. To capture uncertain events such as noisy sensor data, hybrid predictive reasoning is used that has both logical and probabilistic reasoning capabilities. In addition, when the incoming datastream contains incomplete, inaccurate, or missing information about some specialized complex event, probabilistic inference is used to determine the possible state, after which the reasoning engine predicts the failure of the process.

The use of ontology makes it possible to reduce the amount of sensor output data at the processing stage. The user can query the system for predefined patterns of the most common events, search for events, and retrieve specific facts. An ontology can provide syntactic and semantic interoperability of multimedia sensors by indexing different types of data. Some the works reviewed here extend existing ontologies by adding additional features. This solution avoids the need to create an ontology from scratch but adds a set of specific properties to an existing ontology to perform the tasks required.

Classification of event creation methods. Most of the sensor systems discussed here have the ability to generate complex events automatically. It is also possible for the user to generate and request an event through the system interface using a SPARQL query. Figure 1 shows how events can be extracted from streams coming from data sensors.

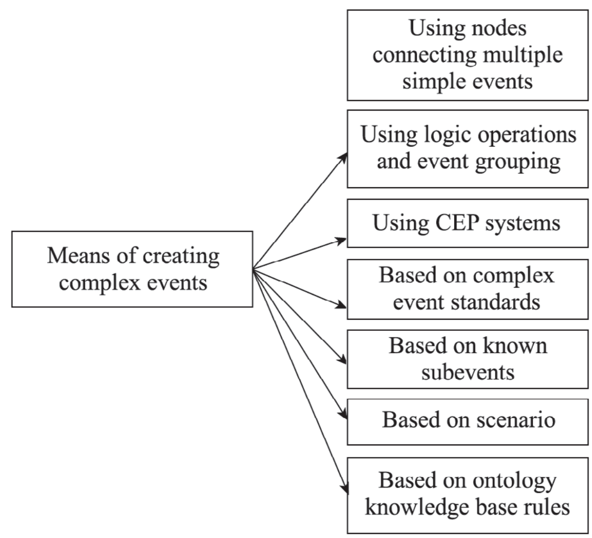

Fig. 2. Means of creating complex events.

Analysis results indicate that the most popular of the methods considered is the comparison of data obtained from sensors with a sample. To reduce the amount of data at the event selection level, the selected event is deleted from the system after a certain time has elapsed or when the entire system operation cycle is completed. Methods for creating complex events are shown in Fig. 2.

The most popular way to create complex events is to use the CEP system and complex event templates. The disadvantage of CEP is the inability to combine multiple distributed data sources and perform predictive reasoning. The systems discussed here also have some disadvantages: limitation on the rapid addition or removal of new devices to the system without a complete reconfiguration; impossibility of use in applications and systems other than those for which they were created; limitations in event detection and automatic addition of new types of simple and complex events. Most systems are static and do not provide for sensor system node failures, though incomplete or noisy information can be used to predict failures through the use of probabilistic inference in the system [Nawaz et al., 2019].

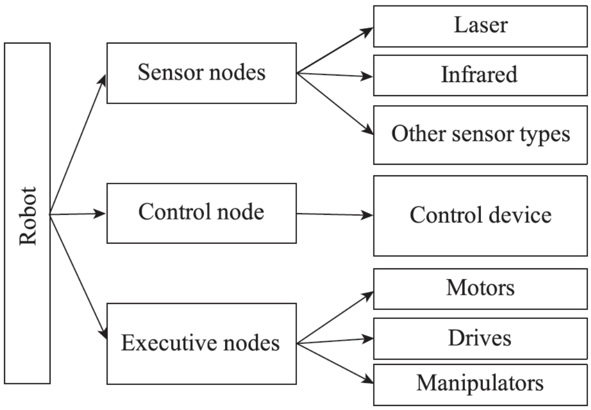

Fig. 3. Ontology model for RD.

Conceptual model of the proposed ontology. The use of sensors with an event-based operating principle as part of sensor networks is not widespread. Most networks contain many sensors that transmit data in a continuous stream. Methods using ontologies have been addressed with the aim of extracting events from datastreams. Given the disadvantages of these methods noted above, there is a need to develop a multi-level architecture for sensor systems in which one of the levels is represented by an editable ontology that implements the following features: identification of the types of connected or disconnected nodes of the sensor system, detection of node failures and allowing for the consequences of such failures for the PC, implementation of SPARQL queries, and extraction of events. Figure 3 presents a conceptual model of the ontology proposed for implementation in Protégé.

This ontology has one class and several subclasses, each of which contains defined objects. Object properties describe objects and the relationships between them. Data properties are proposed to describe the numerical values of objects. For each object added to the ontology, there is a set of predefined object properties and data properties that allows the Protégé algorithm to determine its subclass and relationships with other objects in the ontology. A sensor node failure or removal from the system can be detected using a SPARQL query. Hierarchical complexity, the necessary set of properties of objects, and the possibility of reasoning based on them will allow RD to understand what actions are possible when connecting a new type of node or disconnecting some of the nodes already in the system. SPARQL queries will allow information to be obtained on the state of the environment based on events that are significant for the RD at the ongoing time point. Use of a neural network is proposed to process data coming from sensors. These improvements will allow an RD to receive information about environmental events, monitor the state of the sensor network, and, accordingly, know which types of individual nodes are currently available and which are not.

Conclusions. This paper analyzes sensor systems for RD with sensors operating on the event-based operating principle. It also reviews methods of event processing of information received from sensors not operating on this principle. Methods for highlighting events for both types of sensor are identified. Methods for creating complex sensor system events using ontology-based methods are considered. The analysis is used as the basis for proposing a conceptual ontology model for use in sensor systems with sensors from which a continuous datastream is received. The model takes into account the possibility of connecting or disconnecting new nodes, including failures, and allows the PC to understand the state of the sensor system, extract events which are important at any given time, and execute SPARQL queries.

Further research will be aimed at developing an ontology based on the proposed conceptual model and applying it to extract events from sensor datastreams of the sensor systems of mobile robotic platforms [Saveliev et al., 2019; Vatamanyuk and Saveliev, 2017].

The authors of this article confirm that they have no conflicts of interests to report.

REFERENCES

Angsuchotmetee, C., Chbeir, R., and Cardinale, Y., “MSSN-Onto: An ontology-based approach for flexible event processing in Multimedia Sensor Networks,” *Fut. Gener. Comp. Syst.*, **108**, 1140–1158 (2020), https://doi.org/10.1016/j.future.2018.01.044.

Anumula, J., Ceolini, E., He, Z., et al., “An event-driven probabilistic model of sound source localization using cochlea spikes,” *IEEE Int. Symp. on Circuits and Systems (ISCAS)* (2018), pp. 1–5, https://doi.org/10.1109/iscas.2018.8351856.

Belbachir, A. N., Litzenberger, M., Posch, C., and Schon, P., “Real-time vision using a smart sensor system,” *IEEE Int. Symp. on Industrial Electronics* (2007), pp. 1968–1973, https://doi.org/10.1109/ISIE.2007.4374909.

Belkaroui, R., Bertaux, A., Labbani, O., et al., “Towards events ontology based on data sensors network for viticulture domain,” in: *Proc. 8th Int. Conf. on the Internet of Things* (2018), pp. 1–7, https://doi.org/10.1145/3277593.3277619.

Bhargavi, R., Vaidehi, V., Bhuvaneswari, P. T. V., et al., “Complex event processing for object tracking and intrusion detection in wireless sensor networks,” in: *11th Int. Conf. on Control Automation Robotics & Vision* (2010), pp. 848–853, https://doi.org/10.1109/ICARCV.2010.5707288.

Brickley, D. and Miller, L., *FOAF Vocabulary Specification 0.99* (2014), http://xmlns.com/foaf/spec/URL, acc. Oct. 20, 2021.

Brickley, D., *Geo Ontology* (2003), https://www.w3.org/2003/01/geo/#development, acc. Oct. 20, 2021.

Cao, Q., Abdelzaher, T., He, T., and Stankovic, J., “Towards optimal sleep scheduling in sensor networks for rare-event detection,” in: *IPSN 2005. 4th Int. Symp. on Information Processing in Sensor Networks* (2005), pp. 20–27, https://doi.org/10.1109/IPSN.2005.1440887.

Ceolini, E., Frenkel, C., Shrestha, S. B., et al., Hand-gesture recognition based on EMG and event-based camera sensor fusion: A benchmark in neuromorphic computing,” *Front. Neurosci.*, **14**, 637 (2020), https://doi.org/10.3389/fnins.2020.00637.

Dunkel, J., “On complex event processing for sensor networks,” *Int. Symp. on Autonomous Decentralized* Systems (2009), pp. 1–6, https://doi.org/10.1109/ISADS.2009.5207376.

Ghosh, R., Mishra, A., Orchard, G., and Thako, N. V., “Real-time object recognition and orientation estimation using an event-based camera and CNN,” *IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings* (2014), pp. 544–547, https://doi.org/10.1109/BioCAS.2014.6981783.

Glover, A. and Bartolozzi, C., "Robust visual tracking with a freely-moving event camera," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)* (2017), pp. 3769–3776, https://doi.org/10.1109/IROS.2017.8206226.

Herzog, A., Jacobi, D., and Buchmann, A., "A3ME – an Agent-Based middleware approach for mixed mode environments," *2nd Int. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies* (2008), pp. 191–196, https://doi.org/10.1109/UBICOMM.2008.78.

Hobbs, J. R. and Feng Pan, *Time Ontology* (2017), https://www.w3.org/TR/owl-time/, acc. Oct 20, 2021.

Kasi, M. K., Hinze, A., Legg, C., and Jones, S., "SEPSen: Semantic event processing at the sensor nodes for energy efficient wireless sensor networks," in: *Proc. 6th ACM Int. Conf. on Distributed Event-Based Systems*, 119–122 (2021), https://doi.org/10.1145/2335484.2335497.

Khaidarova, R. R., Konev, A. S., Lapaev, M. V., and Bondarenko, I. B., "An ontological approach to automating the processes of microclimate control and safety of museum objects," *Nauchn. Tekhn. Vestn. Inform. Tekhnol. Mekhan. Opt.*, **19**, No. 1, 118–125 (2019).

Kim, H., Leutenegger, S., and Davison, A. J., "Real-time 3D reconstruction and 6-DoF tracking with an event camera," in: *Europ. Conf. on Computer Vision* (2016), pp. 349–364, https://doi.org/10.1007/978-3-319-46466-4_21.

Kuznetsov, B. F. and Buzunova, M. Yu., "Ontology of a street lighting control system," in: *Climate, Ecology, and Agriculture of Eurasia, Conference Proceedings* (2018), pp. 217–222.

Lee, W. and Bailer, W., *Ontology for Media Resources 1.0* (2012), https://www.w3.org/TR/mediaont-10/, acc. Sept. 9, 2021.

Lefort, L., Henson, C., and Taylor, K., *Semantic Sensor Network XG Final Report* (2011), http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/, acc. Oct 20, 2021.

Li, C. H., Delbruck, T., and Liu, S. C., "Real-time speaker identification using the AEREAR2 event-based silicon cochlea," *IEEE Int. Symp. on Circuits and Systems(ISCAS)* (2012), pp. 1159–1162, https://doi.org/10.1109/iscas.2012.6271438.

Liu, H., Moeys, D. P., Das, G., et al., "Combined frame- and event-based detection and tracking," *IEEE Int. Symp. on Circuits and Systems (ISCAS)* (2016), https://doi.org/10.1109/iscas.2016.7539103.

Mazo, M., and Tabuada, P., "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE T. Autom. Contr.*, **56**, No. 10, 2456–2461 (2011), https://doi.org/10.1109/TAC.2011.2164036.

Mueggler, E., Huber, B., and Scaramuzza, D., "Event-based, 6-DOF pose tracking for high-speed maneuvers," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2014), pp. 2761–2768, https://doi.org/10.1109/iros.2014.6942940.

Mueggler, E., Rebecq, H., Gallego, G., et al., "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *Int. J. Robot. Res.*, **36**, No. 2, 142–149 (2017), https://doi.org/10.1177/0278364917691115.

Nawaz, F., Janjua, N. K., and Hussain, O. K., "PERCEPTUS: Predictive complex event processing and reasoning for IoT-enabled supply chain," *Knowl.-Based Syst.*, **180**, 133–146 (2019), https://doi.org/10.1016/j.knosys.2019.05.024.

Raimond, Y. and Abdallah, S., *The Event Ontology*. http://motools.sourceforge.net/event/event.html, acc. Oct 20, 2021.

Ramesh, B., Zhang, S., Lee, Z. W., et al., "Long-term object tracking with a moving event camera," in: *Proc. 29th British Machine Vision Conference* (2018), p. 241.

Rebecq, H., Horstschaefer, T., Gallego, G., and Scaramuzza, D., "EVO: A Geometric approach to event-based 6-DOF parallel tracking and mapping in real time," *IEEE Robot. Automat. Lett.*, **2**, No. 2, 593–600 (2017), https://doi.org/10.1109/lra.2016.2645143.

Rinne, M., Blomqvist, E., Keskisärkkä, R., and Nuutila, E., "Event processing in RDF," in: *Proc. 4th Int. Conf. on Ontology and Semantic Web Patterns* (2013), Vol. 118, pp. 52–64.

Savel'ev, A. I., Khar'kov, I. Yu., Pavlyuk, N. A., and Karpov, A. A., *Mobile Autonomous Robotic Platform with Modular Structure*, RF Patent No. RU 2704048 C1 (2019).

Shaukat, A., Shah, K., Ullah, I., et al., "SmartOntoSensor: Ontology for Semantic Interpretation of Smartphone Sensors Data for Context-Aware Applications," *J. Sensors*, 1–26 (2017), https://doi.org/10.1155/2017/8790198.

Sokolova, A. and Konushin, A., "Human identification by gait from event-based camera," *16th Int. Conf. on Machine Vision Applications (MVA)* (2019), pp. 1–6, https://doi.org/10.23919/MVA.2019.8758019.

Somov, A., Baranov, A., Savkin, A., et al., "Development of wireless sensor network for combustible gas monitoring," *Sens. Actuat. A: Physical*, **171**, No. 2, 398–405 (2011), https://doi.org/10.1016/j.sna.2011.07.016, EDN: PEDPJX.

Taylor, K. and Leidinger, L., "Ontology-driven complex event processing in heterogeneous sensor networks," in: *Extended Semantic Web Conference* (2011), pp. 285–299, https://doi.org/10.1007/978-3-642-21064-8_20.

Thierry, M., *Ontology for Media Resources* (2012), https://www.w3.org/ns/ma-ont#, acc. Oct 20, 2021.

Vatamanyuk, I. V. and Savel'ev, A. I., "A mobile robotic platform as a component of the cyberphysical intellectual space," *Ekstrem. Robotekhn.*, **1**, No. 1, 37–42 (2017).