sustainability

Article

ASPIE: A Framework for Active Sensing and Processing of Complex Events in the Internet of Manufacturing Things

Shaobo Li [1,2], Weixing Chen [1], Jie Hu [1,3] and Jianjun Hu [2,3,*]

[1] Key Laboratory of Advanced Manufacturing Technology of Ministry of Education, Guizhou University, Guiyang 550025, China; lishaobo@gzu.edu.cn (S.L.); weix_chen@yeah.net (W.C.); jason.houu@gmail.com (J.H.)
[2] School of Mechanical Engineering, Guizhou University, Guiyang 550025, China
[3] Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208, USA
* Correspondence: jianjunh@cse.sc.edu

Received: 2 February 2018; Accepted: 1 March 2018; Published: 4 March 2018

Abstract: Rapid perception and processing of critical monitoring events are essential to ensure healthy operation of Internet of Manufacturing Things (IoMT)-based manufacturing processes. In this paper, we proposed a framework (active sensing and processing architecture (ASPIE)) for active sensing and processing of critical events in IoMT-based manufacturing based on the characteristics of IoMT architecture as well as its perception model. A relation model of complex events in manufacturing processes, together with related operators and unified XML-based semantic definitions, are developed to effectively process the complex event big data. A template based processing method for complex events is further introduced to conduct complex event matching using the Apriori frequent item mining algorithm. To evaluate the proposed models and methods, we developed a software platform based on ASPIE for a local chili sauce manufacturing company, which demonstrated the feasibility and effectiveness of the proposed methods for active perception and processing of complex events in IoMT-based manufacturing.

Keywords: Internet of Manufacturing Things (IoMT); critical event; active perception; complex event processing; big data; real-time event

1. Introduction

With the interpenetration of information technology and manufacturing, the integration of manufacturing physical systems and information systems has been increasingly enhanced [1]. The IoMT technology with embedded systems, RFID, and wireless sensor networks at its core, has come into shape as a ubiquitously-perceptual, heterogeneously-integrated, self-adaptive, interoperable, and intelligent manufacturing and information service model. The application of Internet of Things (IoT) technology in the manufacturing industry provides the technical basis for interconnection, real-time position tracking, intelligent control, and so on in the manufacturing process [2]. It also promotes the fusion of manufacturing physical systems and information systems, and can achieve dynamic perception and optimized processing of resources and information production. IoT has been utilized in decision-making in manufacturing to contribute to knowledge-based intelligent manufacturing [3,4].

With the development of more intelligent manufacturing processes, an increasing number of studies have been conducted to explore techniques for manufacturing information perception. Since 2003, the emergence of RFID and other IoT technologies have brought an intelligent revolution in manufacturing [5], which has helped to achieve automated data collection [6], assurance of data dependencies, improvements in production and inventory visibility [7], etc. Kenneth et al. [8] proposed a Radio Frequency Identification (RFID) -based risk management system (RRMS) with rule-based reasoning and RFID integration technology to more effectively monitor real-time operations on site. Barenji et al. [9] proposed a multi-agent RFID-enabled distributed control and monitoring system for a flexible manufacturing shop to enhance the agility, flexibility, and reconfigurability of its manufacturing system. Brintup [6] proposed to use RFID technology to achieve leaner manufacturing through automated data collection and improvements in better monitoring of production and inventory. Liukkonen [10] showed that most application targets of RFID are related to warehouse management, process management, tool management, supply chain management, and life cycle management. Wireless sensor networks [11] have been used for the perception of mobile units to achieve anticipatory control for a flexible manufacturing system. Zhang et al. [12] developed a real-time information capturing and integration framework of the Internet of Manufacturing Things based on Internet of Things (IoT) technology. Tao et al. [13] proposed a framework to use IoT for achieving intelligent perception and access of various resource in cloud manufacturing. Jeschke et al. [14] and Tao et al. [15] introduced frameworks for industrial IoTs and cyber-manufacturing systems.

It is widely recognized that comprehensive perception and intelligent processing of multi-source manufacturing information are critical to decision support for intelligent controls in production processes [16]. The decreasing cost of sensors and the need for intelligent manufacturing has led to the adoption of diverse types of sensors in modern factories for quality control, fault diagnosis, etc. However, this has caused significant challenges in big data processing of such enormous amounts of heterogeneous information from such sensors: how to detect different types of events; how to match events to appropriate decision-making rules; and how to handle the real-time requirement of complex events processing.

Among the datasets produced in the process of interconnected manufacturing, the manufacturing events that guide decision support include primitive events and critical events [17]. Primitive events occur when the perception components of equipment collect raw data. Critical events are complex decision-making events that are generated by association rules from primitive events or complex events [18]. The manufacturing events of a manufacturing process can be complicated due to the distributed deployment structure of the manufacturing system. A complex event processing engine takes the massive heterogeneous data as the observation and extraction objects, obtains meaningful event information from regular operations, and makes up the complex events so that meaningful events can be extracted and summarized in real-time [19]. These concurrent event processes have been discussed by Tanase et al. [20].

The main focus of this paper is how to deal with the complex relationships among multi-level manufacturing events to achieve fast and intelligent complex event processing. Several pioneering studies have been done in this area. Vaidehi et al. [21] designed an ECA language for wireless sensor networks to generate data and events. A single ECA was used to describe simple events and ECA combinations were used to represent complex events. Gerald et al. [22] developed the Cordis event processing model by adopting complex event processing in the event-related research of distributed systems. They also suggested an algebra to describe the association between event information and an algorithm to detect the independent event information in the association description. Romdhane et al. [23] presented a unified processing framework for uncertain information based on probability theory in the process of complex event recognition and created a new composite event model description language to overcome the "complex temporal constraints". Wang et al. [24,25] developed a distributed complex event processing architecture based on the Common Object Request Broker architecture. They also designed an algorithm of plan-based distributed CEP (PDCEP) and a high-performance distributed complex event processing method based on probability, extending the basic finite automata to support the probability of event flow processing models. Yao et al. [26] defined RFID event modeling and complex event operators and presented an architecture for complex event processing for real-time processing of RFID events and the integrity of logical semantic description.

Hu et al. [16] showed that event construction, state prediction, and disturbance detection suitable for big-data environments of modern complex manufacturing systems are critical for RFID-enabled intelligent manufacturing. It is also noted that the greatest challenge is to design efficient and intelligent schemes for manufacturing event processing. Allen et al. [27] proposed an anomaly detection approach for event-based systems that consist of processes that interact through shared resources. A resource-based Petri net formalism is introduced to model these types of systems. Their model generation uses an algorithm based on workflow mining to generate resource-based models. Dias [28] studied how to use timed event graphs to simulate synchronizing operations on productive systems. Complex event processing in IoMTs can also exploit the emerging big data technologies and platforms, such as Apache Spark [29], real-time stream processing platforms [30,31], etc., [32–34].

The complex events in IoT-based manufacturing processes can be regarded as flows of information from multiple sources, for which Cugola and Margara [35] conducted a comprehensive review considering issues from data streams to complex event processing. Fang et al. [36] introduced an information processing mechanism based on a critical event model to organize real-time field data in various abstract levels for enterprise decisions. Their case study at an air conditioner manufacturing company showed its effectiveness. Fault diagnosis of discrete event systems has also been analyzed by Zaytoon et al. [37]. Event-based models have also been used for modeling distributed sensor networks in battery manufacturing [38]. Due to the ubiquity of RFID-based manufacturing plants, new computing methods are needed for rapid processing of the big event data from hundreds or thousands of sensors. Dagkaris et al. [39] proposed and shared ManPy, an open-source software tool for building discrete event simulation models of manufacturing systems. Nie et al. [40] adopted an event handling engine (PUCEP) to improve uncertain complex event matching efficiency by combining the data stream processing engine (SASE) with data generation management and the probability flow theory. The purpose is to link the Internet of Things to data collection and improve the accuracy and reliability of complex events matching. Li et al. [41] suggested a spatiotemporal event model for the complex event processing of farmland IOT, and used the complex event relation operator to judge the spatial-temporal relationship between sub-event and complex event. They also designed and implemented a real-time complex event processing engine, to determine complex events through real-time matching of atomic events and complex event patterns. However, current complex event processing for manufacturing processes lacks a standard model and mechanisms for semantic expression of complex association relations among events. It becomes an urgent need to improve complex event matching processing with strong association rules.

To achieve active sensing and processing of key events in manufacturing processes, this paper proposes an active sensing and processing architecture (ASPIE) for processing manufacturing process events. First, an active sensing method is developed for obtaining key events in manufacturing processes. To address the issue of event expression and processing, an XML-based extensible and easy-to-parse event description language with related event operators is designed to realize the unified description of complex semantic relations. We also proposed a sophisticated event matching scheme to conduct complex event processing based on correlation templates. To evaluate our active sensing and processing architecture and methods, we implemented an ASPIE module embedded in the manufacturing resources management system at a large factory famous for chili sauce production, aiming to support intelligent decisions and optimization of their manufacturing system. The framework and methods for handling the perception and processing of key events in manufacturing processes may provide theoretical and practical guidance for developing enterprise production management decision-making systems.

The remaining paper is organized as follows: In Section 2, an architecture for active sensing and processing of key events in IoMT is introduced. In Section 3, we proposed the methodology for active sensing of key events in manufacturing processes including system design, data processing and packaging, complex events association mining and matching, and transmission and access of the data obtained via active sensing. Section 4 describes how to describe and process the manufacturing events in a standardized way using an improved XML language and template matching. In Section 5, we described our implementation and evaluation of the proposed architecture, models and methods for active sensing and processing of key events at a chili oil sauce manufacturing company. In Section 6, the concluding remarks are presented, and the contributions of this paper and directions for future research are discussed.

2. Architecture of Active Sensing and Processing of Key Events of IoMT

2.1. Architecture of the Internet of Manufacturing Things

The Internet of Manufacturing Things is an open network system that integrates advanced manufacturing, Internet of Things, information, and modern management. It serves all stages of the product manufacturing and service process and the whole manufacturing cycle. Based on the characteristics of manufacturing technology and the expectation of the objectives, we proposed a framework of hierarchical manufacturing object networking with frame supporting function (Figure 1).

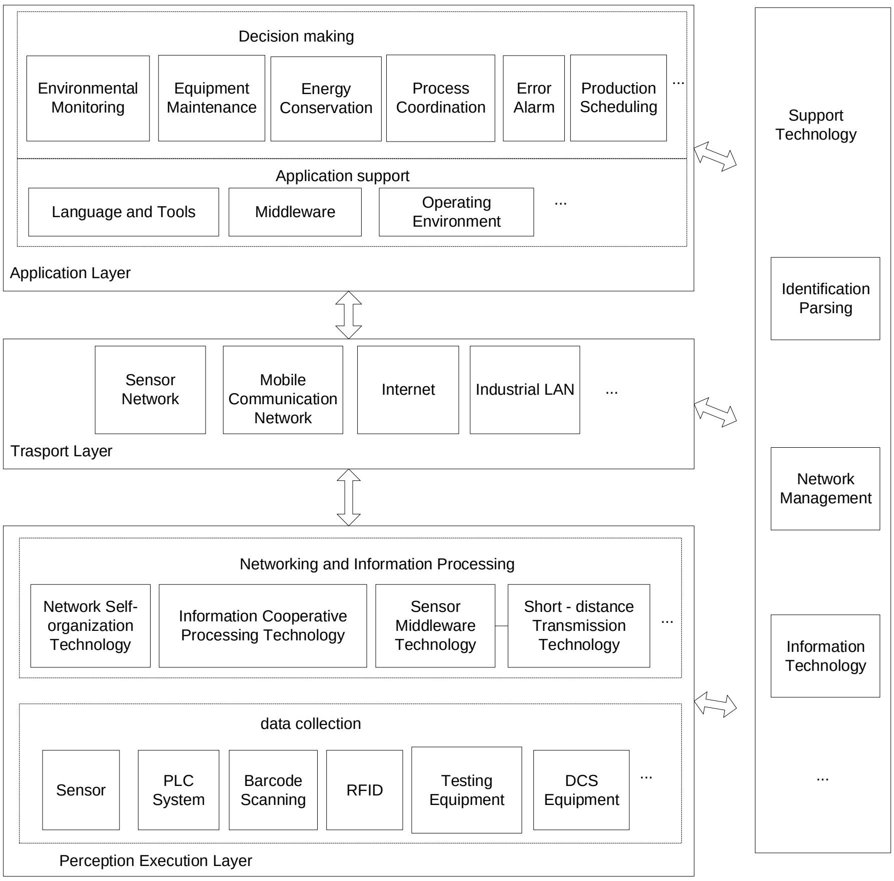

Figure 1. Manufacturing networking architecture composed of perception layer, transport layer, and application layer.

The whole architecture includes the perception execution layer, the transport layer, and the application layer according to the application pattern and development trend of the Internet of Things in manufacturing. The sensing layer executes the perception of the manufacturing process resources, environmental information, and decisions after receiving feedback. The transport layer mainly implements data access and network transmission, provides a real-time transmission service for the production scene examination data, and issues the management level monitoring instructions. The data transmission layer defines the network identification and communication protocol so that the heterogeneous network data can be recognized and merged with each other to gain access to the heterogeneous network. The application layer provides support for decision-making application services with the information processing tools.

2.2. Active Process Perception Model for Manufacturing Processes

Active sensing of IoMT events in the manufacturing processes involves five stages: (1) multi-source heterogeneous manufacturing data acquisition and transmission; (2) data isomorphism, standardized processing, a unified description of the production process events; (3) multi-level event association modeling, and multi-level event relationship analysis; (4) multi-level event correlation and operation of the match operations, obtaining the perceived results of key events; and (5) push and access of perceived results [42].

An active sensing model of the manufacturing process event is designed as shown in Figure 2, for the active sensing stage of the manufacturing event.

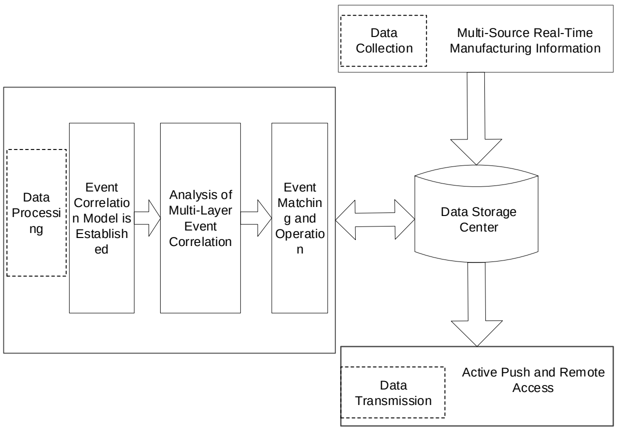
Data Collection
Multi-Source Real-Time Manufacturing Information
Data Processi ng
Event Correlatio n Model is Establish ed
Analysis of Multi-Layer Event Correlation
Event Matchin g and Operatio n
Data Storage Center
Data Transmission
Active Push and Remote Access

Figure 2. The active sensing model of IoMT events. It is composed of data collection, data transmission, and data processing units.

2.3. Architecture of Active Sensing for Complex Events in IoMT

In this paper, we propose the following active sensing architecture for manufacturing events (Figure 3), which includes data acquisition, data transmission, data characterization, data integration, data application, and data service [43].

The sensing layer (Figure 3) is a highly-reliable manufacturing data-aware environment. The main task i : (1) to build a sensing environment for data acquisition and transmission, to establish a sensor network configuration model for measuring different parameters, and to determine the target of the sensor network function and constraints; (2) to establish sensor information registration/information processing library and data description, to describe and define manufacturing resource information with various sensors; and (3) the perception and transmission of real-time data of manufacturing resources, which is manifested when the manufacturing resources reach the sensing area. At that time, the data is automatically collected or identified by the equipments that are aware of manufacturing resource information. Then, the event is perceived by the sensor in real-time. Through the network communication protocol, the sensor device will be able to collect data of the manufacturing resources. The data acquisition layer is used to collect the multi- source manufacturing data and manage the heterogeneous sensor. The data transmission layer is employed to integrate the heterogeneous networks and provide the interoperability to meet the very large volume of real-time sensing data transmission needs. As the heterogeneous network in the manufacturing of things uses different protocols, the data transmission layer ensures that transmission of heterogeneous network data to be real-time and transparent. The data characterization layer is for sensor equipment registration, data definition, and its standardization.

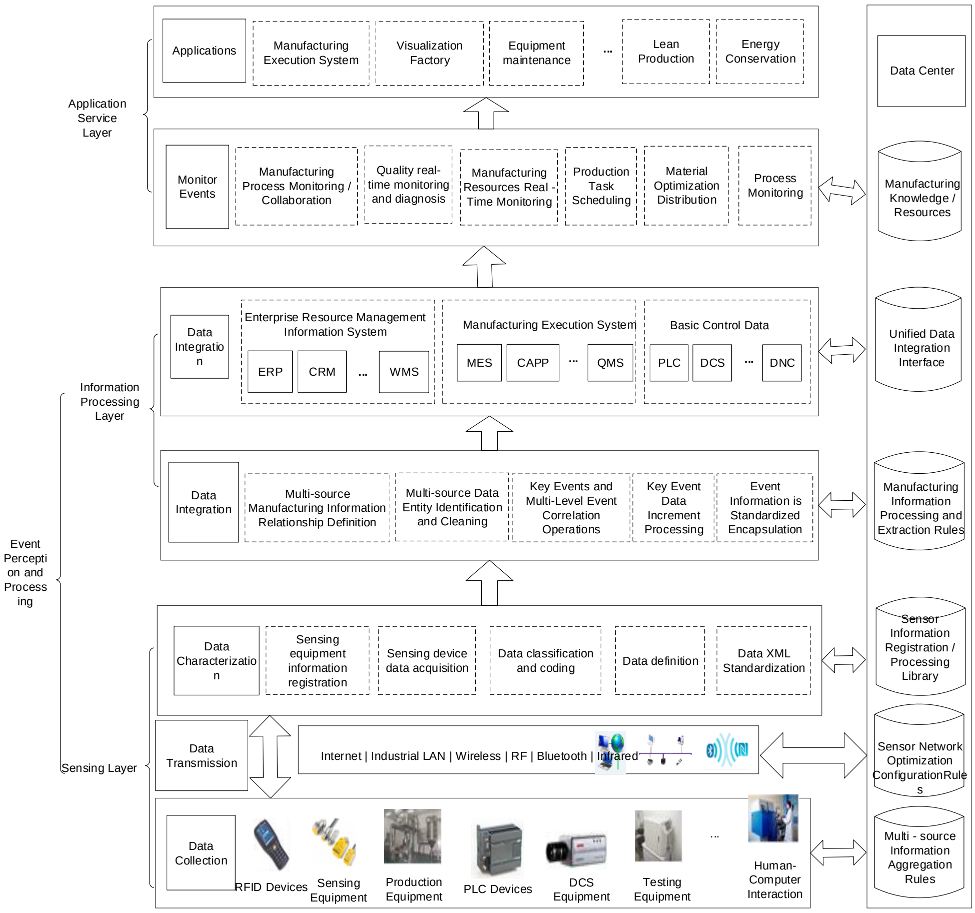
Applications
Manufacturing Execution System
Visualization Factory
Equipment maintenance
...
Lean Production
Energy Conservation
Data Center
Application Service Layer
Monitor Events
Manufacturing Process Monitoring / Collaboration
Quality real-time monitoring and diagnosis
Manufacturing Resources Real - Time Monitoring
Production Task Scheduling
Material Optimization Distribution
Process Monitoring
Manufacturing Knowledge / Resources
Data Integratio n
Enterprise Resource Management Information System
ERP
CRM
...
WMS
Manufacturing Execution System
MES
CAPP
...
QMS
Basic Control Data
PLC
DCS
...
DNC
Unified Data Integration Interface
Information Processing Layer
Data Integration
Multi-source Manufacturing Information Relationship Definition
Multi-source Data Entity Identification and Cleaning
Key Events and Multi-Level Event Correlation Operations
Key Event Data Increment Processing
Event Information is Standardized Encapsulation
Manufacturing Information Processing and Extraction Rules
Event Percepti on and Process ing
Data Characterizatio n
Sensing equipment information registration
Sensing device data acquisition
Data classification and coding
Data definition
Data XML Standardization
Sensor Information Registration / Processing Library
Data Transmission
Internet | Industrial LAN | Wireless | RF | Bluetooth | Infrared
Sensor Network Optimization ConfigurationRule s
Sensing Layer
Data Collection
RFID Devices
Sensing Equipment
Production Equipment
PLC Devices
DCS Equipment
Testing Equipment
...
Human-Computer Interaction
Multi - source Information Aggregation Rules

Figure 3. Architecture of active sensing and processing of IoMT events. It is composed of three layers: the sensing layer, the information processing layer, and the application service layer.

The information processing layer (Figure 3) is employed to receive the data sent by the object perception layer. According to the established data integration rules, the perception data is processed in real time. It establishes data mapping and event handling mechanism (such as standard output data model formulation, event correlation operation and value-added processing rules). It also implements data processing and information integration, and obtains key events information of monitoring and decision-making in the production process. The data integration layer is used for relationship definition, correlation operation, and value-added processing in the production process of multi-source information. The data integration layer performs the collection of production-aware events and the classification of storage to support efficient data transfer.

In order to meet the requirements of real-time monitoring in the manufacturing process, production plan execution, equipment condition detection, material optimization and distribution, and product quality inspection, the application service layer calls or accesses the perceptual information stored in different systems of manufacturing enterprises, and is used to optimize production decisions. The data center provides the service information needed for the manufacturing process management system, such as sensor network optimization configuration rules, multi-source information aggregation rules, sensor information registration database, manufacturing information processing and extraction rules, and enterprise manufacturing resource knowledge bases.

3. Active Sensing Method for Key Events of Manufacturing Processes Based on IoMT

In order to realize active sensing and standard output of the shop manufacturing process, the multi-hop heterogeneous sensor network can be adopted in the manufacturing workshop to manage the heterogeneous sensors and to collect the dynamic data. In data processing, firstly, the sensor information is defined according to the sensor type and data description template, and the heterogeneous data is normalized by the ISA95 standard encapsulation data [44]. Secondly, based on the event business logic and rule based algorithms, key events are obtained by matching the perceptual data based on the XML templates. Finally, the perceived key events are transmitted through Web service technology.

The implementation steps of active sensing of manufacturing process events based on the IoT technology are as follows:

(1) The design of the perception system;
(2) Data processing and standardized packaging;
(3) Complex event correlation matching; and
(4) Transmission and access of perception results.

The process of active sensing and processing manufacturing complex events is shown in Figure 4. The details of the processes are explained as below in subsections.

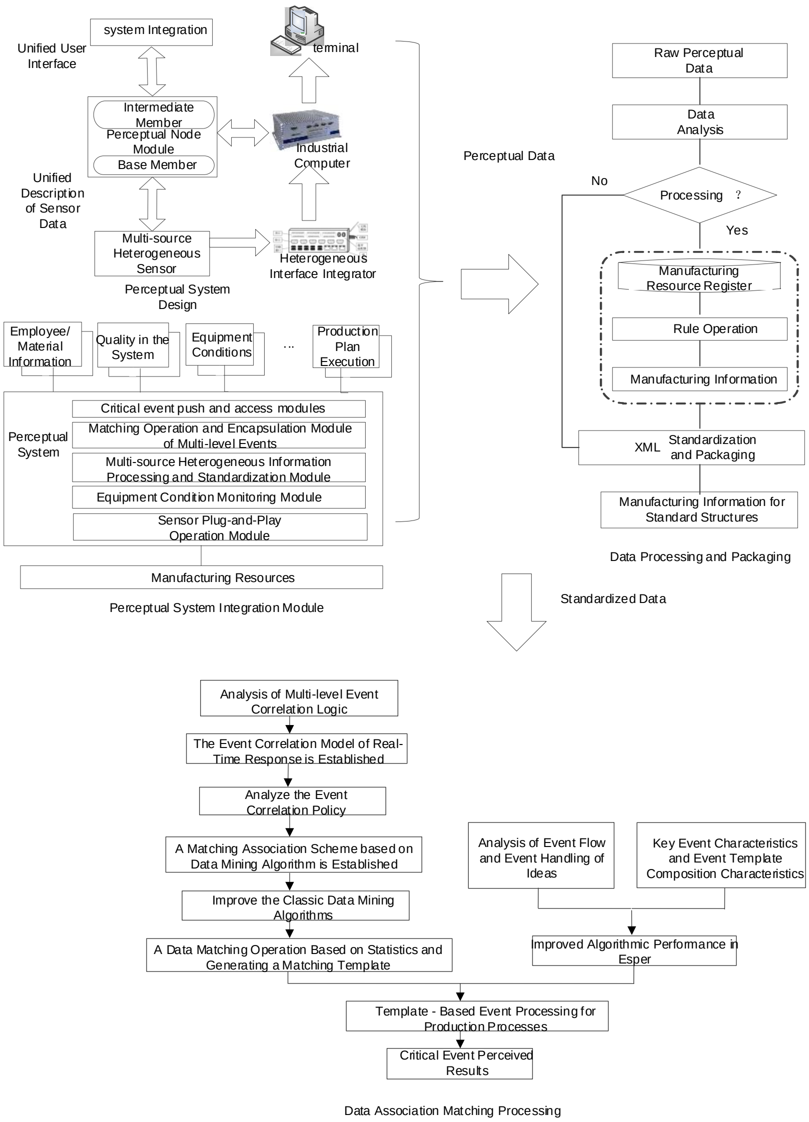
system Integration
Unified User Interface
terminal
Intermediate Member
Perceptual Node Module
Base Member
Industrial Computer
Unified Description of Sensor Data
Multi-source Heterogeneous Sensor
Heterogeneous Interface Integrator
Perceptual System Design
Employee/ Material Information
Quality in the System
Equipment Conditions
...
Production Plan Execution
Perceptual System
Critical event push and access modules
Matching Operation and Encapsulation Module of Multi-level Events
Multi-source Heterogeneous Information Processing and Standardization Module
Equipment Condition Monitoring Module
Sensor Plug-and-Play Operation Module
Manufacturing Resources
Perceptual System Integration Module
Perceptual Data
Raw Perceptual Data
Data Analysis
No
Processing ?
Yes
Manufacturing Resource Register
Rule Operation
Manufacturing Information
XML Standardization and Packaging
Manufacturing Information for Standard Structures
Data Processing and Packaging
Standardized Data
Analysis of Multi-level Event Correlation Logic
The Event Correlation Model of Real-Time Response is Established
Analyze the Event Correlation Policy
A Matching Association Scheme based on Data Mining Algorithm is Established
Improve the Classic Data Mining Algorithms
A Data Matching Operation Based on Statistics and Generating a Matching Template
Analysis of Event Flow and Event Handling of Ideas
Key Event Characteristics and Event Template Composition Characteristics
Improved Algorithmic Performance in Esper
Template - Based Event Processing for Production Processes
Critical Event Perceived Results
Data Association Matching Processing

Figure 4. Architecture of active sensing and processing of IoMT events.

3.1. Perception System Design

Due to the sensor heterogeneity, an embedded manufacturing information sensing device is designed to conduct multi-source heterogeneous data acquisition, network I/O control, rule based algorithm processing, and data storage and transmission. It also realizes heterogeneous sensor concentration management and heterogeneous information-aware processing. In the perception system, real-time manufacturing information can be acquired with the underlying components of the sensing node module. The intermediate component integrates the data collected by the bottom node, real-time transmission, and decision briefing, and generates events through the logical operation of data information. A unified user interface facilitates the development of the system and the efficient transfer of information. A unified description of the sensor information achieves the standardization of information processing and expression. For the sake of generality, real-time sensing data per sensor is encapsulated as a standard Web service so that real-time data can be transmitted over the network.

Perception system integration of functional modules mainly includes plug-and-play of heterogeneous sensors, condition monitoring equipment, multi-source heterogeneous information processing and standardization, multi-level event matching operations and packaging, and event push and access.

3.2. Processing and Standardized Package of Perception Data

According to the data structure of storage, the real-time manufacturing information is divided into dynamic data and static data. Dynamic data is stored as XML documents, of which the data structure is semi-structured and unstructured. Static data (basic data) is the property data of manufacturing resources, also referred to as the basic information of the object, which is stored in a relational database. In order to achieve the unified expression and efficient call of heterogeneous data, the relational data and XML data can be exchanged, and the perception data is encapsulated and stored as XML documents.

3.3. Correlation Matching of Complex Events

First of all, we need to integrate the manufacturing information that's processed and packaged in a standard package (including multi-source manufacturing information related to definition of association, template matching of multi-level events, regular operation, and increment processing of events). The event association matching process consists of three steps: (1) to establish an event association model using XML language to describe the association of multi-source manufacturing information, and to establish an XML file describing the event structure; (2) to mine association rules of events based on association data mining algorithms and to establish the correlation matching templates of complex events; and (3) to improve the performance of the algorithm in the event stream processing engine and perform the calculation of the complicated events in the production process based on template matching.

3.4. The Transmission and Access of Perception Results

Perceived complex events are mainly based on Web service technology for active push and remote access, as shown in Figures 5 and 6.

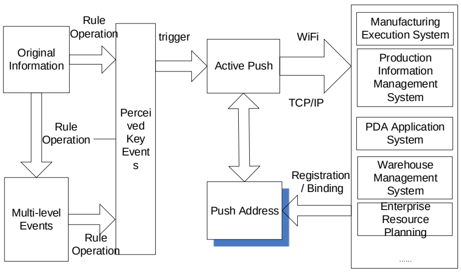

Figure 5. Active push service of key events.

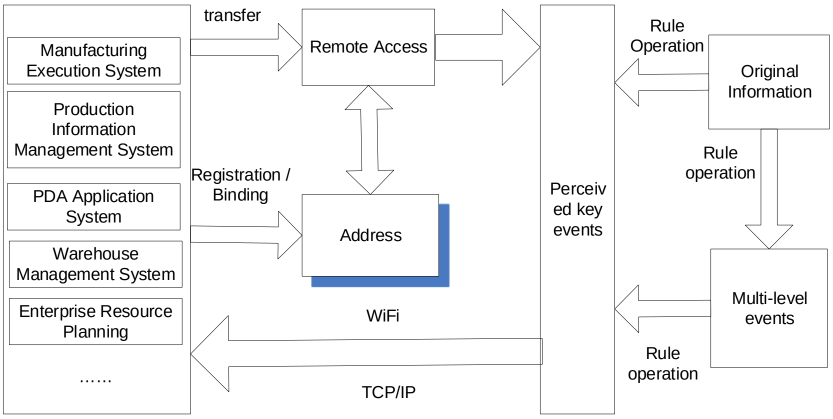

Figure 6. Remote access service of key events.

4. Standardized Descriptions and Processing Of Manufacturing Process Events

With the improved XML language, the standardized processing of manufacturing events is achieved in order to realize the unified semantic expression of the complex relations of multi-level events in the manufacturing process. At the same time, we developed a complicated event association matching scheme to process complex events based on templates.

4.1. Correlation Model of Manufacturing Association Events

In general, the relationship between the manufacturing process events are: (1) temporal relationships or time-sequence relationships, which can be described using time models; (2) logical hierarchies, which can be described using hierarchical models; and (3) causal relationships, described using semantic operators for expression. Assuming CE represents a key event or complex event. PE represents an original event. The expression for the key event can be written as: CE = f(PE1, PE2, ⋯, CE1, CE2, ⋯). The logical operator f defines the logical relationship between events and can be nested in multiple layers.

Based on the production process event type and correlation analysis, a complex event structure model oriented to the production process is established (see Figure 7). Relational operators are used to describe the relationship between the multi-level events, such as timing and logic hierarchy, to reveal the information of sub-events and their implications in multi-level complex events, and to explain the selection and matching of sub-event instances of complex events. The temporal and semantic relations between multi-level events can also be identified.

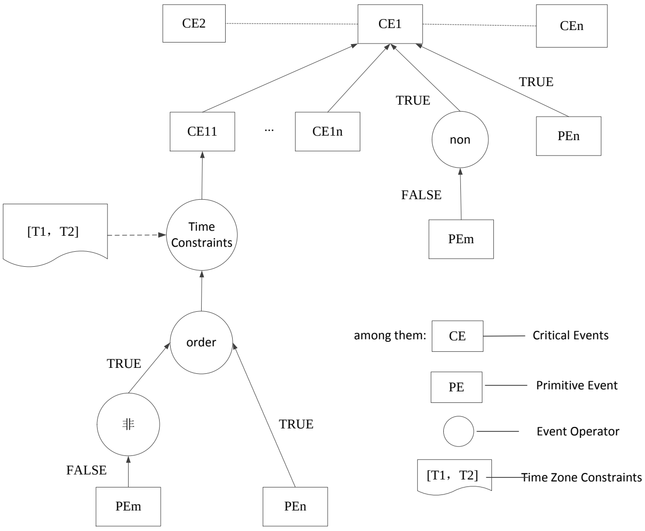

Figure 7. Structure model of complex events in IoMT.

There exists an association of similar event type aggregations between event CE1 and event CEn. For event PE, the CE1n upper level event is CE1, and there is a generalized relationship between event CE1, events CE1n and PE. The aggregation process for the critical event CE11 [45] is: (1) if the original event PEn occurs, the event value is TRUE; (2) if the original event PEm does not occur, the event value is FALSE, and the non-operator aggregation event value becomes TRUE; (3) after the two original events are processed, the event value becomes TRUE by the sequence operator; and (4) after the time constraint operator is aggregated, the event value becomes TRUE, i.e., the event CE11 occurred.

4.2. Description of XEDL-based Manufacturing Events

(1) Basic Syntax Structure

Our basic syntax definition of XML-based event description language (XEDL) includes three aspects: (1) XEDL text should conform to the XML specification, the starting line should be: <?xml version = "1.0" encoding = "utf-8"?>; (2) With XEDL as root: <XEDL> . . . /XEDL>; (3) Under the root, there are three parts: event structure, event strategy and event attributes.

The XEDL syntax consists of a collection of raw event registries and key event sachems. The syntax is described as follows:

```
<XEDL>
<PEset>//Primitive event registry
<PE> . . . </PE>//Primitive event description
</PEset>
<CEset>//A collection of key event patterns
<CE> . . . </CE>//Key event patterns
</CEset>
</XEDL>
```

(2) Description Syntax of Original Events

The primitive event triggered by the sensor node is represented by a unique identifier ID and it represents a change in the same class or similar state. The original event body contains the source sensing device, the time, the location of occurrence, and other attributes.

Description syntax of the primitive event is as follows:

```
<PE>
<ID>primitive event_i</ID>//The original event i uniquely identifies the ID
<sources>//The original event occurs from the source sensor node
<source>sensor_n</source>//Sensing source n
</sources>
<other_attributes>//Other attribute definitions
<attribute>otherattr_m</attribute>//The attribute m
</other_attributes>
</PE>
```

(3) Description Syntax of Key Events

A key event is described as a complex event pattern, which includes three aspects: the cause event set, the instance strategy, and the logic operation function. The key event expression consists of the event operator and the operand, denoted as CE expression (operator, operand). The operands represent the event type, and the operators combine the event types. The definition of event operator [46,47] consists of three parts: (1) the number of instances of the operands; (2) the usage strategy and consumption strategy of the event instance; (3) the usage strategy of event instances consists of four operators: primitive, recent, continuous, and cumulative; and the consumption strategy of the event instance consists of four operators: reserve, remove-early, remove-current, remove-all. Qualified events are reserved following the usage strategy or consumption strategy of the operand processing or after deleting the ineligible instances when the number of event instances of the operand is exceeded. The usage strategy of an event instance specifies how an event instance can be used for event aggregation. The consumption policy of the event instance determines whether the complex event is saved or deleted after it is involved in complex event calculations.

Combined with the complex event operators and XML syntax, the key event XEDL syntax is defined as follows:

```
<CE>
<ID>complex_event_n</ID>// The unique ID of the key event n
<attributes>●●●</attributes>// Definition of event attributes
<eventset>// An set of cause event instances
<event>●●●</event>// Acause event instance and strategies
</eventset>
<logicfunction="judgefunction | constructorfunction">// The logical operation function of the event
<judgefunction>conjunction | disjunction</judgefunction>//Judgmentfunction
<constructorfunction>●●●</constructorfunction>// calculation function of attributes
</logicfunction>
</CE>
```

(1) The Policy Description of the Causal Event Instance

The use of the event instance, the consumption policy, and the cause event are described in XEDL syntax as follows:

```
<event>
<use>// Use policy definitions
primitive | recent | continuous | cumulative// Use the policy options
</use>
<consume>// Definition of consumption policy←←
reserve | removecurren | removeearly | removeall// Consumption policy options
</consume>
</event>
```

(2) Syntax Description of the Logic Operation Function

```
<logicfunction>// Event logic operation function
<judgefunction>// Judgment function
<attributes>position | source | time | other</attributes>// The type of the event attribute
●●●
</judgefunction>
<constructorfunction>// The calculation function for the attribute
<attributes>●●●</attributes>// The event property name is evaluated
<variables>●●●</variables>// The calculation of the variables
●●●
</constructorfunction>
</logicfunction>
```

4.3. Manufacturing Process Event Processing Based on Complex Event Processing (CEP) and Associated Templates

The manufacturing process event is processed by matching relation scheme based on template sequences. Firstly, the Apriori algorithm is used to mine the event data set and generate the matching template, which is stored in the memory database in XML format. Then, the template-based complex event processing operation is performed, and the key events of the manufacturing process are obtained [17].

(1) Manufacturing Process Event Matching Association Scheme

The Apriori algorithm is used to mine association rules and generates matching templates. The algorithm rules are: (1) if there is a causal relationship between active-perceived events and sub-events, the confidence level of the association rules is 1; (2) there are one-to-one, many-to-one association rules between perceived complex events and the causal events. There is no need to consider one-to-many or many-to-many cases; and (3) for the one-to-one association rules of events, there is a sufficient non-essential conditional relation between the source event and the target event.

The algorithm of matching template [48] is generated as follows:

```
begin
   U ← (History-aware event patterns from the database);
for all Ei∈U do
       Di.id = newid;//Complex Event Matching Pattern Detection
for all Di∈U do
  if (Di = Φ)
      Delete (Di);
         if (Di - 1! = Φ)
         Di-2 = Di-2∪Di-1;//The last level of event handling
Al ← {i∈Di, and i satisfies the term of the support degree S }; //A 1-level association rules with minimum support
for k = [2,3, . . . ] do
 Tk = Ak -1 × Al; //Multilayer transaction matching
 Ak {i∈Tk and i satisfies the term of the support degree S}; //A k-level association rule with minimum support
   if (Ak = Φ)
        break;
 E ← (The event group in Ak (k! = 1) conf(e) = 1);
 Association template ← {the event group e∈E, conf(e) = 1};
 end
```

(2) Template-Based Matching of Production Process Events

To achieve the complex processing and analysis of multiple atomic events in real-time, we chose template matching as the complex event processing data stream processing scheme. The template is stored in the knowledge base of “in-memory database”, and the production process events are handled by template matching and complex event processing. The main flow is shown in Figure 8.

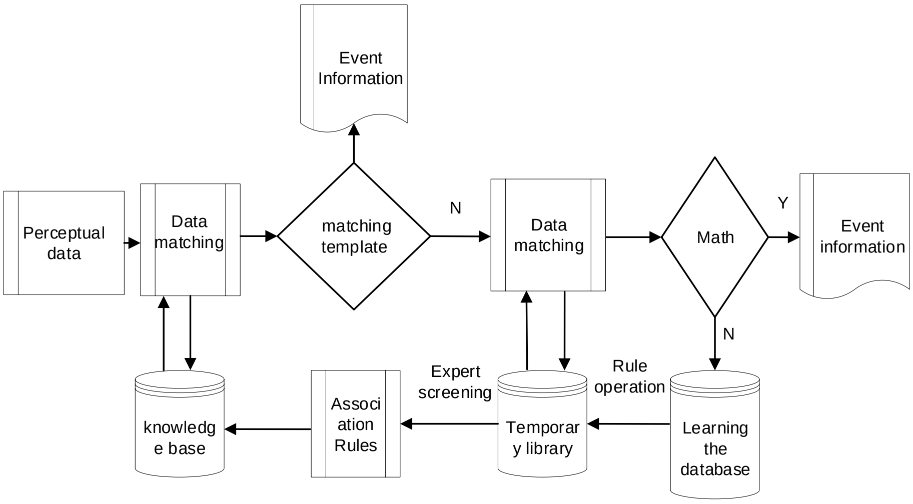

Figure 8. Template-based complex event processing.

In complex event processing, the NFA is used to describe the push decision of an atomic event. Depending on the business needs, the application of nondeterministic finite automata in complex event processing is extended and innovated as follows:

(1) When the life cycle of the automatic validation machine is overdue, the automatic machine that does not complete dies, and the event request was cancelled.

(2) During the concurrent event processing, the automaton calculates and matches the matching system separately, and provides the suspending function for event handling.

(3) For those automaton that “from suspend to die”, atomic events will be deposited into a temporary library for subsequent repair.

The complex event processing algorithm of the finite machine is described as follows:

```
1: e = GetEvent (); // Get the atomic event
2: if (Timeup (e)) do
3:    Cep Start With (e); // If the time of event e exceeds the verification period, a new periodic process is detected
4: if Request wait (e) do
5:    Send (e); // If there is a request for an e event, an e event is sent
6: result = result Cep (e); // for complex event processing
7: if Cep List push (e) do
8:    goto step 1; //determine the completion of the event is not processed and re-verified
9: if result = Cep End do
10:    SendtotempDB (e); //complex event aborted, the event e stored in the memory database
11: if result = CepSuccess
12:    Outputresult (resultCep (e)); //complex event processing operations to terminate and output critical events
13: goto step 1;
```

(3) Key Processes for Manufacturing Processes Based on Matching Templates

(1) Heuristic Real-Time CEP of the Directed Graph

To realize real-time active sensing of complex events by the CEP engine, we have developed a set of algorithms based on association templates to improve the speed of event detection and correlation matching. One of the major contributions of this paper is to develop methods to achieve real-time processing of multi-level event instances by performing complicated event pattern matching according to the set task priorities, thus reducing the processing time and facilitating real-time active sensing of key events.

Due to the relatively fixed mechanism of automata and Petri nets in the CEP intermediate structure, it is difficult to adjust the priority of tasks. We, thus, proposed a complex event structure model according to event expression and the type of the manufacturing process. A directed graphic [49] is chosen to show the mapping model of complex event processing. Specifically, complex event patterns in the library are mapped to a directed graph and stored in the memory database, where there are directed graph nodes, intermediate nodes, and root nodes, which correspond to atomic events, intermediate modes, and complex event modes. Identical nodes are merged, as shown in Figure 9. There exist necessary relationships between event PE (process event) and CE (complex event).

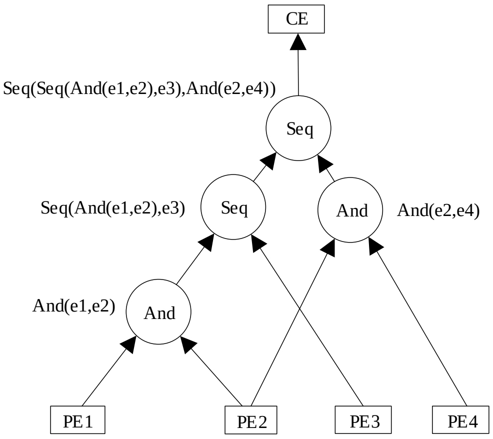

Figure 9. Directed graph based on a complex event pattern.

The mid-point node performs complex event correlation matching according to the CEP patterns and transmits the result to the parent node when the atomic event is generated by the complex event processing procedure of the directed graph. When the corresponding parent node detects the input of the sub-node event, the CEP matching pattern in the corresponding node is searched and the association process is performed. If the match is successful, the intermediate node outputs the processed complex event to the parent node for subsequent association, and the root node outputs the final event-aware event. Otherwise, the corresponding event is stored or discarded depending on the semantic judgment.

(2) Event Matching Based on the Heuristic CEP Algorithm

The essence of our heuristic is to find the shortest path function that defines the path from the sub-node to the parent node in a complex event pattern tree, which is used to search and evaluate the optimal complex event processing scheme. The heuristic function considers two factors: (I) when some or all of the sub-events of the node occur, the pattern matching is carried out; (II) the shorter the path between event processing nodes, the fewer the event composition layers, and the easier it is for complex event pattern matching. Heuristic functions are defined as follows:

$$\mathrm{H(n)} = \mathrm{min}_{\mathrm{r}\in \mathrm{L(n)}}\left(\sum_{\mathrm{n_i}\in \mathrm{r}} \mathrm{W(n_i)}\right) \tag{1}$$

$$\mathrm{W(n_i)} = \mathrm{k_{max}} - [\mathrm{P(n)}/\mathrm{Q(n)} - 1](\mathrm{k_{max}} - \mathrm{k_{min}}) \tag{2}$$

where $H(n)$ is the heuristic function of node n. $W(n)$ denotes all sets of paths from the node n to each exit node, representing that node n_i is a node on the path r from node n to a particular exit node. $W(n)$ is a waiting function of node n, used to ensure the efficiency of synchronous threads and programs. The value of $\mathrm{W(n)}$ is determined according to the value range of $W(n)$, $[k_{max}, k_{min}]$, $P(n)$, and $Q(n)$. $P(n)$ represents the number of sub-events that have occurred for the complex event pattern. $Q(n)$ represents the minimum number of sub-events required to trigger the pattern matching.

Our heuristic based complex event pattern matching algorithm is shown as follows:

```
Imports: subevent flow = [e1, e2, e3, ...] // input atomic event stream
Exports: complex event flow / / output according to complex event pattern matching the key
event flow
Function Heuristic Search (Instance II ext)
    For each subevent flow e {
        hb set to null //initialization engine, the node queue hb is empty
        n← e // Case e The corresponding node
        For the parent node f of n{
            append e into the end of the cache queue at f
            Put f into hb}}
While hb is not null {
n←The smallest node existed in hb
ce← the first event instance in the Buffer queue at node n
/ / event instance with priority to handle
ins ←new event instance or null generated by the ce pattern matching at node n
remove ce from the the cache queue n
Recalculate H(n) and all its children node H($n_i$)
If ins <> null {
If n is an exit node {
        append ins at the end of output queue of complex events
Else
        For the parent node f of n{
            append ins at the end of the buffer queue of
            save f into hb}}}}
If the instance cache queue of n is null {
        remove n from hb}}
End function
```

(3) Real-time CEP Framework and its Working Mode

Our real-time CEP engine was developed under the Eclipse environment using the Java language. The overall framework is shown in Figure 10. Atomic events are formed after the sensed data is cleaned and encapsulated. The atomic events are extracted by atomic event extractor and the atomic events stream is generated. At the same time, atomic events are standardized and encapsulated, and entered into the real-time CEP engine in XML format. The CEP engine receives the atomic event stream and simultaneously detects the complex event pattern and completes the event correlation matching. The complex event stream is exported at last. The domain expert can add or delete the matching template generated by association rules mining online.

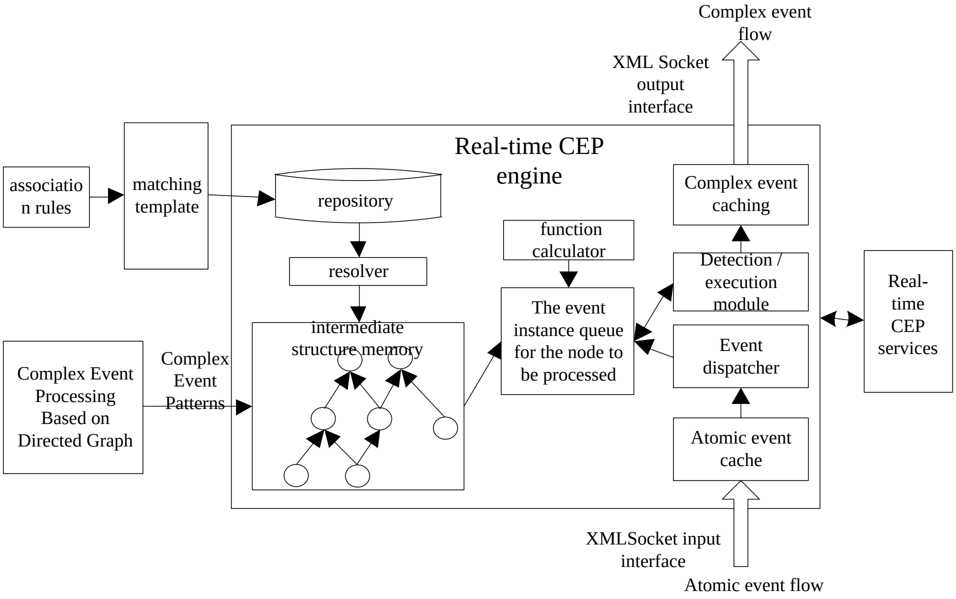

Figure 10. Framework of the real-time CEP engine.

The working principles of the real-time CEP engine associates atomic events with complex event patterns generated by association rules and outputs complex events. The CEP engine first buffers and distributes the atomic events. Then, the fast heuristic CEP algorithm based on the directed graph is implemented by the detection/execution module, and the complex events resulting from the correlation matching are placed in the complex event cache in XML format. Finally, complex events are exported to the information application system to provide decision support for the manufacturing process. The real-time CEP service evaluates the latest response time of the complex event pattern during the detection process and ensures the real-time performance of the event sensing. The real-time CEP engine communicates via two types of interfaces: (I) XML documents are transmitted through the XML Socket; and (II) data are transmitted through Java RMI that invokes the event processing methods in the engine remotely.

In order to deal with complex event pattern processing in the CEP engine, domain experts generate event-matching templates through complex event association rule mining and improving related algorithms, and transform them into XML-based complex event correlation matching description templates, which are saved in the knowledge base. The complex event correlation matching template is mapped by a parser into a complex event pattern, which is identified by a pattern-matching structure based on a directed graph to facilitate the transmission of complex event processing.

According to the framework of the real-time CEP engine, the real-time CEP engine operating mode f is defined as follows: (I). The atomic event stream is entered into the atomic event cache of the CEP engine via the XML Socket interface. (II) According to the event type and the control of the event processing, the atomic event dispatcher is used to load the atomic event into the corresponding node processing unit and cache the queue according to the heuristic function value. The heuristic function calculator is used to recalculate the heuristic function of the node when the node instance cache changes, and, thus, rearrange the node queue. (III) The priority node is extracted in the queue through the detection/execution module and the corresponding node in the node is deleted. Meanwhile, the node processing unit script is calculated. (IV) The heuristic function calculator together with the detection/execution module process the event instances until there are no event in the pending node queue. A, and the priority event in the atomic event cache continues to load the node processing queue through the atomic event dispatcher. (V) Complex events generated by the detection module and the CEP node are appended to the complex event cache and will be exported through the XML Socket interface

5. Deployment of Active Perception and Processing Technology for Manufacturing

5.1. Application Development Platform

To evaluate the effectiveness of the proposed ASPIE framework for active sensing and processing of IoMT events, we developed a software platform for a local chili oil sauce manufacturing company using the technology framework as shown in Figure 11. The object-oriented management platform was implemented using C# and Java with Microsoft SQL Server 2008 R2 database, Windows 7 operating system, and the .NET 4.0 architecture.

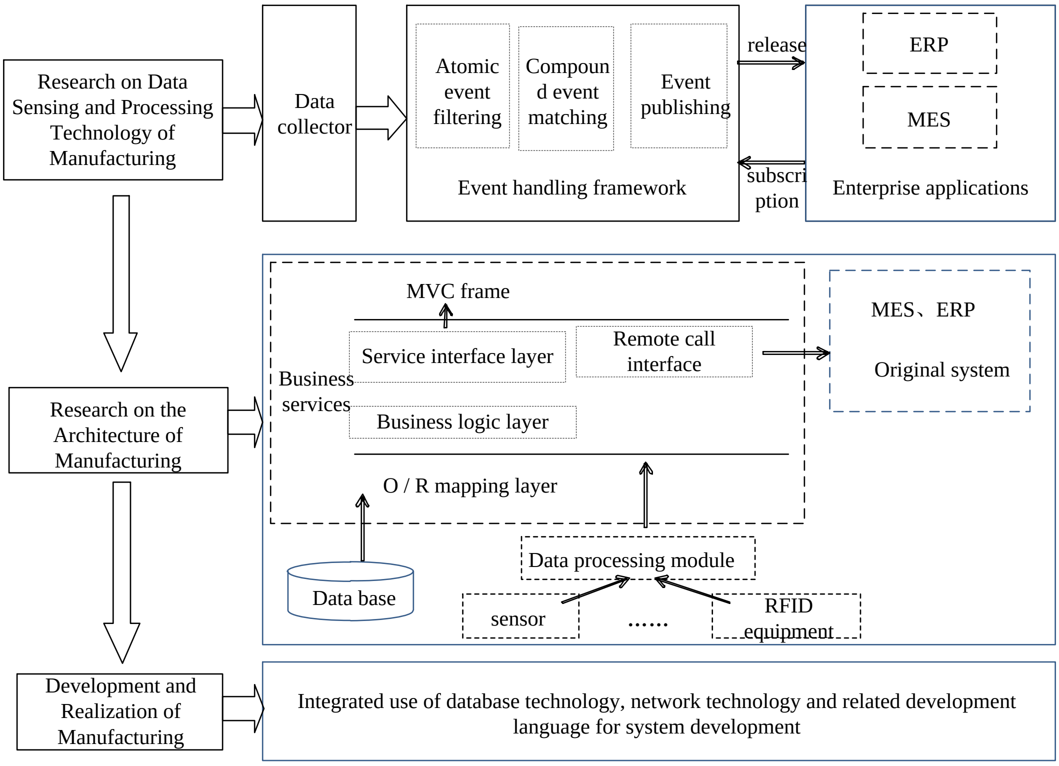

Figure 11. Technology framework for ASPIE implementation.

5.2. System Implementation

The ASPIE framework has to be embedded into the current ERP system at the LGM chili oil manufacturing company to evaluate how they can help to improve the management of the manufacturing processes. LGM has implemented the ERP system aiming to achieve manufacturing resource data perception by the heterogeneous sensing equipment. It can use data resource management and knowledge data mining and analysis of manufacturing data to provide data assets for business needs inside and outside the enterprise. Furthermore, data need to be shared in the cloud to support ubiquitous collaborative manufacturing through the manufacturing resource visualization and manufacturing resources service-on-demand. The whole dataflow and related modules of LGM ERP is shown in Figure 12 with the ASPIE modules included. In this framework, the production process sensing data and their processing is used to provide decision support for dynamic monitoring of discrete manufacturing processes for effective improvement of manufacturing execution and management.

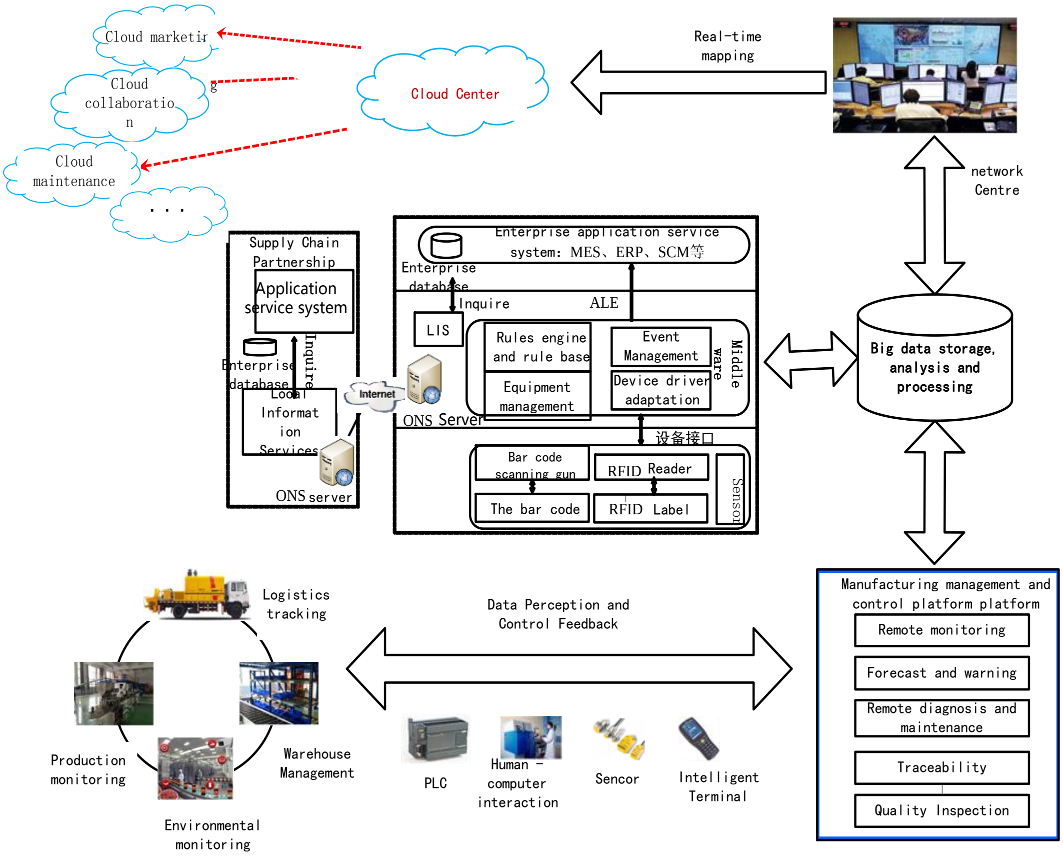

Figure 12. Production data flow and ASPIE module at LGM, a Chili oil manufacturer.

According to the function modules of the manufacturing system platform and processing of data sensing, the applications of comprehensive data are analyzed in Table 1.

Table 1. Event and data flow of LGM manufacturing processes.

Case Name	Production Process IoMT Events
Display content	Real-time video monitoring of production processes, real-time manufacturing environment, product quality inspection data, raw material quality data analysis, equipment working conditions, key business point checks.
Basic event flow	(1) Real-time production video monitoring via video camera: to obtain the real-time status of production line. (2) Real-time monitoring of environmental data (via real-time sensors): to obtain environmental monitoring information, including air temperature, humidity and combustible gas concentration. (3) Product quality data monitoring: to observe the trend graph of qualified product ratios. (4) Raw material quality data analysis: to observe the quality of raw materials and its changes at different time periods. (5) Equipment work status checking: to observe the work status changes of forklifts (we used the forklift status information as example). (6) Key business point check: to check key information at the critical business sites, including raw materials shelf life reminding, equipment status reminding, to-do task reminding, information of the staff on site.

In our LGM chili sauce manufacturing company example, we analyzed the sensing data of personnel, forklifts, an manufacturing environment to provide the production managers real-time environment information, equipment information, and working conditions of forklifts, which provided decision support for effective control of the manufacturing environment of the workshop and resource scheduling.

6. Conclusions

This paper proposed an active sensing and processing architecture (ASPIE) of IoMT complex events based on data-driven manufacturing processes, characteristics of IoMT, and complex event processing technology. ASPIE will be able to support production control and intelligent decision-making of manufacturing enterprises, which is composed of an active sensing method of the complex events in IoMT processes, a unified descriptive language of event models, and a template-oriented processing method for complex events. The proposed ASPIE framework has already been implemented and evaluated at a food manufacturer, which showed its feasibility and effectiveness.

Considering the application requirements of intelligent manufacturing, the research in this paper can be extended in the following aspects: (1) To establish a highly-efficient interactive manufacturing data management system; (2) to study the parallel and efficient data mining algorithm for complex manufacturing events; (3) due to the complex relationship among the events of the IoMT processes, the event instance strategy and logic operation function needs to be further studied in the XEDL event model description language for a more unified and accurate description of the complex relationships between the objects, between the activities, or between the objects and activities contained in the manufacturing data; and (4) to construct big data application planning based on an intelligent manufacturing model

Acknowledgments: This work is supported by the China Scholarship Council and the National Natural Science Foundation of China under grant nos. 91746116 and 51741101, and Science and Technology Foundation of Guizhou Province under grant no. Talents [2015]4011 and [2016]5013, Collaborative Innovation [2015]02. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. We would also like to thank Qin Li and Hongjing Wei for their help in English writing of this paper.

Author Contributions: S.L. and W.C. conceived of and designed the study. W.C, S.L., and Jianjun worked on the algorithm design. W.C., Jianjun, and S.L. wrote the manuscript. Jie made the figures and reformatted the manuscript. Jie and Jianjun revised and polished the manuscript. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jardim-Goncalves, R.; Romero, D.; Grilo, A. Factories of the future: Challenges and leading innovations in intelligent manufacturing. *Int. J. Comput. Integr. Manuf.* **2017**, *30*, 4–14.
2. Lu, Y.J.; Cecil, J. An Internet of Things (IoT)-based collaborative framework for advanced manufacturing. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 1141–1152.
3. Yao, X.; Yu, M.; Chen, Y.; Xiang, Z. Connotation, architecture and key technologies of Internet of manufacturing things. *Comput. Integr. Manuf. Syst.* **2014**, *20*, 1–10.
4. Hou, R.-C.; Ding, X.-Q.; Tao, Y.; Wang, L.-S.; Jing, R.-H.; Li, J.-H. Internet of manufacturing things and relevant technical architecture. *Comput. Integr. Manuf. Syst.* **2014**, *20*, 11–20.
5. Potyrailo, R.A.; Surman, C.; Monk, D.; Morris, W.G.; Wortley, T.; Vincent, M.; Diana, R.; Pizzi, V.; Carter, J.; Gach, G.; et al. RFID sensors as the common sensing platform for single-use biopharmaceutical manufacturing. *Meas. Sci. Technol.* **2011**, *22*, 082001. [CrossRef]
6. Brintrup, A.; Ranasinghe, D.; McFarlane, D. RFID opportunity analysis for leaner manufacturing. *Int. J. Prod. Res.* **2010**, *48*, 2745–2764. [CrossRef]
7. Huang, G.Q.; Zhang, Y.F.; Jiang, P.Y. RFID-based wireless manufacturing for real-time management of job shop WIP inventories. *Int. J. Adv. Manuf. Technol.* **2008**, *36*, 752–764. [CrossRef]
8. Poon, K.T.C.; Choy, K.L.; Lau, H.C.W. A Real-time Manufacturing Risk Management System: An Integrated RFID Approach. In Proceedings of the 2007 Portland International Conference on Management of Engineering & Technology, Portland, OR, USA, 5–9 August 2007.

9. Barenji, R.V.; Barenji, A.V.; Hashemipour, M. A multi-agent RFID-enabled distributed control system for a flexible manufacturing shop. *Int. J. Adv. Manuf. Technol.* **2014**, *71*, 1773–1791. [CrossRef]
10. Liukkonen, M. RFID technology in manufacturing and supply chain. *Int. J. Comput. Integr. Manuf.* **2015**, *28*, 861–880. [CrossRef]
11. Silva-Faundez, J.A.; Duran-Faundez, C.; Melin, P.; Aguilera, C. An Anticipatory Control for a Flexible Manufacturing System Based on the Perception of Mobile Units Using WSNs. *Int. J. Comput. Commun. Control* **2015**, *10*, 702–717. [CrossRef]
12. Zhang, Y.; Zhan, G.; Wang, J.; Sun, S.; Si, S.; Yang, T. Real-time information capturing and integration framework of the internet of manufacturing things. *Int. J. Comput. Integr. Manuf.* **2015**, *28*, 811–822. [CrossRef]
13. Tao, F.; Zuo, Y.; Xu, L.D.; Zhang, L. IoT-Based Intelligent Perception and Access of Manufacturing Resource Toward Cloud Manufacturing. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1547–1557.
14. Jeschke, S.; Brecher, C.; Meisen, T.; Özdemir, D.; Eschert, T. *Industrial Internet of Things and Cyber Manufacturing Systems*; Industrial Internet of Things; Springer: Cham, Switzerland, 2017; pp. 3–19.
15. Tao, F.; Cheng, J.; Qi, Q. IIHub: An industrial internet-of-things hub towards smart manufacturing based on cyber-physical system. *IEEE Trans. Ind. Inform.* **2017**, *99*. [CrossRef]
16. Hu, J.W.; Lewis, F.L.; Gan, O.P.; Phua, G.H.; Aw, L.L. Discrete-Event Shop-Floor Monitoring System in RFID-Enabled Manufacturing. *IEEE Trans. Ind. Electron.* **2014**, *61*, 7083–7091. [CrossRef]
17. Zhao, X.; Zhang, Y.; Sun, S.; Wang, J.; Si, S. Implementing method and key technology for IoT-based manufacturing execution system. *Comput. Integr. Manuf. Syst.* **2012**, *18*, 2634–2642.
18. Nan, L.; Hu, K.; Zhu, Y. A Novel Distributed Complex Event Processing for RFID Application. In Proceedings of the Third International Conference on Convergence and Hybrid Information Technology, Busan, Korea, 11–13 November 2008; pp. 1113–1117.
19. Li, Y. *Correlation Solution of System Monitoring Events Analysis and Administration based on Complex Event Processing*; Donghua University: Shanghai, China, 2012.
20. Tanase, N.; Iosif, G.; Nedelcu, A.; Iordache, I.; Luchian, A.M.; Suciu, G.; Suciu, V.; Costea, E. Concurrent engineering application in microsystem technology. *J. Sci. Arts* **2017**, *3*, 597–610.
21. Vaidehi, V.; Vasuhi, S.; Ganesh, K. Person tracking using Kalman Filter in Wireless Sensor Network. In Proceeding of the 2010 International Conference on Advanced Computering, Chennai, India, 14–16 December 2010; pp. 60–65.
22. Gerald, G.K.; Boris, K.; Kurt, R. Cordies: Expressive event correlation in distributed systems. In Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems, Cambridge, UK, 12–15 July 2010; pp. 26–37.
23. Romdhane, R.; Boulay, B.; Bremond, F.; Thonnat, M. Probabilistic recognition of complex event. In Proceeding of the ICVS 2011, LNCS 6962, Sophia Antipolis, France, 20–22 September 2011; pp. 122–131.
24. Wang, Y.H.; Yang, S.H. Plan-based Distributed Complex Event Processing for RFID Application. In Proceedings of the International Conference on Computational Intelligence and Software Engineering, Wuhan, China, 11–13 December 2009; pp. 1–4.
25. Wang, Y.H.; Cao, K.; Zhang, X.M. Complex event processing over distributed probabilistic event streams. *Comput. Math. Appl.* **2013**, *66*, 1808–1821. [CrossRef]
26. Yao, Y.; Meng, C.; Wang, C.; Tian, Z.; Feng, D. Complex event processing in RFID-aided real-time production monitoring. *Mod. Manuf. Eng.* **2015**, *1*, 24–29.
27. Allen, L.V.; Tilbury, D.M. Anomaly Detection Using Model Generation for Event-Based Systems without a Preexisting Formal Model. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2012**, *42*, 654–668. [CrossRef]
28. Dias, J.R.S.; Maia, C.A.; Lucena, V.F. Synchronising operations on productive systems modelled by timed event graphs. *Int. J. Prod. Res.* **2016**, *54*, 4403–4417. [CrossRef]
29. Zaharia, M.; Xin, R.S.; Wendell, P.; Das, T.; Armbrust, M.; Dave, A.; Meng, X.; Rosen, J.; Venkataraman, S.; Franklin, M.J.; et al. Apache spark: A unified engine for big data processing. *Commun. ACM* **2016**, *59*, 56–65. [CrossRef]
30. Congosto, M.; Basanta-Val, P.; Sanchez-Fernandez, L. T-Hoarder: A framework to process Twitter data streams. *J. Netw. Comput. Appl.* **2017**, *83*, 28–39. [CrossRef]
31. Basanta-Val, P.; Fernández-García, N.; Wellings, A.J.; Audsley, N.C. Improving the predictability of distributed stream processors. *Future Gener. Comput. Syst.* **2015**, *52*, 22–36. [CrossRef]

32. Stoica, I. Trends and challenges in big data processing. *Proc. VLDB Endow.* **2016**, *9*, 1619. [CrossRef]
33. Basanta-Val, P.; Audsley, N.C.; Wellings, A.J.; Gray, I.; Fernández-García, N. Architecting time-critical big-data systems. *IEEE Trans. Big Data* **2016**, *2*, 310–324. [CrossRef]
34. Lv, Z.; Song, H.; Basanta-Val, P.; Steed, A.; Jo, M. Next-generation big data analytics: State of the art, challenges, and future research topics. *IEEE Trans. Ind. Inform.* **2017**, *13*, 1891–1899. [CrossRef]
35. Fang, J.; Huang, G.Q.; Li, Z. Event-driven multi-agent ubiquitous manufacturing execution platform for shop floor work-in-progress management. *Int. J. Prod. Res.* **2013**, *51*, 1168–1185. [CrossRef]
36. Cugola, G.; Margara, A. Processing Flows of Information: From Data Stream to Complex Event Processing. *ACM Comput. Surv.* **2012**, *44*. [CrossRef]
37. Zaytoon, J.; Lafortune, S. Overview of fault diagnosis methods for Discrete Event Systems. *Annu. Rev. Control* **2013**, *37*, 308–320. [CrossRef]
38. Li, Y.; Chang, Q.; Biller, S.; Xiao, G.X. Event-based modelling of distributed sensor networks in battery manufacturing. *Int. J. Prod. Res.* **2014**, *52*, 4239–4252. [CrossRef]
39. Dagkakis, G.; Papagiannopoulos, I.; Heavey, C. ManPy: An open-source software tool for building discrete event simulation models of manufacturing systems. *Softw. Pract. Exp.* **2016**, *46*, 955–981. [CrossRef]
40. Nie, J.; Sun, R.; Deng, X. Uncertain Complex Event Processing in Precision Agriculture Based on Data Provenance Management. *Trans. Chin. Soc. Agric. Mach.* **2016**, *47*, 245–253.
41. Li, X.; Gao, H.; Qiao, Y. Real-time Complex Event Processing Engine in Internet of Things. *J. Chin. Comput. Syst.* **2015**, *36*, 2047–2052.
42. Chen, W.; Li, S. Implementation technology and method for active perception of critical events in IOMT-based manufacturing process. *Manuf. Autom.* **2015**, *30*, 148–152.
43. Chen, W.; Li, S.; Huang, H. Active perception and management model for manufacturing data in discrete IoMT-based process. *Comput. Integr. Manuf. Syst.* **2016**, *22*, 166–176.
44. Zhang, Y.; Sun, S.; Wang, J.; Si, S.; Yang, T. Embedded Device and Implementing Method for Multi-Source Information Perception. CN Patent 201310100948.X, 27 March 2012.
45. Bai, A.; Tang, R.Z.; Lu, J.X. Semantic analysis of discrete shop logistics real-time data oriented to ubiquitous computing-based intelligent manufacturing. *Comput. Integr. Manuf. Syst.* **2012**, *18*, 550–559.
46. Zhan, C.; Fan, Y. Complex event processing of real time enterprises based on smart items. *Chin. J. Mech. Eng.* **2007**, *43*, 22–32.
47. Zhang, C.; Fan, Y. Complex event processing of manufacturing enterprises based on smart items. *Comput. Integr. Manuf. Syst.* **2007**, *13*, 2243–2253.
48. Sun, L.; Li, Y.; Li, J. The CEP-based correlation solution of monitoring events on distributed IT resources. *Comput. Appl. Softw.* **2013**, *30*, 302–306.
49. Kia, T.; Adrian, P. Semantic Rule-Based Complex Event Processing. *Lect. Notes Comput. Sci.* **2009**, *5858*, 82–92.