doi:10.6041/j.issn.1000-1298.2015.S0.026

面向农田物联网复杂事件处理的时空事件模型*

李　想　王建仑　高红菊
(中国农业大学信息与电气工程学院, 北京 100083)

摘要: 在农田物联网中,控制命令往往由具有多维度信息的复杂事件触发,因此需要根据传感器所监测的大量单一事件检测复杂事件,即复杂事件处理。复杂事件模型描述了原子事件组合成复杂事件的组合模式,是复杂事件处理的基础。现有的复杂事件模型主要考虑原子事件的时间分布特性,而未考虑农田事件具有的空间分布特性。本文研究了面向农田物联网复杂事件处理的时空事件模型并定义了合适的事件模型描述语言,首先建立了描述复杂事件模式的时空事件模型,针对多个复杂事件间的关系,通过有向图建立了复杂事件层次模型;针对一个复杂事件内部关系,通过对农田物联网事件常见时序逻辑关系和空间拓扑关系的分析,定义了 9 类时间关系算符、8 类空间关系算符用于判断子事件与复杂事件间的时空关系,并定义了 5 类时间耦合算符、7 类空间耦合算符用于计算复杂事件的时空属性,为描述事件组合模式提供了基础;在此基础上,考虑可读性和易解析性,设计了基于 XML 的事件模型描述语言;基于案例,通过与其他常用的复杂事件模型相比较,说明了本文事件模型和描述语言更加适用于描述农田物联网复杂事件。

关键词: 农田物联网　复杂事件处理　事件模型　事件描述语言
中图分类号: TP391.4　　**文献标识码**: A　　**文章编号**: 1000-1298(2015)S0-0153-09

Time and Space Event Model for Complex Event Processing in Internet of Things in Farmland

Li Xiang　Wang Jianlun　Gao Hongju
(*College of Information and Electrical Engineering*, *China Agricultural University*, *Beijing* 100083, *China*)

Abstract: In internet of things in farmlands, control commands are usually triggered by complex events which contain many dimensionalities of information. Complex events are detected by combining mass of atomic events directly sensed by sensors. That is a typical procedure of complex event processing (CEP). As a basis of CEP, a model of complex events, which describes how atomic events construct complex events, is necessary. Current complex event models focus on temporal logic, but spatial logic relations among farmland events are not considered. In this paper, a novel model that considers both temporal logic and spatial logic was proposed and a relevant XML-based specification language was designed. In the model, multi-level complex events were modeled to a directed graph. For one complex event in the graph, based on analysis of temporal and spatial logic relations among component events, nine time relation operators and eight spatial relation operators were defined to describe above relations; five time combination operators and seven spatial combination operators were defined to calculate attributes of complex events from attributes of atomic events. Based on the model, a XML-based language was defined, which balances readability of users, descriptive abilities and difficulties of parsing and compiling. The proposed event model and language were compared to those used in popular CEP systems e.g. SASE +, Esper, etc. From the comparison, the proposed model and language were more suitable for describing complex events in internet of things in farmlands.

Key words: Internet of things in farmland　Complex event processing　Event model　Event specification language

收稿日期: 2015-10-28　修回日期: 2015-11-09
*国家自然科学基金资助项目(31371531)
作者简介: 李想,讲师,博士,主要从事物联网实时复杂事件处理研究,E-mail: cqlixiang@cau.edu.cn
通讯作者: 高红菊,副教授,博士生导师,主要从事物联网数据融合研究,E-mail: hjgao@cau.edu.cn

引言

农田物联网是物联网在农业领域应用的重要形式,其作用是通过各类传感器设备监测农田土壤、环境、作物等信息,用于指导农田控制措施,如灌溉、施肥、喷药等。一个传感器所监控的是某个地点某个时刻的某一原子信息,而农田控制措施应对的往往是有意义的复杂情景,由数个或数十个状态量或变化量反映。因此需要引入数据决策专家系统,综合分析传感器数据,自动指导控制措施的实施。

复杂事件处理(Complex event processing,CEP)是一种脱胎于数据库领域的主动数据处理技术,是构建数据处理专家系统的一种常用方法。它将单一数据状态变化视作原子事件,将复杂情景视作复杂事件,通过将原子事件的发生情况与复杂事件模式相匹配及逻辑计算,检测复杂事件是否发生、何时发生、事件内容等情况,从而触发进一步的动作。

进行复杂事件处理,首先需要建立事件模型,即对什么是原子事件和复杂事件以及原子事件如何组合成复杂事件的模式进行定义。同时,需要一种兼顾可读性与易解析性的描述语言,即方便领域专家描述复杂事件模型的同时在计算机中能方便解析复杂事件的处理结构。

农田物联网传感器数据不仅有时间分布特点,还具有较为广域的空间分布特点。在描述农田物联网复杂事件时,空间关系与时间关系同样重要。

在事件模型方面,早期的复杂事件处理系统采用与处理结构相同的模型描述。如 HiPac 系统[1]、SAMOS 系统[2]、COMPOSE 系统[3]分别采用有向图、Petri 网、有限状态自动机模型化复杂事件。近些年来,事件模型得到了长足的发展,Cayuga 系统[4]改进自动机模型提出了 NFA 模型,从确定性的有限状态自动机改进为依据时序条件的非确定性状态机。SASE + 系统[5]等继续改进该模型,支持特殊的操作符,如克林闭包等。数据流研究者将复杂事件处理过程视作数据查询过程,基于数据库的关系型模型扩展了复杂事件模型[6]。孟由等[7]发展了基于图的事件模型,建议通过算子连接事件的形式描述复杂事件模型。针对智能家居环境的特点,IBM Amit 系统也严格定义了一系列算子,将情景描述方法引入复杂事件模型[8]。

在复杂事件描述语言方面,不同的事件模型有不同的描述方式。其中 NFA、有限状态自动机、Petri 网模型、类编程语言的描述方式使用较多[9]。对于数据流中的关系型扩展模型的事件模型,类 SQL 语句较为流行[10],该方法得到了 Esper、SQL Server StreamInsight 等商业系统的采用。针对算子类复杂事件模型,常采用经过设计的 XML 语言描述复杂事件[11],并且设计了配套的图形化定义工具[12]。

现有复杂事件处理技术所定义的事件模型主要考虑了成员事件间的时间逻辑关系,而没有考虑空间逻辑关系,相应的描述语言对这一点也缺乏考虑。原因是,复杂事件处理技术此前多用于工业领域,如生产流程控制等[11-13]。针对这类场景,事件的时序信息较为重要,而事件源的空间信息不需要特别关注,只需要通过指派 id 等方式区分不同事件源即可。对空间关系描述能力的不足,将影响现有复杂事件模型及描述语言用于农田物联网事件处理。虽然聂娟等[14]考虑了精准农业信息物理融合系统(CPS)的时空事件模型,但仅针对较为简单的情况,未针对较复杂的时空逻辑关系。

本文考虑农田物联网事件的时空逻辑关系,研究并建立一种面向农田物联网的时空事件模型,在此基础上建立一种复杂事件模型描述语言,为今后的农田物联网复杂事件处理研究奠定基础。

1 农田物联网复杂事件模型及描述语言

1.1 基本概念

定义 1:农田物联网事件是农田物联网所监测的对象,也就是作物、土壤、环境等因素的状态变化。用 E 表示事件,S 表示状态,则:

$$E \equiv S_0 \rightarrow S_1 \quad (S_0 \neq S_1) \tag{1}$$

农田物联网中的原子事件是一个传感器在一个时刻监控到的一次单点状态变化。在一定粒度下,可认为它在空间上的一个点发生,也在时间轴上的一个点发生。

农田物联网中的复杂事件是一种复杂情景,是一种有意义的事件,是复杂的状态变化,由多个原子事件或其他复杂事件聚合在一起决定是否发生。一个复杂事件可表示为

$$CE = f(PE_1, PE_2, PE_3, \cdots, CE_1, CE_2, CE_3, \cdots) \tag{2}$$

CE 表示复杂事件模式,PE 表示原子事件,f 是逻辑算符,定义了事件间的逻辑关系,可多层嵌套。

定义 2:事件类型,用大写字母表示,如 E,$E = (i, a)$,$a = (d, t, r, p, o)$,i 表示唯一标识事件类型,a 表示事件属性集合,即 $a = \{a_1, a_2, \cdots, a_n\}$,$n \geqslant 0$。一个事件包含如下几个属性:$d$ 表示事件发生的时间段,即 $d = [t_0, t_1]$,t_0 为开始时间,t_1 为结束时间。t 表示发生时间,在实践中,往往指定一个时间点作

为事件的发生时间。r 表示事件发生的空间范围。$p=(x,y,z)$ 表示事件的发生位置，在实践中，往往指定一个点作为事件的发生位置。对于原子事件，d、r 均收缩为点 t、p。o 表示其他属性。

定义 3：事件实例，是指某一类型事件的一次发生，用小写字母表示，如 e。用操作符 ins 表示事件类型实例集合，如：$e \in ins(E)$。原子事件一个实例在四维时空中映射为一个点。用 $E.x$ 或 $e.x$ 表示事件 E 或事件实例 e 的 x 属性。

1.2　典型场景描述

复杂事件处理系统应用的典型场景，其输入往往是从多事件源汇集的原子事件流，其输出是处理后的复杂事件流。

由于农田物联网传感器广域分布式的特点，事件源的空间位置信息成为组合复杂事件的重要参考因素，不同事件源的数据采集周期和数据类型具有较大差异。如果将输入事件合并为一个事件流，将单方面强化事件之间的时序逻辑，而弱化空间拓扑关系，不便于处理。因此，保留空间分布特征，原子事件流可由多路输入，复杂事件流仍由单路输出。多路输入有助于对每一路输入施加不同控制策略，从而更准确地处理复杂事件。如图 1 所示。

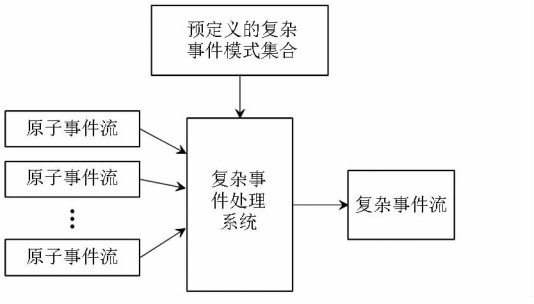

图 1　农田物联网复杂事件处理的典型场景

Fig. 1　Typical scene of complex event processing in internet of things in farmland

输入原子事件流：每一路输入的原子事件流包含来源于一个数据源（传感器或其他数据源）的事件实例序列。每个事件实例至少带有时间戳，即发生时间属性 t，及数据源的空间位置信息，即发生位置属性 p。在事件流中，所有原子事件实例均按照发生时间先后排序。同一个输入事件流中事件实例的 p 属性均相同。

输出复杂事件流：复杂事件处理系统经过处理输出复杂事件流。其中每一个复杂事件均根据预定义的复杂事件模式匹配成员事件的时间、空间属性关系及其他属性关系得到。每个复杂事件至少带有 2 个属性：发生时间段 d 及发生范围 r，通过复杂事件中成员事件的时间、空间属性计算耦合得到。

1.3　时空事件模型

1.3.1　模型概览

一个复杂事件，往往是多层次的，即复杂事件的子事件可以是原子事件也可以是另一些复杂事件。层次关系可用有向图描述：有向图的叶节点是原子事件，而上层节点均是复杂事件，如图 2 所示；也可用类似函数嵌套的方式表示，例如复杂事件 $A_1(B_1(D_1,E_1),C_1(F_1))$。

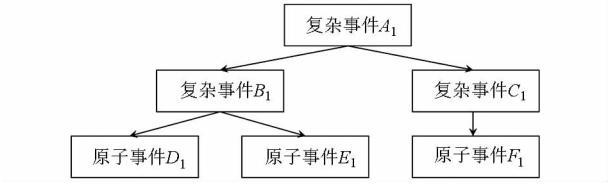

图 2　事件层次关系示例

Fig. 2　An example of hierarchical relations among events

针对任意层次复杂事件，事件模型将回答：该复杂事件由哪些子事件（包含原子事件或其他复杂事件）构成，这些子事件选取哪些实例、如何匹配才能说明该复杂事件发生，该复杂事件实例如何通过子事件实例属性集构造得到。

因此，一个复杂事件模式可表示为四元组，即

$$CE \equiv (ES, IS, FJ, FC) \tag{3}$$

ES 是该复杂事件的子事件集，即

$$ES = (E_{i1}, E_{i2}, \cdots, E_{in}) \tag{4}$$

IS 是每个子事件实例队列的使用、消耗、窗口策略函数，即

$$IS = (IS_U, IS_C, IS_W) \tag{5}$$

复杂事件不仅需要考虑子事件间的时间关系、空间关系，还由于每个子事件类型都可产生大量实例构成实例队列，如果不进行控制，需要进行的模式判断数量将呈几何级数增长，直至耗尽 CPU 处理资源或储存空间资源。因此，需要从 3 方面对此进行控制。定义使用策略函数 IS_U：指明选取哪些子事件实例来进行聚合，即制定事件实例使用策略。定义消耗策略函数 IS_C：指明一个事件实例参与聚合复杂事件后，被删除掉的条件。定义窗口策略函数 IS_W：指明滑动窗口，硬性控制输入实例队列长度。

IS_U 决定了下一时刻所耦合的子事件的集合。IS_C 和 IS_W 的交集决定了某一子事件处理后一时刻的实例队列，即

$$ins_t(E) = IS_C(ins_{t-1}(E)) \cap IS_W(ins_{t-1}(E)) \tag{6}$$

FJ 是判断函数，通过计算子事件实例属性集，判断复杂事件是否发生，其结果是布尔值（True 或 False），即

$$FJ(\{e,a \mid \forall e \in IS_U(ins(E_{i1}), ins(E_{i2}), \cdots, ins(E_{in}))\}) = \begin{cases} \text{True} & (\text{事件发生}, E_{ij} \in ES) \\ \text{False} & (\text{事件未发生}, E_{ij} \in ES) \end{cases} \tag{7}$$

判断函数将用于判断事件的发生时间 t、范围 d、发生位置 p、空间范围 r 以及其他属性 o 是否满足逻辑关系的要求,即

$$\begin{cases} FJ = FJ_{d,t} \ \& \ FJ_{r,p} \ \& \ FJ_o \\ FJ_{d,t} \equiv TJ(e_{i1}.t, e_{i1}.d, \cdots, TJ_1(e_{j1}.t, e_{j2}.d, \cdots), \cdots) \\ FJ_{r,p} \equiv SJ(e_{i1}.p, e_{i1}.r, \cdots, SJ_1(e_{j1}.p, e_{j2}.r, \cdots), \cdots) \\ FJ_o \equiv OJ(e_{i1}.o, \cdots, OJ_1(e_{j1}.o, \cdots), \cdots) \end{cases} \tag{8}$$

式(8)中,$FJ_{d,t}$是时间判断函数,由时间关系算符 TJ 连接事件实例的时间属性 t 和时间段属性 d 组成,用于判断事件发生时域属性间的时序关系及时间数值关系。$FJ_{r,p}$是空间判断函数,由空间关系算符 SJ 连接事件实例的发生范围属性 r 和发生位置属性 p 组成,用于判断事件的发生地点或发生地域属性间的拓扑关系。FJ_o 是其他判断函数,由一般的数值比较关系算符($=$、$\neq$、$<$、$>$、$\leqslant$、$\geqslant$)连接事件实例的其他数值属性构成,用于判断事件其他属性间的关系。所有的判断算符均可多层次嵌套。

FC 是属性计算函数,通过子事件实例属性集计算复杂事件 CE 一次检测产生的实例 ce_1, ce_2, …, ce_m 的属性,即

$$\begin{aligned} &\{ce_1.a, ce_2.a, \cdots, ce_m.a\} = \\ &FC(\{e.a \mid \forall e \in IS_U(ins(E_{i1}), \\ &ins(E_{i2}), \cdots, ins(E_{in}))\}) \quad (E_{ij} \in ES) \end{aligned} \tag{9}$$

属性计算函数将分别计算事件属性时间及范围 t、d,发生位置及空间范围 p、r,以及其他属性 o 5 个方面,即

$$\begin{cases} FC \equiv (FC_{d,t}, FC_{r,p}, FC_o) \\ FC_{d,t} \equiv TC(e_{i1}.t, e_{i1}.d, \cdots, TC_1(e_{j1}.t, e_{j2}.d, \cdots), \cdots) \\ FC_{r,p} \equiv SC(e_{i1}.p, e_{i1}.r, \cdots, SC_1(e_{j1}.p, e_{j2}.r, \cdots), \cdots) \\ FC_o \equiv OC(e_{i1}.o, \cdots, OC_1(e_{j1}.o, \cdots), \cdots) \end{cases} \tag{10}$$

式(10)中,$FC_{d,t}$、$FC_{r,p}$、FC_o 分别是时间属性计算函数、空间属性计算函数和其他属性计算函数,分别描述通过时间耦合算符 TC、空间耦合算符 SC 以及其他耦合算符 OC 多层次嵌套计算子事件的 t、d、p、r、o 等属性值,得到父事件的相应属性值。

下面通过对时间逻辑和空间逻辑关系的分类,详细讨论时间、空间关系算符及耦合算符 $FJ_{d,t}$、$FJ_{r,p}$、$FC_{d,t}$、$FC_{r,p}$。

1.3.2 时间关系算符 $FJ_{d,t}$

原子事件的发生是瞬间的,因此其发生时间是一个时间点。时间点之间的时序关系包括“早于、晚于、同时”3 种。注意,“早于”和“晚于”对称。

定义 4:定义时间点关系判断算符 *sync*, *after*:*sync*(A, B)表示 A、B 同时发生;*after*(A, B)表示 B 在 A 之后发生。

复杂事件发生可持续一段时间,因此其发生时间是一个时间段。时间段之间的时序逻辑关系有 13 种,如图 3 所示。

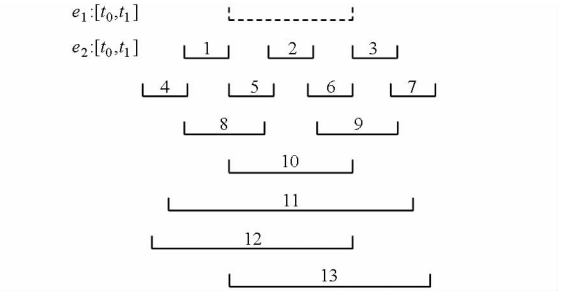

图 3 2 个时间段之间的 13 种时序逻辑关系

Fig. 3 13 kinds of temporal logic relations between two periods of time

用事件实例的开始时间和结束时间关系即可表示事件间的时间关系。

定义 5:时间段二元关系算符 *in* 表示一个事件的发生时间在另一个事件的发生时间内,即

$$in_{align}(e_1, e_2) \equiv e_1.t_0 \geqslant e_2.t_0 \ \& \ e_1.t_1 \leqslant e_2.t_1 \tag{11}$$

对应图 3 中情况 10、11、12、13。其中,情况 2 与 11、5 与 13、6 与 12 是对称的。区分这几种情况可以用条件 *align* 表示,其取值可以是 *begin*、*end*、*both*、*none*,分别为开始时间对齐、结束时间对齐、都对齐、不对齐。例如,图 3 中的情况 13 可表示为 $in_{begin}(e_1, e_2)$,意味着 $e_1.t_0 = e_2.t_0$ & $e_1.t_1 \leqslant e_2.t_1$。

时间段二元关系算符 *follow* 表示一个事件发生在另一个事件之后,即

$$follow_{align}(e_1, e_2) \equiv e_1.t_0 < e_2.t_0 \ \& \ e_1.t_1 < e_2.t_1 \tag{12}$$

对应图 3 中情况 3、7、9。其中情况 1 与 3、4 与 7、8 与 9 是对称的。区分这几种情况可用条件 *align* 表示,其取值分别为 *overlap*、*tangency*、*separate*,分别表示有交集、相切、分开。例如,图 3 中情况 3 可表示为 $follow_{tangency}(e_1, e_2)$,意味着 $e_1.t_0 \leqslant e_2.t_0 = e_1.t_1 \leqslant e_2.t_1$。

1.3.3 空间关系算符 $FJ_{r,p}$

原子事件在一个空间点发生。空间点的二元关系有 2 种,即相同和分离。

定义 6:定义空间点的二元关系判断算符:*same*(相同)、*split*(分离)。

复杂事件可以发生在一定的空间范围内。空间范围的二元关系通过点集定义有 8 种,即相离、外切、相交、内切、严格包含、相等、被内切、被包含[14],如图 4 所示。

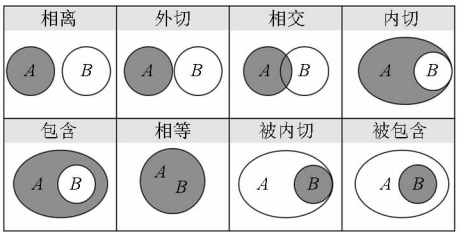

图 4　简单空间关系描述

Fig. 4　Simple spatial relations

定义 7：定义空间范围的二元关系判断算符：*disjoint*（相离）、*meet*（外切）、*overlap*（相交）、*cover*（内切）、*contain*（包含）、*equal*（相等）。

1.3.4　时间耦合算符 $FC_{d,t}$

原子事件发生时间点间的常见耦合方式有 2 种：通过对多个时间点的数值运算得到另一个时间点，或通过一定规则构造一个时间段。前者可以通过数值运算得到，下面给出一种构造后者的方式。

定义 8：时间点耦合算符 *continue*：用 t 表示时间点，则 *continue* 算符表示从众多时间点中选择最早和最晚的时间点作为首尾构造时间段，语义为

$$continue(t_1,t_2,\cdots,t_i,\cdots,t_k)\equiv[\min t_i,\max t_i] \tag{13}$$

复杂事件的发生可能持续一段时间，因此，复杂事件之间的耦合是时间段之间的运算。时间段之间的耦合可以通过计算开始时间和结束时间得到。这里定义几种常用的耦合算符。

定义 9：时间段耦合算符 *intervalunion*，表示求并集；*intervalintersection*，表示求交集；*intervaldifference*，表示求差集；*intervalcontinue*，表示分别将最早的一个开始时间和最晚的一个结束时间作为首尾构造时间段。

1.3.5　空间范围耦合算符 $FC_{r,p}$

对于原子事件，事件发生位置点之间的常见耦合方式有 2 种：通过对多个位置点的数值运算得到另一个位置点，或者通过一定规则构造一个范围。前者可以通过数值计算得到，不需要特别定义算符，而后者，给出算符定义如下：

定义 10：位置点耦合算符 *polygon*：表示形成以所有位置点为顶点的多边形（二维）或多面体（三维）。位置点耦合算符 *circle*：表示构造恰好能容纳所有位置点的圆形区域（二维）或球形区域（三维）。

对于复杂事件，事件发生范围之间的耦合可以通过计算集合运算得到。这里定义几种常用的耦合算符。

定义 11：空间范围耦合算符 *areaunion*，表示求并集；*areaintersection*，表示求交集；*areadifference*，表示求差集；*areapolygon*，表示以所有顶点重新构造多边形；*areacircle*，表示构造恰好能容纳所有空间范围的圆形区域（二维）或球形区域（三维）。

1.3.6　事件实例使用消耗策略

定义 12：4 种事件实例选择策略[1]，有 *recent*（最新）：事件实例队列中，最新加入的实例参与聚合复杂事件；*chronicle*（最旧）：事件实例队列中，最早加入的实例参与聚合复杂事件；*continuous*（连续）：事件实例队列中的每个实例均参与聚合复杂事件，例如 *conjunction*(A,B)，A、B 的实例队列分别为 $\{a^1,a^2\}$ 和 $\{b^1,b^2\}$，那么在连续策略下将产生4 组复杂事件实例 (a^1,b^1)、(a^1,b^2)、(a^2,b^1)、(a^2,b^2)。*cumulative*（累积）：队列中所有实例都参与聚合复杂事件，如上例中，在累积策略下，将产生一组复杂事件实例 $((a^1,a^2),(b^1,b^2))$。

事件实例的消耗策略，是指当事件实例参与聚合复杂事件后，移除部分事件实例。

定义 13：事件实例消耗策略有：*reserve*（保留）：不移除任何实例；*removeone*（移除当前）：移除当前参与聚合复杂事件的事件实例；*removeearly*（移除更早）：移除当前参与聚合复杂事件的实例，同时移除比它早的实例；*removeall*（移除所有）：移除当前队列中所有复杂事件实例。

事件实例消耗策略虽然一定程度上减少了实例数量，但事件实例队列不可控，因此需要通过滑动窗口去硬性限定实例队列长度，有 2 种方式：时间滑动窗口和队列长度滑动窗口。

定义 14：窗口策略有：*timewindow*（时间滑动窗口）：距离现在时刻大于一定值的事件实例就从队列中移除。*lengthwindow*（队列长度滑动窗口）：队列中实例数量保持小于上限，超过上限，最旧的实例将被删除。

1.4　农田物联网复杂事件描述语言

基于事件模型设计复杂事件描述语言，需要满足无二义性、较强的可读性、便于计算机解析并处理等要求。这里，考虑可读性和计算机解析的方便，选用 XML 语言作为基础。

1.4.1　基本语法结构

复杂事件描述语言的基本语法结构包含原子事件注册和复杂事件模式集 2 部分，语法描述如下：

```
<cepl>
  <atomset> //原子事件注册表
    <atom> … </atom> //一个原子事件描述
    …
  </atomset>
  <ceset> //复杂事件模式集
    <ce> … </ce> //一个复杂事件模式
```

```
    …
  </ceset>
</cepl>
```

1.4.2 原子事件描述语法

每个原子事件将有一个全局唯一的标识符 *id* 并指出来源的传感器节点,这里多个传感器节点的同一类状态变化可指定为一种原子事件。

在属性方面,将默认包含来源传感器 s、发生时间 t、开始时间 t_0、结束时间 t_1、发生位置 p、影响范围 r 这几项属性。其他属性需要定义。

一个原子事件描述的语法如下:

```
<atom>
  <id>atomevent1</id> //该原子事件全局唯一id
  <sources> //该原子事件来源的传感器节点
    <src>sensor1</src> //一个来源
    …
  </source>
  <otherattrs> //除了默认属性外,定义其他属性
    <attr>o1</attr> //一条属性
    …
  </otherattrs>
</atom>
```

1.4.3 复杂事件模式描述语法

一个复杂事件模式,将描述子事件集 *ES*、实例策略 *IS*、判断函数 *FJ*、属性计算函数 *FC* 4 个方面,其语法结构如下:

```
<ce>
  <id>complexevent1</id> //该原子事件全局唯一id
  <otherattrs> //除了默认属性外,定义其他属性
    <attr>o1</attr> //一条属性
    …
  </otherattrs>
  <eventset> //子事件集 ES
    <event>…</event> //一个事件及其实例策略
    …
  </eventset>
  <judgefunctionlogic="conjunction|disjunction">
  //判断函数
    <jfunc>…</jfunc> //一条判断函数
    …
  </judgefunction>
  <constructorfunction> //属性计算函数
    <cfunc>…</cfunc> //一条属性计算函数
    …
  </constructorfunction>
</ce>
```

其中,实例策略与每个子事件相关,因此与子事件集共同描述。判断函数的 *logic* 属性可取值 *conjunction* 或者 *disjunction*,前者表示每条判断函数都必须为 True 才能说明复杂事件发生,后者表示只需要一条判断函数为 True 即可说明复杂事件发生。

(1) 子事件及实例策略描述语法

由于实例的使用、消耗和窗口策略是加载到每个子事件上的,因此,实例策略与子事件一同描述,语法如下:

```
<event>
  <evn>…</evn> //用于引用事件
  <usestrategy> //定义事件实例使用策略
    recent | chronicle | continuous | cumulative //选项
  </usestrategy>
  <consumestrategy> //定义事件实例消耗策略
    reserve | removeone | removeearly | removeall //选项
  </consumestrategy>
  <timewindow>time1</timewindow> //时间窗口
  <lengthwindow>10</lengthwindow> //长度窗口
</event>
```

这里事件实例使用策略取值共 4 个,分别为 *recent*(最新)、*chronicle*(最旧)、*continuous*(连续)、*cumulative*(累积)。事件实例消耗策略取值有 4 个,分别为 *reserve*(保留)、*removeone*(移除当前)、*removeearly*(移除更早)、*removeall*(移除所有)。

2 类窗口,即时间滑动窗口与队列长度滑动窗口之中,定义 1 个即可。当 2 个都定义时,表示以窗口更小的那个为准。

(2) 判断函数描述语法

```
<jfunc>
  <attr> //用于判断的事件属性类型
   time | interval | position | area | source | other
//可选项
  </attr>
  <variables> //用于判断的变量
    <var id="v1" attr="o1">atomicevent1</var
> //用于判断的一个事件及其属性
    <constant id="c1" type="default| float">
//用于判断的一个常量
      value //可取值见下文
    </constant>
    …
  </variables>
  <func>f(${v1},${c1},…)</func> //判断
```

函数

</jfunc>

其中，按照复杂事件模型，用于判断的事件属性应包含 6 种类型，包括 *time*（事件发生时间）、*interval*（事件发生时间范围）、*position*（事件发生位置）、*area*（事件影响范围）、*source*（事件来源）、*other*（事件其他属性）。

用于判断的变量可包括事件属性，也可包括一些常数。事件属性需要指明来自于哪个事件，当属性类型是 *other* 时，还需要指明其属性名。常量的类型有 2 种，*default* 表示与用于判断的事件属性 *attr* 定义一致；*float* 专用于定义常数，取值为浮点数。表 1 列举了常量取值示例。

表 1　常量的取值示例

Tab. 1　Examples for values of constants

类型	示例	说明
time	2015 - 09 - 01, 22:21:12.000	精确到毫秒
interval	[2015 - 09 - 01, 22:21:12.000; 2015 - 09 - 02, 22:21:12.000]	精确到毫秒
position	(1;3.1)	坐标位置
area	((1;2);(2;2);(2;1);(1;1), …)或[(1;2);1]	前者是多边形，定义其顶点位置；后者是圆形，定义圆心位置及半径
source	{sensor1;sensor2}	传感器名称集合
float	1.23	浮点数

判断函数中用“${变量名}”和“${常量名}”表示变量和常量。函数 *f* 可由时间判断算符或空间判断算符构成，如“*after*(${*v*1}, ${*v*2})”；或空间二元算符，如“*split*(${*v*1}, ${*c*1})”；或者任何逻辑表达式，如“${*v*2} - ${*v*1} ≤ ${*c*1}”。

（3）属性计算函数描述语法

<cfunc>

<attr>//用于计算的事件属性名

time | tbegin | tend | px | py | radius | …//可选项

</attr>

<variables>　//用于计算的变量

<var id = "v1" attr = "o1" > atomicevent1 </var>//用于构造的一个事件

…

</variables>

<func>f(${*v*1}, ${*v*2}, …) </func>//属性计算函数

</cfunc>

这里用于计算事件属性名的有 6 种保留值，也可选择 *otherattrs* 标签中预定义的属性名。保留属性名含义为：*time*（发生时间 t）、*tbegin*（发生时间范围开始 t_0）、*tend*（发生时间范围结束 t_1）、*px*（发生位置 p 的 x 坐标值）、*py*（发生位置 p 的 y 坐标值）、*radius*（影响范围半径 r）。这里，复杂事件的来源属性（*source*）不显式出现，由所有子事件的来源属性求并集得到。

变量中的 *attr* 属性取值与上述属性名相同，不再赘述。属性计算函数 *f* 可以是任何算式，如${*v*1} + ${*v*2}；也可以是任何时间耦合算符，如 *intervalcontinue*(${*v*1}, ${*v*2})；或者任何空间耦合算符 *areacircle*(${*v*1}, ${*v*2})。

2　与其他事件模型的比较及案例分析

2.1　案例分析

通过案例分析比较 SASE + 系统[5]所采用的及本文提出的事件模型及描述语言。SASE + 系统自从 2007 年公布以来，成为最流行的复杂事件处理系统之一，得到了广泛的应用[15-16]。

案例：“复杂事件 E：在喷头 1 和喷头 2 喷水（事件 A 与事件 B）后某地点仍然缺水（事件 C），该地在喷头 1、2 设计范围内。”

用本文介绍的事件模型描述该情景，首先，考虑时序关系：C 在 A、B 之后发生。需要 2 层算符嵌套，用 *intervalcontinue* 算符构造 A 与 B 的时间段用于约束 C。接下来，E 的判断函数用 $follow_{separate}$ 算符判断 C 该时间段的跟随关系。其次，考虑空间关系：C 的发生范围处于 A、B 的共同的影响范围中。仍然需要 2 层算符嵌套，首先采用 *areaintersection* 算符求 A、B 影响范围交集。接下来构造 E 的判断函数，用 *contain* 算符判断 C 该交集的包含关系。具体描述如下：

$$E:(ES=\{A,B,C\}$$

$$IS:\left\{\begin{array}{l} IS_U = recent \\ IS_C = removeall \\ IS_W = lengthwindow(1) \end{array}\right\}$$

$$FJ:\left\{\begin{array}{l} after(A,B) \\ follow_{separate}(C.d, intervalcontinue(A.d, B.d)) \\ contain(C.r, areaintersection(A.r, B.r)) \end{array}\right\}$$

$$FC:\{E.d = C.d, areaintersection(A.r, B.r)\})$$

采用 SASE + 语法描述该复杂事件如下：

$PATTERN: SEQ(A,B,C)$,

WHERE:

$$\begin{cases} A.t < B.t \wedge C.d.t0 \geqslant B.t \\ \sqrt{(A.p.x - C.p.x)^2 + (A.p.y - C.p.y)^2} < A.r.R - C.r.R \\ \sqrt{(B.p.x - C.p.x)^2 + (B.p.y - C.p.y)^2} < B.r.R - C.r.R \end{cases}$$

WITHIN:1*MIN*

RETURN:*C.d*

(15)

上述描述中,假设 A、B、C 的空间范围都是圆形,$A.r.R$、$B.r.R$、$C.r.R$ 分别是其半径。

从上面的对比可以看到,相对于 SASE + 语言,本文提出的语言更加直观,特别是在表示空间属性的时候。例如上述描述中 SASE + 需要用多个算式进行几何图形包含关系的运算,A、B、C 的影响范围几何形状越复杂,算式将越多,算式的内容也将越复杂,可读性较差。另外,在 *RETURN* 中难以表示 $A.r$ 和 $B.r$ 的交集。

而本文定义的算符只需要 *contain*、*areaintersection* 算符就能清楚表意,可读性更强。对 *contain*、*areaintersection* 算符的计算细节交给复杂事件处理系统完成,不需要对用户展现。

2.2 与其他事件模型的比较

表 2 列举了本文的事件模型与常用的 SASE +[5]、Esper[17]、臧传真[11](简称臧模型)提到的模型间对不同特性支持的比较。

从表 2 可以看出,本文提出的事件模型和复杂事件模式描述语言,在描述具有空间逻辑关系的复杂事件时,表意能力更强,且支持自定义属性及针对不同输入事件流选择不同控制策略,进一步增强了表意能力。同时,采用 XML 语法,增强了可读性和易解析性。

表 2 不同事件模型的比较

Tab. 2 Comparison among different event models

特性	本文模型	SASE +	Esper	臧模型
支持时序关系	支持	支持	支持	支持
支持空间逻辑关系	支持	不支持	不支持	不支持
支持针对每个输入事件流选择不同策略	支持	不支持	不支持	不支持
支持克林闭包	不支持	支持	不支持	不支持
支持自定义属性	支持	部分支持	支持	支持
易读性	较好	较好	较差	较好
易解析性	较好	较差	好	较好

3 结论

(1) 针对农田物联网复杂事件具有时空性的特点,基于时序逻辑、空间拓扑建立了面向农田物联网复杂事件处理的时空事件模型。通过定义 9 类时间关系算符、8 类空间关系算符、5 类时间耦合算符、7 类空间耦合算符,用于判断计算复杂事件的时空属性,为描述复杂事件组合模式奠定了基础。

(2) 在事件模型基础上,基于 XML 设计了复杂事件描述语言。

(3) 通过案例分析验证了语言的表意能力比常用的其他模型和语言更适应农田物联网的要求。

参 考 文 献

1 Chakravarthy S, Mishra-Snoop D. An expressive event specification language for active databases[R]. University of Florida, Technical Report UF-CIS-TR-93-007, 1993.

2 Gatziu S, Dittrich K R. SAMOS: an active object-oriented database system[J]. IEEE Bulletin of the Tc on Data Engineering, 1992, 15(1-4): 23-26.

3 Gehani N H, Jagadish H V, Shmueli O. Event specification in an active object-oriented database[J]. ACM Sigmod Record, 1992, 21(2): 81-90.

4 Demers A J, Gehrke J, Panda B, et al. Cayuga: a general purpose event monitoring system[C]// CIDR 2007 Proceedings of 3rd Bienniol Conference on Innovative Date System Research, 2007: 412-422.

5 Gyllstrom D, Agrawal J, Diao Y, et al. On supporting kleene closure over event streams[C]// Proceedings of IEEE 24th International Conference on Data Engineering, 2008: 1391-1393.

6 Krämer J, Seeger B. Semantics and implementation of continuous sliding window queries over data streams[J]. ACM Transactions on Database System, 2009, 34(1), Aricle No.4.

7 孟由, 栾钟治, 谢明, 等. 一种基于算子的可扩展复杂事件处理模型[J]. 软件学报, 2014, 25(11): 2715-2730.
Meng You, Luan Zhongzhi, Xie Ming, et, al. Operator-based extendable complex event processing model[J]. Journal of Software, 2014, 25(11): 2715-2730. (in Chinese)

8 Magid Y, Adi A, Barnea M, et al. Application generation framework for real-time complex event processing[C]// Proceedings of the 32nd Annual IEEE International Computer Software and Applications Conference, 2008: 1162-1167.

9 Anicica D, Rudolphb S, Fodorc P, et al. Real-time complex event recognition and reasoning—a logic programming approach[J]. An International Journal of Applied Artificial Intelligence, 2012, 26(1-2): 6-57.

10 Rao J, Doraiswamy S, Thakkar H, et al. A deferred cleansing method for RFID data analytics[C]// Proceedings of the 32nd

International Conference on Very Large Data Bases. VLDB Endowment, 2006: 175 – 186.

11 臧传真, 范玉顺. 基于智能物件的实时企业复杂事件处理机制[J]. 机械工程学报, 2007, 43(2): 22 – 32.
Zang Chuanzhen, Fan Yushun. Complex event processing of real time enterprises based on smart items[J]. Chinese Journal of Mechanical Engineering, 2007, 43(2): 22 – 32. (in Chinese)

12 Liu W, Qiao Y, Li X, et al. A visual specification tool for event-condition-action rules supporting web-based distributed system [C]// Proceedings of the International Conference on Enterprise Information Systems. ICEIS, 2008: 246 – 251.

13 沈敬伟, 温永宁, 闾国年,等. 时空拓扑关系描述及其推理研究[J]. 地理与地理信息科学, 2010, 26(6): 1 – 5.
Shen Jingwei, Wen Yongning, Lü Guonian, et al. Representation and reasoning about spatial-temporal topological relationships [J]. Geography and Geo-Information Science, 2010, 26(6): 1 – 5. (in Chinese)

14 聂娟, 孙瑞志, 曹振丽, 等. 精准农业信息物理融合系统的事件模型研究[J]. 农业机械学报, 2015, 46(1): 285 – 291.
Nie Juan, Sun Ruizhi, Cao Zhenli, et al. Preliminary study on event model in cyber-physical systems for precision agriculture [J]. Transactions of the Chinese Society for Agricultural Machinery, 2015, 46(1): 285 – 291. (in Chinese)

15 Mao N, Tan J. Complex event processing on uncertain data streams in product manufacturing process[C]// 2015 International Conference on Advanced Mechatronic Systems (ICAMechS), IEEE, 2015: 583 – 588.

16 彭商濂, 李战怀, 李强, 等. RFID 数据流上多目标复杂事件检测[J]. 计算机研究与发展, 2015, 49(9): 1910 – 1925.
Peng Shanglian, Li Zhanhuai, Li Qiang, et al. Multiple objects event detection over RFID data streams[J]. Journal of Computer Research and Development, 2015, 49(9): 1910 – 1925. (in Chinese)

17 EsperTech. Esper: event processing for Java [EB/OL]. [2015 – 09 – 01]. http://www.espertech.com/products/esper.php.