

基于事件逻辑的 CPS 组件协同模型

尹忠海, 褚亚男✉

(空军工程大学理学院, 西安, 710051)

摘要 针对 CPS 系统的异构性、实时性、物理世界与信息世界的高度融合等特征, 提出了基于事件逻辑关系的组件协同代数模型。首先, 定义了基于时空事件接口的组件形式化表示方法, 建立了基于事件驱动的组件协同体系架构。然后, 根据组件协同方式分析了事件间的逻辑关系, 建立了组件协同代数系统。最后, 以智能家居的非法入侵行为为例, 使用协同代数式描述事件驱动组件行为的过程。分析表明, 基于事件逻辑的组件协同代数模型能够清晰地反应组件间的交互关系, 屏蔽了组件的异构性, 实现了物理进程与计算进程的融合, 为 CPS 系统的形式化建模提供了新的方法。

关键词 物理信息融合系统; 事件接口; 事件逻辑关系; 组件协同代数系统

DOI 10.3969/j.issn.1009-3516.2017.05.012

中图分类号 TP301 **文献标志码** A **文章编号** 1009-3516(2017)05-0067-06

Event Logic-based Component Coordination Model of CPS

YIN Zhonghai, CHU Yanan✉

(Science College, Air Force Engineering University, Xi'an 710051, China)

Abstract: According to the characteristics of heterogeneity, instantaneity, and high integration of physics and cyber in CPS, an event logic-based component coordination algebra model is built. Firstly, a temporal-spatial event interface based on component formal expression is defined. and an Event-Driven based component coordination behavior model is proposed. Then, according to the coordination modes between components, logical relationships between events are analyzed. Finally, taking the illegal invasion into smart home as an example, the paper uses the coordination algebra model to describe the process of event-driven component conduct. The analysis show that event logic-based component coordination model can realize interaction between components and integration between physical process and computational process, shield the heterogeneity between components, and provides a novel method for formal modeling of CPS.

Key words: cyber-physical system; event interface; logic relations between events; component coordination algebra system

信息物理融合系统 (Cyber-Physical Systems, CPS) 是一类融合计算、通信和控制的新型智能复杂

收稿日期: 2017-04-27

基金项目: 国家自然科学基金(61472443)

作者简介: 尹忠海(1964—), 男, 陕西西安人, 教授, 主要从事计算智能和无线自组织网络方面的研究。E-mail: yinzhonghai2005@yahoo.com.cn

通讯作者: 褚亚男(1984—), 女, 河北沧州人, 硕士生, 主要从事计算数学与智能信息处理研究。E-mail: chuyanana_229@sina.com

引用格式: 尹忠海, 褚亚男. 基于事件逻辑的 CPS 组件协同模型 [J]. 空军工程大学学报(自然科学版), 2017, 18(5):67-72. YIN Zhonghai, CHU Yanan. Event Logic-based Component Coordination Model of CPS [J]. Journal of Air Force Engineering University (Natural Science Edition), 2017, 18(5):67-72.

系统,强调计算控制子系统与被控制的物理设备和子系统间的实时协同交互^[1-2]。构建 CPS 模型的难点之一是实现多异构模型的互操作^[3]。一种模型设计方法是基于组件的形式建模,使用统一的组件形式化方法定义异构的物理设备和计算及控制单元,利用标准的组件接口构建协同实体模型,实现组件间互通互连^[4-5]。这种基于组件的 CPS 体系结构将计算和协同分离,有利于实现系统的动态性、可扩展性和验证性。

目前组件协同主要是基于消息传递或功能上的逻辑关系来实现的。文献[6]定义了标准的消息表达形式,构建了语义明确的基于消息的树形组件协同模型。文献[7]根据组件间的功能调用或消息驱动等定义组件组合运算,使用进程代数描述系统行为,建立了组件组合代数系统。文献[8]分析组件接口间的交互方式,定义了连接代数系统,以一种有形的结构化概念直观的反应组件分层结构。文献[9]采用 Tile 模型描述重写规则,并引入了互斥连接,拓展了组件同步规则。文献[10]在 BIP 组件框架基础上提出组件间的 3 种依赖关系,使用非循环 Petri 网描述组件行为。为满足分布式系统的实时性要求,普遍采用事件驱动(Event-Driven Architecture, EDA)与面向组件和服务相结合的方式建立系统行为模型。文献[11]采用 EDA 实现物理和计算单元关于感知、计算和控制决策的功能。文献[12]将面向服务架构与事件驱动架构思想结合,引入服务总线结构,实现了服务的即插即用和服务之间松散灵活的交互关系。文献[13]基于 Petri 网建立了嵌入式系统的设备、计算与物理交互、组件及通信过程的模型。

以上建立的组件协同模型在分布式计算系统中能够清晰地描述组件协同语义,但不能直接应用于具有时空特性的异构系统 CPS。本文从事件逻辑关系角度建立 CPS 组件协同模型,实现了组件间的紧密衔接,避免了不必要的资源浪费。

1 基于事件驱动的组件协同机制

1.1 组件及接口形式化定义

定义 1 接口形式化表示方法。定义组件接口 I 为事件接口,即:

$$I = \Gamma \{C_{id}, S_{id}, J_{id}, A(R), T^o, L^o, T^g, L^g\}$$

其中: Γ 表示事件类型名; C_{id} 表示组件 ID; S_{id} 表示事件类型 Γ 实例的生成者 ID; J_{id} 表示被观察实体 ID^[14]; $A(R)$ 表示事件所关联的属性信息; T^o, L^o 表示事件发生的时间和地点, T^g, L^g 表示事件产生的

事件和地点^[15]。对于 CPS 系统,本文对事件的时空一致性不做深入研究,具体见文献[15]。

定义 2 组件形式化表示方法。将组件用七元组 $C = (C_{id}, Q_d, Q_c, Q_{init}, H(Q_d, Q_c), E, \rightarrow)$ 表示, Q_d 为离散状态集; Q_c 为连续状态集,用与时空属性相关的连续变量描述; Q_{init} 为初始离散状态和连续状态的集合; $H(Q_d, Q_c)$ 表示每个离散状态对应的连续状态的集合; $E = \{I_1, I_2, \dots, I_n\}$ 为 C_{id} 的事件接口集,表示系统的状态转移的触发条件; $\rightarrow \subset H(Q_d, Q_c) \times E \times H(Q_d, Q_c)$ 为状态转移集合。已知状态 $q, q' \in H(Q_d, Q_c)$, 接口 $\Gamma \in E$, 若 $q \xrightarrow{\Gamma} q'$, 则 $(q, \Gamma, q') \in \rightarrow$ 。

若组件 $C = (C_{id}, Q_d, Q_c, Q_{init}, H(Q_d, Q_c), E, \rightarrow)$ 由 n 个子组件 $C_i = (C_{id_i}, Q_{d,i}, Q_{c,i}, Q_{init,i}, H_i(Q_{d,i}, Q_{c,i}), E_i, \rightarrow_i)$ 构成 ($i = 1, 2, \dots, n$), 则 $C_{id} = \{C_{id_1}, C_{id_2}, \dots, C_{id_n}\}$ 且 $C_{id_i} \cap C_{id_j} = \emptyset$, $Q_d = \bigotimes_{i=1}^n Q_{d,i}$, $Q_c = \bigotimes_{i=1}^n Q_{c,i}$, $Q_{init} = \bigotimes_{i=1}^n Q_{init,i}$ (\bigotimes 运算表示各个组件状态连接构成的复合组件 C 的状态), $H(Q_d, Q_c) = \bigcup_{i=1}^n H_i(Q_{d,i}, Q_{c,i})$, $E = \bigcup_{i=1}^n E_i$ 且 $E_i \cap E_j = \emptyset$, (其中 $1 \leq i, j \leq n$), “ \rightarrow ” 满足: $q_i, q_i' \in H_i(Q_{d,i}, Q_{c,i})$, $\Gamma_i \in E_i$, $q_i \xrightarrow{\Gamma_i} q_i'$, 组件协同器 $con = ROP\{\Gamma_1, \Gamma_2, \dots, \Gamma_n\}$, 则 $(q_1, q_2, \dots, q_n) \xrightarrow{con} (q_1', q_2', \dots, q_n')$ 。

定义 3 协同器。用于连接组件接口,实现组件间的交互。已知组件 C_1, C_2, \dots, C_n , 协同器根据各组件的状态来生成组件响应事件的方式,用 $con = ROP\{\Gamma_1, \Gamma_2, \dots, \Gamma_n\}$ 表示,其中 $\Gamma_1, \Gamma_2, \dots, \Gamma_n$ 分别为组件 C_1, C_2, \dots, C_n 的事件类型接口,ROP 关系符表示 $\Gamma_1, \Gamma_2, \dots, \Gamma_n$ 间的协同关系。

1.2 基于事件驱动的 CPS 组件协同行为模型

CPS 系统的一个重要特征是计算进程与物理进行的反馈循环和深度融合。本文建立的基于事件驱动的 CPS 组件协同行为模型见图 1。执行器等组件或环境的状态变化可以触发传感器的初始化,从而监测环境或实体。传感器的监测数据触发计算单元对数据加工,并根据监测数据及当前执行器状态生成控制策略。计算单元加工生成的控制事件触发执行器的活动,从而进一步调整实体或环境的状态。系统中的协同器根据传感器、计算单元、执行器的状态协调完成事件响应活动。

传感器组件:传感器组件的连续状态表示监测的物理环境或实体的属性值随时空的变化;事件接口集包括传感器初始化事件接口和数据输出接口。图 1 所示,传感器初始化的事件接口 ϵ_{sen} 由执行器状

态或环境变化触发,数据输出接口由计时器触发。这 2 类事件接口引起相应传感器激活,并输出 ϵ_{CU} ,触发计算单元进行数据处理。

计算单元组件:计算单元组件的连续状态表示不同时刻加工生成的事件;事件接口集用于触发各计算子单元协同执行数据的存储或计算。图 1 所示, ϵ_{CU} 触发计算单元对传感器的监测数据进行处理,并生成 ϵ_{act} 分派到相应执行器,触发相应执行器执行物理操作。

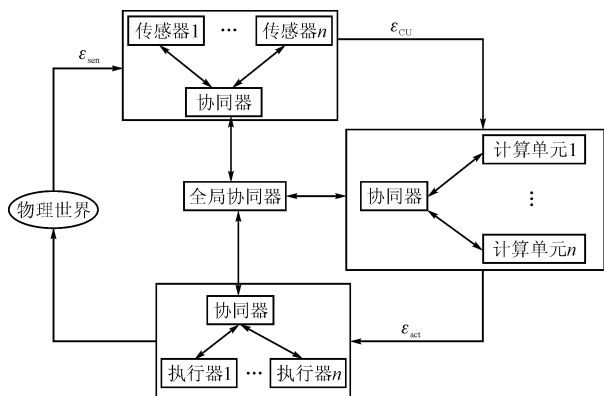


图 1 基于事件驱动的 CPS 组件协同行为模型

Fig. 1 Event-Driven based component coordination model of CPS

执行器组件:执行器组件的连续状态表示物理环境或实体的物理属性值随执行器物理状态和时空属性的变化;事件接口集均由计算单元加工生成的事件触发,驱动执行器活动来改变物理能或形状等。

表 1 事件接口逻辑关系表

Tab. 1 Logic relationships between event interfaces

事件接口逻辑关系	执行语义
并发关系“和”	$\epsilon_1, \epsilon_2, \dots, \epsilon_n$ 的对应组件独立响应事件
强同步关系“且”	$\epsilon_1, \epsilon_2, \dots, \epsilon_n$ 均激活后对应组件方可同步响应事件
条件同步关系“或”	不论 $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ 中有多少个接口被激活,这几个被激活接口的对应组件均可同步响应事件
弱同步关系“因果”	若 ϵ_{i+1} 之前的 $\epsilon_1, \epsilon_2, \dots, \epsilon_i$ 被激活,则 $\epsilon_1, \epsilon_2, \dots, \epsilon_i$ 的对应组件顺序响应事件,其中 $i \in [1, n-1]$
强互斥关系“有且只有”	$\epsilon_1, \epsilon_2, \dots, \epsilon_n$ 均激活后方可选择被激活接口的对应组件中的一个响应事件
条件互斥关系“互斥或”	不论 $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ 中有多少个接口被激活,这几个被激活接口对应的组件中只能有一个响应
弱互斥关系“互斥因果”	若 ϵ_{i+1} 之前的 $\epsilon_1, \epsilon_2, \dots, \epsilon_i$ 均激活,则只能选择 $\epsilon_1, \epsilon_2, \dots, \epsilon_i$ 对应的组件中的一个响应事件, $i \in [1, n-1]$

2.2 组件协同代数系统定义及性质

Simon Bliudze^[8]从组件依赖关系角度提出了连接代数系统。在此基础上,基于事件接口间的依赖和互斥关系,定义了组件协同代数系统,从而为协同器建立了基于事件逻辑的代数模型。

定义 4 组件协同代数系统 $AC(E)$ 。令 E 为组件的事件类型接口集, $0, 1 \notin E$, “0”表示系统处于异常状态,“1”表示系统处于正常状态。定义一元分型操作符“ $[\cdot]^*$ ”,其中 $[\cdot]^0$ 表示汇聚型事件,简

图 1 所示, ϵ_{act} 触发执行器行为并引起执行器状态的变化。当执行器发生状态变化,产生 ϵ_{sen} ,触发相应传感器执行监测活动。

协同器:在该基于事件驱动的 CPS 行为模型中,分别定义了协调传感器、计算单元、执行器等组件活动的协同器,同时定义了全局协同器协调传感器组件协同器、执行器组件协同器、计算单元协同器等组件的协同策略,实现 CPS 系统组件的协调。这种组件协同方式屏蔽了组件间的通信,组件间的交互主要通过协同器实现,有利于实现系统组件的动态调整。协同器的具体协同方案在第 2 节中分析。

2 组件协同代数系统

本节在以事件类型为接口的组件定义基础上,建立组件间基于事件逻辑关系的协同器代数模型。

2.1 事件接口逻辑关系

已知组件 C_1, C_2, \dots, C_n , 组件间通过事件接口 $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ 实现协同操作,其中 ϵ_i 为组件 C_i 的事件接口,则根据组件协同方式定义 7 种事件接口逻辑关系,见表 1。

3 种同步关系能够使得所有同步组件的状态发生迁移。3 种互斥关系中,只有被选择的组件发生状态迁移,其他组件的状态未发生改变。

写为 $[\cdot], [\cdot]^1$ 表示触发型事件^[16]。所谓汇聚型事件,即该事件不能单独被响应,而是与其他事件构成逻辑关系,同步响应或互斥响应。所谓触发型事件,即该事件可独立响应。定义 $U_{AC} = \{[0]^*, [1]^*\} \cup \{[\epsilon]^* \mid \epsilon \in E\}$ 上的二元关系运算符“+”(连接运算符)、“ \cdot ”(同步运算符)、“ Θ ”(互斥运算符),满足性质 1、2、3、4:

性质 1 $\forall x, y, z \in U_{AC}$, 代数运算满足(每个等式中的“ $*$ ”类型操作符同时取“0”或“1”):

交换律: $x+y=y+x$

$$x \cdot y=y \cdot x$$

$$x \Theta y=y \Theta x$$

结合律:

$$[[x]^*+[y]^*]^*+[z]^*=[x]^*+[[y]^*+[z]^*]^*$$

$$[x]^* \cdot [y]^* \cdot [z]^*=[x]^* \cdot [[y]^* \cdot [z]^*]^*=$$

$$[[x]^* \cdot [y]^*]^* \cdot [z]^*$$

$$[x]^* \Theta [y]^* \Theta [z]^*=[x]^* \Theta [[y]^* \Theta [z]^*]^*=$$

$$[[x]^* \Theta [y]^*]^* \Theta [z]^*$$

$$[x]^1 \cdot [y]^1 \cdot [z]^1=[x]^1 \cdot [[y]^1 \cdot [z]^1]^1$$

$$[x]^1 \Theta [y]^1 \Theta [z]^1=[x]^1 \Theta [[y]^1 \Theta [z]^1]^1$$

$$\text{分配律: } x \cdot [y+z]^* = x \cdot [y]^* + x \cdot [z]^*$$

$$x \Theta [y+z]^* = x \Theta [y]^* + x \Theta [z]^*$$

$$x \cdot [y \Theta z]^* = (x \cdot [y]^*) \Theta (x \cdot [z]^*)$$

幂等律: $x+x=x, x \cdot x=x$, 其中 x 为单项式

(所谓单项式, 即由“ \cdot ”和“ Θ ”连接的事件)。

性质 2 ①对于操作符“ $+$ ”, 其单位元为 $[0]$, 无零元; ②对于操作符“ \cdot ”, 单位元是 $[1]$, 零元为 $[0]$; ③对于操作符“ Θ ”, 单位元是 $[0]$, 无零元。

性质 3 $\forall x, y, z \in U_{AC}, \alpha, \beta \in \{0, 1\}$, 分型操作符下的代数运算满足:

$$[0]^1=[0]; [[x]^\alpha]^\beta=[x]^\beta;$$

$$[x+y]^\alpha=[x]^\alpha+[y]^\alpha;$$

$$[x]^1[y]^1=[x]^1+[x]^1[y]^1;$$

$$[x]^1 \Theta [y]^1=[x]^1+[x]^1 \Theta [y]^1;$$

$$[x]^1 \Theta [y]^1=[x]^1+[y]^1+[x]^1[y]^1;$$

$$[x]^1 \Theta [y]^1=[x]^1+[y]^1+[x]^1 \Theta [y]^1;$$

$$[[x]^1[y]^1]^\alpha=[x]^\alpha+[[x]^1[y]^1]^\alpha;$$

$$[[x]^1 \Theta [y]^1]^\alpha=[x]^\alpha+[[x]^1 \Theta [y]^1]^\alpha;$$

性质 4 关系运算符优先级由高到底分别为: “ $[]^*$ ” $>$ “ \cdot ”(“ Θ ”) $>$ “ $+$ ”。

综上, 事件接口间的 7 种逻辑关系使用组件协同代数形式表示见表 2。

表 2 事件逻辑关系及组件协同代数式对应表

Tab. 2 Logic relationships between events and the corresponding algebra expressions of connectors

事件接口逻辑关系	组件协同代数式
并发运算“和”	$[\epsilon_1]+[\epsilon_2]+\dots+[\epsilon_n]$
强同步关系“且”	$[\epsilon_1] \cdot [\epsilon_2] \cdot \dots \cdot [\epsilon_n]$
条件同步关系“或”	$[\epsilon_1]^1 \cdot [\epsilon_2]^1 \cdot \dots \cdot [\epsilon_n]^1$
弱同步关系“因果”	$[\epsilon_1]^1 \cdot [[\epsilon_2]^1 [[\epsilon_3]^1 \cdot \dots \cdot [\epsilon_n]]]$
强互斥关系“当且仅当”	$[\epsilon_1] \Theta [\epsilon_2] \Theta \dots \Theta [\epsilon_n]$
条件互斥关系“互斥或”	$[\epsilon_1]^1 \Theta [\epsilon_2]^1 \Theta \dots \Theta [\epsilon_n]^1$
弱互斥关系“互斥因果”	$[\epsilon_1]^1 \Theta [[\epsilon_2]^1 \Theta [[\epsilon_3]^1 \Theta \dots \Theta [\epsilon_n]]]$

3 实例分析

以智能家居中的非法闯入房屋事件为例, 建立基于事件逻辑的组件协同模型, 实现家居的安全防护^[17]。组件包括门感应器 D_{dr} , 定位传感器 D_{loc1} 、 D_{loc2} 、 D_{loc3} , 电话 D_{tel} , 报警器 D_{alm} , 显示器 D_{ply} , 计算单元 D_{CU} 。图 2 模拟了人进入房间的过程。虚线表示人的活动路线。

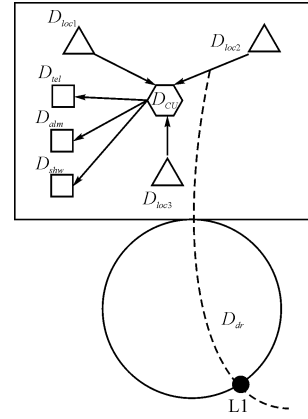


图 2 非法入侵场景

Fig. 2 A scenario of illegal invasion

$$D_{loc1_I_{Init}} = \Gamma_{loc1Init} \{C_{id} = loc1, S_{id} = Timer = J_{id}\}$$

$$D_{loc2_I_{Init}} = \Gamma_{loc2Init} \{C_{id} = loc2, S_{id} = Timer = J_{id}\}$$

$$D_{loc3_I_{Init}} = \Gamma_{loc3Init} \{C_{id} = loc3, S_{id} = Timer = J_{id}\}$$

$$J_{id} = entity, A_{bnear} = TRUE, T_1^o, L_1^o, T_1^g, L_1^g$$

定位传感器实现初始化的事件接口为 $\Gamma_{locInit}$, T_1^o, L_1^o 表示被观察者“门感应器” D_{dr} 感应到电平变化的时间和地点, T_1^g, L_1^g 表示 $\Gamma_{locInit}$ 类型事件(即有实体进入 D_{dr} 感应区域)产生的时间和地点。只有当初始化接口被激活, 才能触发 3 个定位传感器对人物或物的三维空间坐标位置进行监测, 从而避免了传感器时刻采集数据所造成的资源浪费。

$$D_{loc1_I_{Output}} = \Gamma_{loc1Output} \{C_{id} = loc1\}$$

$$D_{loc2_I_{Output}} = \Gamma_{loc2Output} \{C_{id} = loc2\}$$

$$D_{loc3_I_{Output}} = \Gamma_{loc3Output} \{C_{id} = loc3\}$$

$$S_{id} = J_{id} = Timer, A_{bmspan} = T_0, T_2, L_2$$

T_2, L_2 表示计时器 Timer 输出事件 $\Gamma_{locOutput}$ 的时间和地点, 且 T_2 为时钟周期 T_0 的整数倍 $\Gamma_{locOutput}$ 类型事件发生时间和产生时间可认为相同)。当计时器达到输出周期 T_0 时, $\Gamma_{locOutput}$ 触发定位传感器输出检测实体的三维空间坐标数据至缓存区:

三维空间坐标:

$$\begin{cases} \epsilon_1 = \Gamma_{locx} \{S_{id} = loc1, J_{id} = entity, \theta_x, len_x, T'_1\} \\ \epsilon_2 = \Gamma_{locy} \{S_{id} = loc2, J_{id} = entity, \theta_y, len_y, T'_2\}, \text{ 其} \\ \epsilon_3 = \Gamma_{locz} \{S_{id} = loc3, J_{id} = entity, \theta_z, len_z, T'_3\} \end{cases}$$

中 θ , len 表示定位传感器,检测到的实体方法和距离传感器自身的距离; T_i 表示检测时间相对于各定位传感器的时钟刻度,且检测结果的发生时间和产生时间相同。

$$D_{CU_I_{locData}} = \Gamma_{locData} \{C_{id} = CU, S_{kl} = Counter = J_{kl}, A_{nData} \geq N, T_3, L_3\};$$

$$D_{CU_I_{nnlocData}} = \Gamma_{nnlocData}, A_{nData} = 0, T_4, L_4。$$

计算单元执行计算和进入休眠状态的事件接口为 $\Gamma_{locData}$ 和 $\Gamma_{nnlocData}$, 其中 T_3, L_3 为计数器 Counter 检测到缓冲区存放的人或物的空间位置数据达到数量 N 时的时间和地点; T_4, L_4 为计数器 Counter 检测到缓冲区存放的人或物的空间位置数据为空时的时间和地点; 当缓存区空间位置数据总数达到 N 时, $\Gamma_{locData}$ 触发计算单元对坐标数据进行处理, 判断是否进入房间。当缓存区数据类型为空时, $\Gamma_{nnlocData}$ 触发计算单元进入休眠状态。

$$D_{tel_I_{telcall}} = \Gamma_{telcall} \{C_{id} = tel, S_{kl} = CU\}$$

$$D_{alm_I_{almring}} = \Gamma_{almring} \{C_{id} = alm, S_{kl} = CU\}$$

$$D_{ply_I_{plytrace}} = \Gamma_{plytrace} \{C_{id} = ply, S_{kl} = CU\}$$

$$J_{id} = entity, A_{blnrm} = True, A_{loc}, T_5^0, L_5^0, T_5^g, L_5^g。$$

当观察者 D_{CU} 监测到非法入侵事件后, 激活接口 $\Gamma_{telcall}$, $\Gamma_{almring}$ 和 $\Gamma_{plytrace}$, 触发执行器(电话、报警器和显示器)执行电话通知房屋主人或拉响警报的活动, 并使人的活动路线 A_{loc} 在显示器上可视化。 T_5^0, L_5^0 表示计算单元监测到入侵信号的时空信息; T_5^g, L_5^g 表示计算单元生成入侵事件的时空信息。

系统协同器协调定位传感器、计算单元、电话、报警器和显示器, 构成逻辑响应关系:

$$[D_{loc1} - I_{Output} \cdot D_{loc2} - I_{Output} \cdot D_{loc3} - I_{Output}]^1 \cdot [[D_{CU} - I_{locData}]^1 \cdot [[[D_{tel} - I_{telcall}] \Theta [D_{alm} - I_{almring}]]^1 \cdot [D_{ply} - I_{plytrace}]^1]]。$$

3个定位传感器的数据采集构成强同步关系。只有当3个传感器同时成功采集数据后, 计算单元方可执行数据处理。当计算单元判断非法入侵事件发生后, 会触发电话或报警器中的一个来响应事件, 同时将人或物的活动位置在显示器上更新。

这种基于事件逻辑的组件协同模型在实现了异构组件交互的同时, 能够用简洁的代数式表达组件间的所有可能的交互方式。同时建立了全局的组件交互反馈机制, 实现了上下级组件间的紧密衔接, 减少了不必要的资源消耗。

4 结语

本文根据 CPS 的物理过程与信息过程的融合

和反馈循环特性, 定义了具有混成特性的组件形式化表示方法, 建立了基于事件驱动的组件协同体系结构。构造了可以表达 7 种协同关系的组件协同代数系统。该组件协同机制具有鲜明的组件层次结构, 通过定义事件接口来触发组件的状态转移, 能够根据环境和用户需求的变化, 建立动态的组件协同方案。下一步将研究组件协同的一致性、优先级及响应延时性等问题。

参考文献(References):

- [1] MACRO Z, LUCA M. Adaptive Real-Time Communication for Wireless Cyber-Physical Systems [J]. ACM Transaction on Cyber-physical System, 2017, 1(2): 1-8.
- [2] XIAO Y, CHIMAY J A, PARFITT M K. Cyber-Physical Systems for Temporary Structure Monitoring [J]. Automation in Construction, 2016(66): 1-14.
- [3] 景博, 周伟, 黄以锋, 等. 信息物理融合系统及其应用 [J]. 空军工程大学学报(自然科学版), 2014, 15(2): 1-6.
JING B, ZHOU W, HUANG Y F, et al. Reserch of Cyber-physical Systems and Its Application [J]. Journal of Air Force Engineering University: Natural Science Edition, 2014, 15(2): 1-6. (in Chinese)
- [4] MARCIO E F, MAIA, Rossana M C Andrade, Windson V. Towards a Component Infrastructure for Cyber-Physical Systems [J]. Acm Symposium, 2016(4): 626-628.
- [5] FRANCOIS F, OLIVIER B. A Dynamic Component Model for Cyber Physical Systems [J]. International Acm Sigsoft Symposium on Component Based Software Engineering, 2012(6): 135-144.
- [6] 黄万良, 陈松乔. 基于消息的构件组合运算与构件演算 [J]. 小型微型计算机系统, 2007, 7(28): 1216-1220.
HUANG W G, CHEN S Q. Component Composition Operations and Component Evolution Based on Messages [J]. Journal of Chinese Computer Systems, 2007, 7(28): 1216-1220. (in Chinese)
- [7] 赵会群, 孙晶. 面向服务的可信软件体系结构代数模型 [J]. 计算机学报, 2010, 33(5): 890-899.
ZHAO H Q, SUN J. An Algebraic Model of Service Oriented Trustworthy Software Architecture [J]. Chinese Journal of Computers, 2010, 33(5): 890-899. (in Chinese)
- [8] SIMON B, JOSEPH S. The Algebra of Connectors Structuring Interaction in BIP [J]. IEEE Transactions on Computers, 2008, 57(10): 1-16.

- [9] ROBERTO B, IVAN L, UGO M. A Basic Algebra of Stateless Connectors [J]. Theoretical Computer Science, 2006, 366: 98-120.
- [10] ANANDA B, MARIUS B, JOSEPH S. Modeling Heterogeneous Real-time Components in BIP [C]// The Fourth IEEE International Conference on Software Engineering and Formal Methods. IEEE Computer Society Washington DC, USA, 2006: 3-12.
- [11] 刘思中, 曹健. 基于 SOA 的事件协同感知模型 [J]. 计算机工程, 2009, 35(3): 48-50.
LIU S Z, CAO J. Event Collaborative Awareness Model Based on SOA [J]. Computer Engineering, 2009, 35(3): 48-50. (in Chinese)
- [12] 别晓峰, 李为民, 张雅舰, 等. 事件驱动的面向服务作战仿真集成平台架构 [J]. 空军工程大学学报(自然科学版), 2013, 14(2): 37-41.
BIE X F, LI W M, ZHANG Y J, et al. An Event-driven Service-Oriented Architecture of Operation Simulation Integration Platform [J]. Journal of Air Force Engineering University: Natural Science Edition, 2013, 14(2): 37-41. (in Chinese)
- [13] 范贵生, 虞慧群, 陈丽琼, 等. 策略驱动的可靠嵌入式系统建模及分析方法 [J]. 软件学报, 2011, 22(6): 1123-1139.
FAN G S, YU H Q, CHEN L Q, et al. Strategy Driven Modeling and Analysis of Reliable Embedded Systems [J]. Journal of Software, 2011, 22(6): 1123-1139. (in Chinese)
- [14] 尹忠海, 张凯成, 杜华焯, 等. 基于事件驱动的信息物理融合系统建模 [J]. 微电子学与计算机, 2015, 32(12): 126-129.
YIN Z H, Zhang K C, DU H Y, et al. Cyber Physical System Modeling Based on Event Driven [J]. Microelectronics & Computer, 2015, 32(12): 126-129. (in Chinese)
- [15] TAN Y, VURAN M C, GODDARD S, et al. A Concept Lattice-Based Event Model for Cyber-Physical Systems [C]// Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems. ACM New York, NY, USA, 2010: 50-60.
- [16] ZHOU C, YIN Z H, FENG Y Q. Events Algebra of Triggers [J]. The Computer Journal, 2017(23): 1-17.
- [17] 朱振宇, 张仕, 蒋建民, 等. 并发系统中基于优先级的调度分析 [J]. 计算机科学, 2016, 43(11): 523-529.
ZHU Z Y, ZHANG S, JIANG J M, et al. Analyzing Scheduling Based on Priority in Concurrent Systems [J]. Computer Science, 2016, 43(11): 523-529. (in Chinese)

(编辑: 徐敏)