Software Engineering Group
Department of Computer Science
Nanjing University
http://seg.nju.edu.cn

Technical Report No. NJU-SEG-2019-CJ-002

2019-CJ-002

面向事件驱动智能家居物联网系统的 自动化配置、仿真与验证平台

张秋萍，王熙灶，沈思远，张时雨，卜磊，李宣东

物联网学报 Vol.3, No.3, 2019

Most of the papers available from this document appear in print, and the corresponding copyright is held by the publisher. While the papers can be used for personal use, redistribution or reprinting for commercial purposes is prohibited.

面向事件驱动智能家居物联网系统的自动化配置、仿真与验证平台

张秋萍，王熙灶，沈思远，张时雨，卜磊，李宣东
（南京大学软件新技术国家重点实验室，江苏 南京 210023）

摘　要：以 IFTTT 为代表的事件驱动型物联网系统编程框架为用户构建满足其需求的智能家居物联网系统提供了极大的便利，但也带来了严峻的安全隐患。针对此问题，设计并实现了“门神”，这是一个基于模型检验的事件驱动型物联网系统配置、仿真与验证平台。用户可以在门神中自定义其系统，并进行一键式模型驱动的仿真及验证、自动检测并重现错误场景，从而理解系统行为并提升其安全性。通过大量实验可知，门神能在 86.7%的案例中发现安全隐患，且平均耗时仅为 0.7 s。

关键词：物联网；IFTTT 框架；系统安全；模型检验
中图分类号：TP311
文献标识码：A
doi: 10.11959/j.issn.2096−3750.2019.00124

For Research Only

Automated configuration, simulation and verification platform for event-driven home automation IoT system

ZHANG Qiuping, WANG Xizao, SHEN Siyuan, ZHANG Shiyu, BU Lei, LI Xuandong
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

Abstract: The IFTTT style event-driven programming paradigm benefits normal users to build their own customized home automation Internet of things (IoT) system, meanwhile, it also brings serious safety and security risks. To handle this problem, Menshen was designed and implemented, an automated configuration, simulation and verification platform for event-driven home automation IoT system based on model checking. Users can easily set up their own smart home systems in Menshen, and conduct simulation and verification in a push-button style. Menshen could further demonstrate the error trace to help users to understand the behavior of the system and increase the safety and security of the system. An experiment with a large number of cases is carried out, and the results show that 86.7% cases are error-prone, and the verification only took 0.7 seconds in average.

Key words: Internet of things, IFTTT framework, system security, model checking

1　引言

物联网能够将世界更紧密地联系起来，在很多领域都发挥着重要作用，智能家居就是一个典型的物联网应用场景。随着科技的进步与社会的发展，人们更加注重生活的舒适和便利，智能家居的出现满足了人们在居家方式上对更高质量的需求，也因此受到了广泛关注。智能家居允许以“物”影响“物”，家居设备可以通过一定的规则条件互联运转，实现类似于“当检测器发现室外正在下雨，则自动关窗”

收稿日期：2018−11−19；**修回日期：**2019−04−06
通信作者：卜磊，bulei@nju.edu.cn
基金项目：国家重点研发计划基金资助项目（No.2017YFA0700604）；国家自然科学基金资助项目（No.61632015，No.61572249，No.61561146394）
Foundation Items: The National Key R&D Program of China (No.2017YFA0700604), The National Natural Science Foundation of China (No.61632015, No.61572249, No.61561146394)

之类的功能，从而给生活带来了极大的便利。

随着技术的发展，以 IFTTT（if this then that）框架为代表的用户为主体的事件驱动式编程（trigger-action programming）[1]方式被提出。通过相关框架，用户可以编写类似 IFTTT 型的驱动命令，使得多个设备间能进行智能互联。IFTTT 框架的出现使得普通用户不再仅仅是使用者，还能够直接参与智能家居系统的设计，用户可以通过设计 IFTTT 命令将不同设备关联起来，使得家居系统能够智能地运转。但是，在这样一套系统的背后，隐藏着巨大的风险。对于单个互联规则的行为，用户一般是可以控制的，但当互联规则的数量增多时，系统的复杂度快速增加，若系统设计不当，很容易造成一些普通用户无法察觉的安全问题。使问题变得更加严重的因素在于绝大多数普通用户并没有安全概念，他们会在系统配置完成后直接使用，这样的做法存在极大的安全隐患。

显然，帮助用户对系统进行仿真运行，告知用户其定制系统的可能行为，对其理解系统、发现问题具有直接作用。因此，本文设计并开发了一套智能家居配置与仿真平台——门神。在门神中，定义了设备描述格式，并对主流类型设备进行了预置。用户可以自由图形化配置系统中的设备及互联规则，门神可按照用户设定的规则随机模拟仿真系统行为，使得用户能够了解自定义家居系统可能的行为。

但是，仿真仅能帮助用户理解系统行为，发现问题还需要依靠用户自身，用户并没有意识到隐藏较深的安全问题。智能家居系统的安全性与用户的财产及生命安全密切相关，因此，设计一种更加严格的检验手段是非常必要的。目前，已有相关工作提出使用形式化验证方法来分析系统的安全性[2-6]，但是普通用户并不具备使用形式化验证方法的能力，针对该问题，提出将形式化验证流程自动化，以方便用户使用。因此，本文对门神进一步扩展，实现了自动化建模、验证以及相关错误的修复功能，提供了一个面向普通用户的检验平台，以保证用户自定义系统的安全性。

总体而言，本文主要贡献如下：

1) 定义了一套标准的设备信息记录格式，以 JSON 格式文档存储信息，支持大多数主流类型设备，并已为其生成信息文档。

2) 基于设备信息文档，设计了一套针对用户自定义的基于事件触发的家居系统自动建模与规约生成方法，建模中考虑了多设备对环境变量的影响，模拟环境的动态变化，同时考虑了设备事件的可信赖度和信息的隐私度，检查系统是否存在隐患。

3) 实现了首个面向用户的智能家居系统仿真与安全验证平台，为用户提供了可视化界面，用户根据需求选择智能设备，设定 IFTTT 规则，定义自己的智能家居系统，平台底层自动为家居系统建模，根据用户需求进行仿真或验证，并将仿真和验证结果以动画演示形式展示给用户，使用户了解其定义的系统可能存在的安全隐患。

2　背景与动机

2.1　IFTTT 框架

IFTTT 型事件驱动式控制框架由 IFTTT 网站于 2010 年提出，一上线便引起业界与广大用户的密切关注。IFTTT 对各种设备的应用程序编程接口（API，application programming interface）进行适配，由后台完成各种 API 的调用和响应，而在前端，用户只需要使用 IFTTT 框架，以“this”为条件触发器，选择相应的设备及其需要满足的条件，以“that”为条件响应动作，选择响应设备及其需要执行的动作事件，就可以根据需求制定 IFTTT 规则。IFTTT 平台就会在后端对用户制定的规则进行处理，一旦“this”触发器满足条件被触发，相应的指令执行设备的 API 就会被调用，执行“that”动作。用户可以用这种简单的方式进行定制化“编程”，实现智能家居系统的个性化配置，而不需要理解后端各种 API 的处理情况，这让普通用户可以自由地操控和连接各种设备。

2.2　物联网系统安全问题与挑战

用户利用 IFTTT 框架可以简便地将各种智能设备关联起来，使得互连的设备能够按照一定条件智能地运转，为用户提供便利和舒适。但是这种便利的背后往往也伴随着风险和隐患。

智能设备根据用户的主观想法被连接起来，用户定义的规则具有不确定性，如果规则设计不当，就可能引发安全问题。而且设备和设备互相连接、互相影响，在一定程度上可以说设备是自动运转的，那么安全问题就可能在用户注意不到的时候发生，造成无法挽回的后果。

用户根据需求定义规则，往往意识不到自己定义的规则可能引发安全问题，举例如下。

For Research Only

规则 1 if temperature < 26, then turn off air conditioner；

规则 2 if air quality is bad and air conditioner is off, then open window。

结合规则 1 和规则 2，当温度低于 26℃且室内空气质量差时，窗户就会自动打开。在这个例子中，用户将“开窗户”这一事件的控制权交给环境变量是十分危险的，窗户很可能在用户不知情的时候被打开，给偷盗者恶意闯入的机会。

类似的场景在用户自定义网络中屡见不鲜。显然，在用户认识不到且无手段去处理相关问题时，对此类用户定义的 IFTTT 型事件驱动物联网系统进行安全检验具有重要意义。

2.3 模型检验

模型检验[7]是一种重要的自动验证技术。对于一个给定系统，为其建立形式化模型，然后根据描述系统性质的规约，自动地详细检查模型是否满足规约。

随着技术的进步，目前相关领域积累了一批成熟的模型检验工具，如 Spin[8]、NuSMV[9]、BACH[10]等，可以对相关系统的正确性进行有效判断，相关技术与工具已经在软/硬件设计中得到广泛应用。

在智能家居领域，目前已经有多项工作认识到其背后的安全隐患，并提出使用模型检验方法验证智能家居系统行为的正确性[4-6]。相关工作普遍使用有限状态机（FSM，finite state machine）对家居系统行为建立模型，使用规约定义家居系统的正确行为，从而验证家居系统行为是否符合期望。然而，相关工作普遍假设上述内容由用户自主完成，显然，这个假设远远超出了普通用户的能力范畴。针对此问题，本文设计并实现了一个工具平台，能够对相关的建模、规约、验证等环节实现自动化并应用于普通用户层面，从而保证系统的安全性。

3 面向事件驱动智能家居物联网系统的形式化模型自动生成

针对智能家居中可能存在的安全隐患，本文设计并实现了一个面向 IFTTT 框架的物联网系统安全仿真和验证平台——门神，通过随机仿真帮助用户理解其定制的智能家居系统行为，并对其安全性进行自动化模型检验。门神平台框架如图 1 所示。

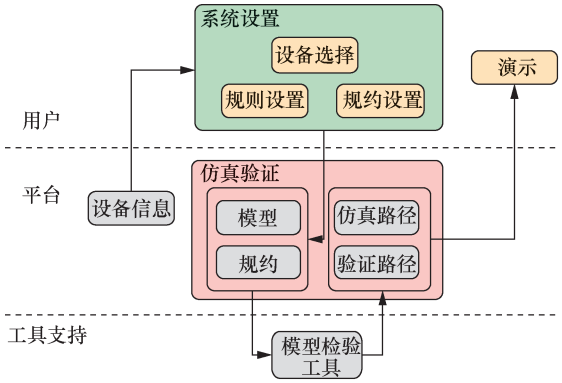


图 1 门神平台框架

门神内置了主流智能设备信息，用户可以从设备列表中选择设备、设定规则、设置安全规约，平台根据用户对系统的设置信息对系统行为进行自动建模，除了将用户设定的规约形式化，还设计了一套可自动生成的基础安全相关规约供用户选择。然后平台使用模型检验工具进行仿真和验证，仿真和验证结果则以一种直观的方式展示给用户，以帮助用户理解系统行为。

门神系统中的仿真与安全验证都基于系统模型进行，因此，相关系统的形式化模型自动构建是其中重要一环。所以，本节的重点是如何将用户自定义的系统信息转换为形式化模型，即如何对家居系统进行建模以及如何生成安全性检验的规约，然后输入模型检验工具中进行验证。

3.1 设备信息文档的定义

若进行基于模型的仿真与验证，首先要对智能家居系统建立模型，由于整个系统行为都基于智能设备，因此，要进行自动化建模，一套标准的、可自动解析的设备信息格式将会带来极大便利。目前，一些商业化服务提供平台如 Google Weave、Amazon AWS、SmartThings 等，都提出了各自的设备信息标准文件格式来进行交互与处理。类似地，针对模型检验需求，定义了所需要的设备信息格式，内容包括如下 6 个方面。

1) 设备名：描述了设备的类型名称；

2) 设备描述：描述了设备的功能等信息；

3) 属性集：描述了设备探测的变量或设备可影响的变量；

4) 状态集：描述了设备可能处于的工作状态及其在对应状态下对变量的影响率；

5) 内部迁移集：描述了设备内部动作及其可能导致状态和属性的变化；

For Research Only

```
{
  "Name": "AC Cooler",
  "Description": "cool ingfunction",
  "Variables": [
    {
      "Name": "temperature",
      "Internal": false
    }
  ],
  "InitState": "Off",
  "WorkingStates": [
    {
      "Name": "Off",
      "Dynamics": [
        {
          "VariableName": "temperature",
          "ChangeRate": " 0"
        }
      ],
      "Invariant": "true",
      "Description": "The equipment is closed",
      "Trust": "trusted"
    },
    {
      "Name": "Working",
      "Dynamics": [
        {
          "VariableName": "temperature",
          "ChangeRate": " -1"
        }
      ],
      "Invariant": "true",
      "Description": "The equipment is working",
      "Trust": "trusted"
    }
  ],
  "Transitions": [],
  "APIs": [
    {
      "Name": "Turn_On",
      "StartState": "Off",
      "EndState": "Working",
      "Trigger": null,
      "Assignments": [],
      "Signal": true,
      "Description": ""
    },
    {
      "Name": "Turn_Off",
      "StartState": "Working",
      "EndState": "Off",
      "Trigger": null,
      "Assignments": [],
      "Signal": true,
      "Description": ""
    }
  ]
}
```


图 2　制冷空调 JSON 文档内容示例

6) 用户可执行操作集：描述了用户可以执行的动作及其可能导致状态和属性的变化。

将这些信息以 JSON[11]格式进行整合，形成设备的信息文档。制冷空调 JSON 文档内容示例如图 2 所示，包含 Name、Description、Variables、InitState、WorkingStates、Transitions 和 API，分别对应设备名、设备描述、初始属性、属性集、状态集、内部迁移集以及用户可执行操作集。此系统中，空调有 Off 和 Working 两个模式，可执行开空调（Turn_On）和关空调（Turn_Off）两个动作，导致空调模式在 Off 和 Working 之间转换，且空调在 Working 模式下会影响温度，使温度降低。

基于上述标准格式，目前门神内置的设备文档类型能够支持家居系统中 45 种主流智能设备和服务，包括 12 种传感器、22 种执行设备以及 11 种网络服务，如运动传感器、温度传感器、门、窗、摄像头、微博和定位服务等。

3.2　设备基础模型生成方法

在构建设备信息文档后，门神可以对设备文档进行解析。设备文档中各类信息可与有限自动机各类内容一一对应，进而可自动化生成对应设备的 FSM 模型[5-6]：

1) FSM 名称与文档中设备名对应；

2) FSM 属性与文档中自身属性集对应；

3) FSM 各个状态与文档中状态集对应；

4) FSM 迁移与文档中内部迁移集和用户可执行操作集对应。对于会与外界进行交流的动作迁移，为其生成一个信号（Signal），并在 FSM 对应迁移上加上将该 Signal 置为 true 的重置动作。

对 2.2 节中的规则 1 及相关设备进行建模得到的 FSM 模型如图 3 所示。

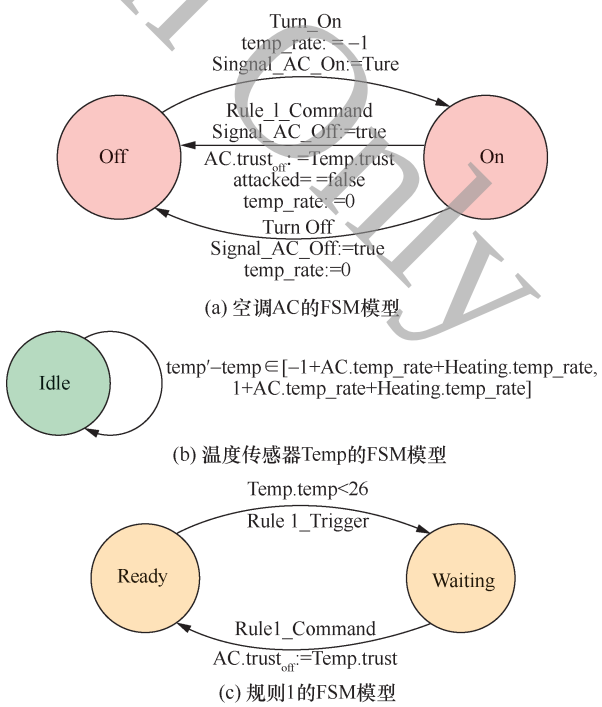


图 3　对 2.2 节中的规则 1 及相关设备进行建模得到的 FSM 模型

For Research Only

对相关系统进行建模的一个主要难题是对环境变量的建模。环境变量在很大程度上受物理环境的影响，且影响因素十分复杂，如家居系统外的环境、周围的家居设备，甚至是不同的物理媒介都对环境变量有不同程度的影响。对于不同的智能家居系统，其环境是不同且不确定的，而基于仅有的设备基础行为信息和用户定义的规则信息难以对其进行精确的自动化建模。

因此，本文基于可获得的系统信息，通过考虑环境变量在目标系统中受到的多种设备对其的交叉影响，来尽可能地模拟环境的变化。由于环境是动态变化的，而且会受到各种设备的影响，如空调和暖气都可以导致温度变化，因此，在建模中，考虑环境和设备的关系以及多种设备对环境变量的多重影响，以尽可能地模拟环境的变化。

在设备信息中记录了设备可影响的变量及其影响方式，基于设备信息文档，可将多种设备对环境变量的多重影响在建模中进行扩展。

1) 基于设备信息文档中所描述的设备对若干环境变量的影响，在设备 FSM 中设定其对环境变量的影响效果。设备在不同状态下对环境变量的影响效果不同，因此，在状态迁移中基于目的状态重置其对环境变量的影响值，空调 AC 的 FSM 模型如图 3(a)所示，图 3(a)中设置了空调在不同状态下对温度的影响值 temp_rate。

2) 探测环境变量的传感器 FSM，综合各种设备对其的影响，在原本的变化率上叠加多个设备对其的影响变量。

温度传感器 Temp 的 FSM 模型如图 3(b)所示，图 3(b)描述了当空调与取暖器同时影响温度时，对环境温度的整体建模方法。由此，可以对整个系统中各种不同设备的行为进行基础建模。

3.3 关联规则下的系统模型生成方法

在相互关联系统中，设备间的关联规则是将各个系统进行关联的核心机制。在 IFTTT 规则中，“this”触发器涉及设备及其属性或状态条件、动作信号，而“that”响应指令则涉及设备及其动作命令，针对这些信息变量设计了规则的记录格式，用户通过选择触发器和响应指令即可完成对相关信息变量规则的设定。

规则将不同设备关联起来，因此，它们之间的关联关系也需要在模型上体现。首先，为每条规则生成一个 FSM，其中，规则 FSM 接收规则中相关设备 FSM 的变量、状态信息或 Signal，若变量或状态满足条件或 Signal 为 true，则进行状态迁移，指令相关设备模型也根据该规则添加状态迁移，迁移标签和规则模型相同，然后通过该共享标签规则模型就可以通知动作指令相关设备 FSM 进行相应的状态迁移。以 2.2 节中的“规则 1：if temperature<26, then turn off air conditioner”为例，规则 1 的 FSM 模型如图 3(c)所示，此模型判断温度是否满足小于 26℃这一条件，若满足条件则触发 Rule1_Trigger 迁移，紧接着触发 Rule1_Command 迁移，图 3(a)的模型有与其共享标签的迁移，则同时触发空调模型的 Rule1_Command 迁移，空调从 On 状态转换到 Off 状态。

3.4 设备事件的可信赖度与信息的隐私度建模

模型可以模拟包含了用户定义规则的整个家居系统的基本行为。而在智能家居中，还有一些非显式信息能够体现设备性质和系统状态，如设备事件的可信赖度和信息的隐私度都可能影响系统的安全性。

物联网将不同设备进行连接，一些设备的动作事件可能在用户不注意时发生，此时，不希望安全相关的重要事件被非高信任度事件触发。因此，定义了设备事件的可信赖度，用来描述事件可由谁引发。如温度的升降是不可控的，对于家庭系统来说，这是非高信任度事件；又如屋外的灯可由路人操控，是不可信赖的；开门、开窗、开空调等事件只能由屋内用户执行，因此是可信赖的。

在 FSM 中为设备事件定义了一个 trust 变量，表示事件的可信赖度。将传感器相关事件和屋外设备相关事件初始化为 untrusted，其他可信赖事件则初始化为 trusted。事件 trust 值会随着规则从条件事件传递给动作事件，同样以 2.2 节中的“规则 1：if temperature<26, then turn off air conditioner”为例，关空调事件是由温度低于 26℃事件导致的，也就是由非高信任度的 untrusted 事件导致的。如图 3(c)所示，规则模型中，空调被关的 trust 值被置为温度传感器的 trust 值，改变了原值，变为 untrusted。显然，不希望安全相关事件由非高信任度事件触发。

同时，也不希望隐私信息被泄露。因此，类似地定义了设备信息的隐私度，用来描述信息可对谁分享。为设备中包含的信息定义一个 privacy 变量，如用户上传到社交网络中的内容初始化为 public，

For Research Only

相机拍摄照片等初始化为 private，同样地，privacy 变量随规则传递。如拍摄照片并上传到社交网络，此时社交网络内容隐私度变为 private，说明社交网络中上传了隐私信息。

结合上述各个环节，图 3 为一个互联智能家居系统中的部分环节。以 2.2 节中的规则 1 为例，生成了各个设备和规则的具体 FSM 模型，包含温度传感器模型、空调模型以及规则模型，规则模型与温度传感器模型和空调模型进行交互，完成从规则条件到规则动作的影响操作。

4　规约置入和基于模型的仿真与验证

4.1　规约生成方法

在第 3 节中，定义了一套对智能家居系统进行自动建模的方法。除了系统模型，在模型检验的输入中，需要规约来定义系统的正确行为。时序逻辑是模型检验中广泛使用的规约形式，典型的时序逻辑有 CTL[12]和 LTL[13]等。门神中将使用这两种规约定义系统安全性需求，并输入模型检验工具中来描述对系统行为的期望。显然，在建模超出普通用户能力的情况下，提供严格时序逻辑这一任务更是超出了用户能完成的任务范畴，因此，自动生成规约是一项必须要做的工作。针对此需求，门神提供了两种解决方案：1) 提供模板，用户填充期望的系统行为，平台自动转换为形式化规约；2) 内置规约，针对智能设备自动生成。

第一种方案设计了 6 类可阅读性较强的规约模板：

1) *E* holds forever；

2) *E* will happen later；

3) *E* never happens；

4) If *E*1 happens, *E*2 should happen at the same time；

5) If *E*1 happens, *E*2 should happen later；

6) If *E*1 happens, *E*2 should happen later and last forever。

其中，*E*、*E*1 和 *E*2 由用户填充事件，完成规约设置，然后平台进行自动转换，生成对应的 CTL 或 LTL 形式化规约：

1) CTLSPEC AG *E*；

2) CTLSPEC AF *E*；

3) CTLSPEC AG !*E*；

4) CTLSPEC AG(*E*1→AX *E*2)；

5) CTLSPEC AG(*E*1→AF *E*2)；

6) LTLSPEC G(*E*1→FG *E*2)。

如用户希望屋内一氧化碳浓度一直保持在一个阈值内，就可以选择模板一，设置“CO<40 holds forever”，然后平台自动转换为形式化规约：CTLSPEC AG CO<40。

依据模板填充用户期望的系统行为，可以降低完成形式化规约的难度，但依然需要用户干预，参与到验证中。如前文所述，一些用户对可能存在的安全隐患并没有概念，也不会意识到应该定义怎样的安全性规约，而且以上模板中定义的规约比较具体，并不能很好地涵盖某一类问题。

因此，针对可信赖度和隐私度问题，门神实现了第二种方案，即设计一套规约格式并内嵌到平台中，为相关设备及其事件自动生成规约：

1) CTLSPEC

AG !(dev.state=st&dev.trust$_{st}$=untrusted)；

2) CTLSPEC

AG !(dev.state=st&dev.privacy$_{st}$=private)。

规约一描述设备 dev 不会因非高信任度事件触发而到达状态 st；规约二则描述设备 dev 不会发布隐私信息。

门神中内置了安全攸关设备及其动作事件列表。当用户制定的规则涉及内置的安全攸关设备及其动作事件时，平台就会使用以上规约格式为其自动生成具体的安全性规约。对于规约一，定义了一些重要事件，如开门、开窗、开空调、发微博、转账等，这些事件如果可由非高信任度事件触发，就可能导致恶意闯入、资源浪费、隐私泄露以及经济损失等，如果规则中涉及这类事件，对应的规约将自动生成。对于规约二，微博是一个典型示例。显然，用户不希望微博将隐私数据发布到网络上，因此，当用户定义的规则涉及发微博事件，平台将为发微博事件自动生成对应规约：CTLSPEC AG !(Weibo.state=Post&Weibo.privacy$_{Post}$=private)，以验证微博不会发布隐私信息。当然，这两类规约也允许用户自己定义安全攸关事件，填充到这两类规约中，平台为其生成形式化规约。

4.2　仿真验证与场景展示

门神将以上自动生成的系统模型与规约输入现成的模型检验工具中，就可进行系统仿真或安全验证。仿真的输出结果是一条状态路径 Trace，验证结果若为错误，则会输出一个反例的状态路径，

For Research Only

路径中包含若干个状态节点 StateNode，节点数量即路径长路，每个状态节点包含了对比上个状态节点由于规则或自发动作发生变化的设备状态、属性或其他变量的值，形式为 Var=Value，表示 Var 在该状态节点被修改为 Value，若某一变量在该状态节点被赋值，则说明它在该节点没有发生改变，和在上一状态节点的值一样。

仿真功能中，门神以系统模型和用户设定的步数（step）为输入，调用成熟的仿真工具，基于该模型语义随机输出一条状态节点数为 step 的状态路径；而模型检验功能则以系统模型和规约为输入，调用成熟的模型检验工具验证系统模型是否满足规约，若有规约没通过验证，工具就会输出一条反例路径。

让用户自己阅读验证工具输出的路径结果是不合理的，因此，门神将对结果路径进行解析，分析事件的变化，并将其以动画演示的形式在可视化界面上显示，更直观地展示仿真和验证结果。

记录路径上各状态节点中设备状态、属性或其他值的变化情况，并对应到各个模型上，如在某一状态节点有 air_conditioner.state=On，则表示 air_conditioner 模型在该节点的状态变更为 On。记录每个状态节点中设备的状态、属性变化以及规则的触发事件，并以动画演示的形式在可视化界面的设备或规则上显示，将仿真或验证结果 Trace 上的变化逐步展示给用户，使结果显示更直观、更易理解。

5 门神的实现与展示

前文介绍了如何将用户层获得的系统信息转换为形式化模型，从而可以使用模型检验工具进行安全验证。进一步将建模和规约生成方法封装起来，同时，以用户为主体设计了一个家居安全验证平台——门神。用户使用门神完成家居系统设置后即可一键式检查系统的安全性，本节将对门神的实现进行系统化展示与阐述。

5.1 系统概述

门神的主要功能包括：智能家居系统的设置与管理，用户可以进行设备选择、添加规则以及定义安全性规约等操作；系统仿真与安全性验证涉及对系统进行自动化建模并自动生成规约，然后调用模型检验工具得到仿真路径或验证反例路径，并以动画演示结果。

门神的主要特点包括：第一个面向用户、支持用户自主进行系统仿真和安全验证的工具平台；提供图形化交互界面，操作简单，结果展示可视化；支持大多数主流设备，对于市面上的常见设备都有典型案例。

门神使用 Java 语言实现，能够在多个平台上运行，其背后调用的形式化验证工具为 NuSMV[9]。门神在为用户自定义的系统自动建立模型后，根据用户需求或内置规约自动生成安全性规约，然后使用 NuSMV 工具对系统进行仿真或安全性验证，同时将仿真或验证结果以直观的动画演示形式反馈给用户。

5.2 系统结构

门神系统工作流程如图 4 所示。用户首先为智能家居系统进行设置，包括添加设备和设置规则；其次，平台根据系统设置进行建模，同时基于内置格式自动生成相关的安全性规约，用户也可以根据模板自定义安全需求，平台转换成形式化规约，然后系统模型和规约都输出到模型文件中。之后，用户可以进行功能选择：仿真或验证。用户可以设置步长 k 对模型进行仿真，NuSMV 工具处理模型文件获得长度为 k 的仿真路径，将路径结果在可视化界面上进行动画演示；用户也可以选择模型验证，使用 NuSMV 工具验证规约是否满足，若违反规约，则获得反例路径，将反例在可视化界面上进行动画演示。

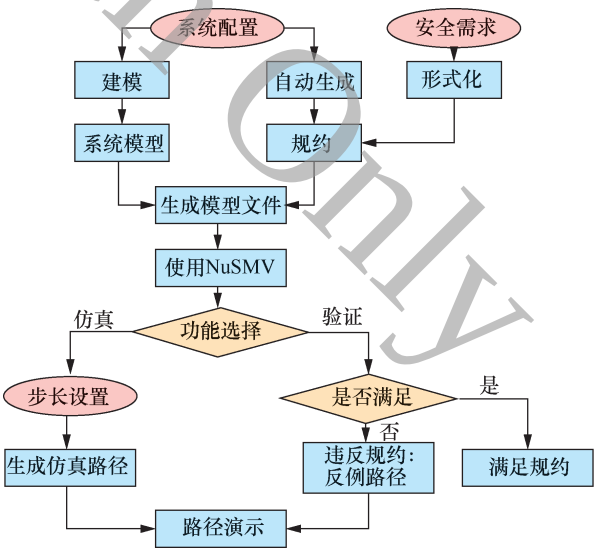


图 4 门神系统工作流程

门神系统主要包/类如图 5 所示，具体如下。

1) IFTTT-IoT-Verifier.common.data 包主要定义了设备基础信息的数据结构以及记录用户选择的设备和添加的规则、规约的数据结构。

For Research Only

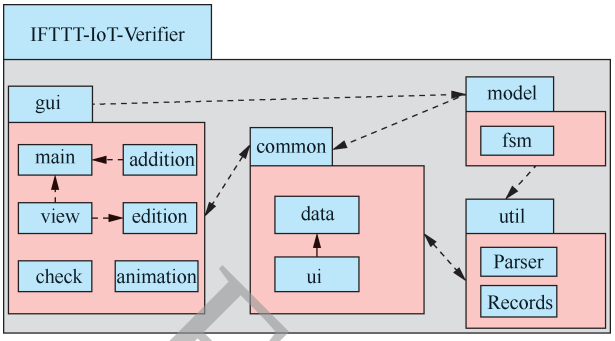


图 5　门神系统主要包/类

2) IFTTT-IoT-Verifier.common.ui 包主要是处理界面信息以及一些通用的 UI 操作。

3) IFTTT-IoT-Verifier.gui 包主要是实现可视化处理界面，包括设备展列、系统设置、仿真、验证和演示等界面。

4) IFTTT-IoT-Verifier.util 包主要是工具包，如 JSON 文件处理、系统设置的保存等。

5) IFTTT-IoT-Verifier.model.fsm 包主要是根据设备、规则和规约生成 SMV 模型文件，然后调用 NuSMV 工具进行仿真或验证，获得相应结果路径。

5.3　主要功能展示

5.3.1　系统设置管理功能展示

如 3.1 节所述，设备信息以 JSON 格式文件存储。首先需要将各种设备文档导入平台，利用 Gson 库解析 JSON 文件，转换为对应的设备对象，放入设备列表中。平台主要界面如图 6 所示，主界面如图 6(a)所示，界面左边一栏展示了可选的设备列表，有文字名称和图标两种展示方案。

(a) 主界面　(b) 设备信息

(c) 规则添加　(d) 规则查看和编辑

(e) 规约添加　(f) 规约查看和编辑

图 6　平台主要界面

For Research Only

以 2.2 节的两条规则为例，进行设备和规则的添加。用户可以从左边列表拖拽设备名称或图标到右边的系统场景界面中，对应设备的图标就会添加到鼠标松开的位置。设备信息如图 6(b)所示，用户可以查看设备的相关信息，还可以移动设备位置、双击鼠标以删除设备等。

规则添加如图 6(c)所示，用户可以选择设备及其相关条件或动作，从而添加规则；规则查看和编辑如图 6(d)所示，用户可以在图 6(d)的界面中查看、编辑或删除规则；规约添加如图 6(e)所示，用户可以选择规约模板，然后选择设备及其相关信息添加规约；同时用户可以在图 6(f)的界面中查看根据用户定义的需求生成的规约以及平台根据用户设置的系统信息自动生成的内嵌规约，并进行编辑或删除。如 4.1 节所述，平台匹配到 Window 这一设备，将为其自动生成如下规约：

CTLSPEC AG !(Window.state=Open &Window.$trust_{Open}$=untrusted)，该规约表示开窗这一安全相关事件不能由非高信任度事件触发。

在完成所有设备添加与配置后，待分析的整体系统场景界面如图 7 所示。

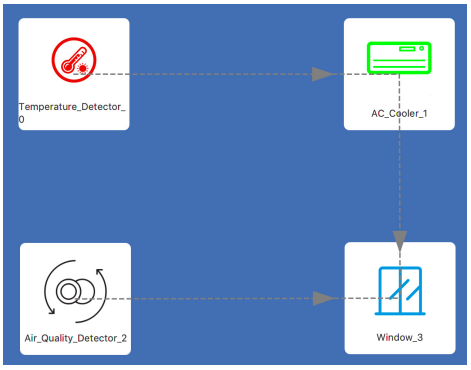


图 7 待分析的整体系统场景界面

5.3.2 仿真验证功能展示

用户可以在主界面选择对设定的家居系统进行仿真或安全验证，对结果路径可选择进行动画演示。

1) 仿真功能：平台首先根据用户设定的系统信息自动生成模型文件，并以仿真执行步长作为输入，然后调用 NuSMV 工具进行随机仿真，得到工具输出的执行路径。

2) 验证功能：平台首先生成模型文件，然后调用 NuSMV 工具进行模型检验，若规约验证结果显示系统违反规约，则可得到工具输出的反例执行路径。

门神对 NuSMV 输出的路径进行处理，将状态节点上的信息对应到相应的设备或规则上，然后进行动画演示。在演示中，高亮发生变化的设备，并弹出信息框，标红状态或属性的变化，同时高亮被触发的关联规则，发生状态改变的设备将更换状态对应的图标，使用户能够更直观地观察系统状态的变化。反例路径演示如图 8 所示，空调处于关状态且空气质量发生变化达到阈值，触发了规则 2，则规则连线高亮，空气质量的 CO_2 属性值的变化也被标红。用户规约违反提示如图 9 所示，高亮了相关设备窗户，窗户图标变更为开窗图标。

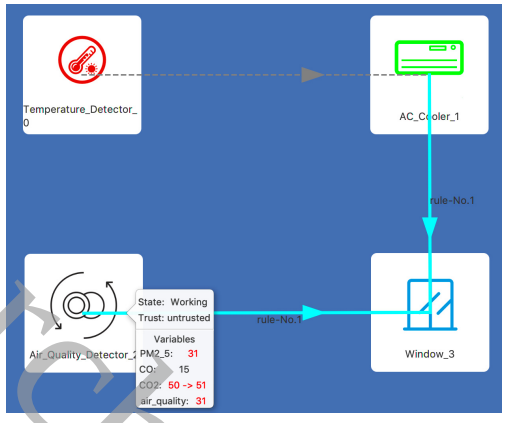


图 8 反例路径演示

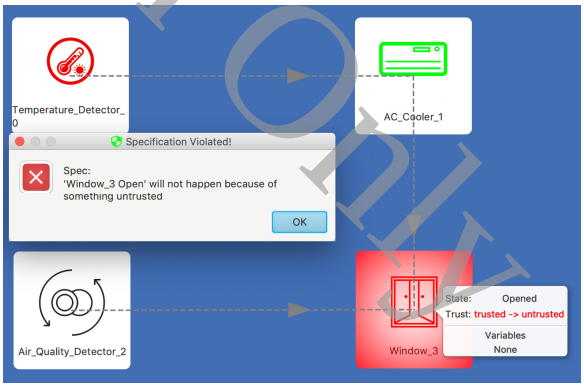


图 9 用户规约违反提示

6 实验

为了评估门神的效率与可扩展性，进行了一系列调研实验，邀请了大量实际用户设计自己的智能家居系统，然后使用工具对其进行自动化的系统建模与安全验证。本节将介绍实验内容和验证发现的问题，并评估平台工具的效率。

For Research Only

6.1　实验设计

本文希望从用户案例中分析现实生活中可能真实存在的问题，因此，在调研实验中，邀请参与者来设计自己的智能家居系统。实验为参与者提供了 45 种常见的物联网设备或服务，包括 12 种传感器、22 种执行设备以及 11 种服务应用，如温/湿度传感器、运动检测器、窗户、空调、微博等。参与者可以使用这些设备或服务，按照 IFTTT 框架编写若干规则，设计一个属于自己的智能家居系统。需要说明的是，一种设备或服务可存在多个实体，如卧室和客厅的灯。

实验收集了来自 91 位参与者设计的案例，每份案例包含的规则为 5～16 条。同时，考虑一个家庭中有多个用户共同设计智能家居系统的情况，随机挑选 2~3 个案例组合构成新的案例集，共获得 59 份新的组合案例集。因此，总共有 150 份家居系统信息，平均每份案例包含 12 条规则，涉及 17 个设备或服务，单份案例最多包含 29 条规则，涉及 50 个设备或服务。

在相关工作的实验评估中，系统场景数目均为几十个左右[4-5]，获取的案例数可以提供比较充分的评估研究数据。在这 150 个系统场景中，使用门神自动化系统进行相关评估，分别对这 150 份案例进行自动化建模，使用系统内置规约，并进行验证。

6.2　实验评估

由于相关的智能家居系统安全研究工作大多面向不同层面，而且很多并不开源，无法获取源代码，因此，本实验评估仅针对本文开发的系统，对其发现的问题和效率进行分析。

在获得的 150 份案例中，有 147 份案例涉及设定的设备事件（开门、开窗、开空调、发微博、转账等）。这 147 份案例中，88 份案例来自参与者设计的案例集，另外 59 份案例是组合案例集。门神为这些案例集自动生成相应的规约，最终实验中平均每份案例包含 3.64 条自动生成的安全验证规约。

实验数据如表 1 所示。数据显示，在所有案例中，有 130 份案例出现违反规约的情况，即相关案例集中有 88.4%（130/147）存在隐患。在单用户案例集中，有 84.1%（74/88）无法通过安全验证；在组合案例集中，有 94.9%（56/59）无法通过安全验证。从总体规约数量来看，在每份案例平均包含的 3.64 条规约中，平均未通过规约数量为 2.09 条，即 57.4%（2.09/3.64）的规约都无法满足。

这说明在用户自定义的规则中，设计不当的情况广泛存在，在多用户参与的系统中，这种情况更加显著。除了单个用户设计不当造成的隐患，在多用户系统中，即使两个用户各自定义了两条没有问题的规则，但它们关联起来却可能出现问题，从而造成严重后果。本文的工具平台则能自动化检查智能家居系统中可能存在的安全隐患，帮助用户了解自己定义的规则可能造成的后果。

6.3　典型问题场景

人工检查了存在规约违反情况的案例，通过综合分析，发现了一些典型问题规则。

1) if temperature>28, then turn on the air conditioner

此例子中，环境温度高于某个阈值这样的非高信任度事件能够触发开空调事件，可能造成用户不在的时候空调被自动打开，从而产生能源浪费。

2) if CO>50, then trigger the alarm

3) if alarm is triggered, then open the door

此例子中，如果一氧化碳浓度设置不当，就可能导致用户不在家时，门会被大概率地自动打开，从而被怀有恶意的人利用进行入室抢劫等。

4) if camera takes a photo, then post it to Weibo

此例子中，用户希望把照片上传到微博分享日常生活，但是有时候用户可能忽视了这条规则，而让一些隐私照片也自动上传到社交网络中，造成了隐私泄露。

用户在定义规则时可能并没有想到这些规则所引发的问题，而门神则可以利用模型检验的方法来检查用户定义的规则是否会导致不安全的后果，帮助用户意识到系统中的隐患。

表 1　　**实验数据**

	案例数量	需验证案例数量	验证未通过案例数量	平均包含规约数量	平均未通过规约数量	平均验证时间/s
单用户	91	88	74	2.70	1.60	0.55
多用户	59	59	56	5.08	2.83	0.94
总体	150	147	130	3.64	2.09	0.7

For Research Only

6.4 效率和可扩展性

记录了每份案例的建模和安全验证时间，从实验数据中可以发现，对于这 150 份智能家居系统案例，工具基本能在 1 s 左右完成系统建模和安全验证，最长时间也只需要不到 10 s，平均处理时间为 0.7 s。从实验结果来看，无论是面对单用户智能家居系统，还是组合多用户系统，本文设计的工具都能高效地处理其安全验证问题。

将案例集中设备和规则数量之和记为该案例集的规模，以案例集规模为横轴、验证时间为竖轴绘制了规模和时间的散点，规模和时间的散点如图 10 所示。实验涉及各种规模案例集，包括从 11 个节点的小规模系统到多至 79 个节点的大规模系统，从图 10 中可以发现，本文设计的工具对不同规模的系统适应性很好，相较于小规模系统，较大规模系统的平均验证时间只是略有提升。而对于建模较复杂、模型嵌套较多的案例，也能在 10 s 内处理完成。实际的普通智能家居系统的规模基本都已涵盖在本文实验中，因此，可以看出，门神在问题处理规模上具有高可扩展性，基本能够符合正常用户需求。

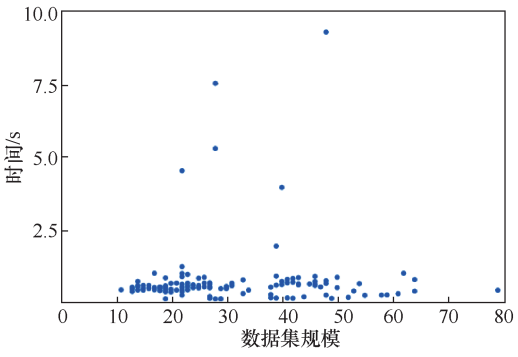


图 10 规模和时间的散点

7 相关工作

智能家居与人们的日常生活息息相关，因此，保证智能家居系统的安全性十分必要，目前，有很多工作都致力于对智能家居系统安全性的分析与研究。

1) IoT 验证与测试

采用形式化建模、验证及模型层面测试仿真来分析 IoT 系统正确性这一方向近年来得到了相关领域的广泛关注。Leelaprute 等[2]设计了一套描述家居系统的语言，再转换到 SMV 语言建立模型进行验证；Corno 等[3]则讨论了如何使用状态图等形式化技术对家居系统进行基础建模，这些工作提出了各自的建模方法，但无法实现自动化，普通用户无法掌握。以 DepSys[14]、SIFT[4]等为代表的工作采用形式化建模与符号执行等方式，通过验证、测试等手段进行系统冲突检测，如设备冲突、规则冲突等。本文的工具平台提供了更加完整的智能家居系统安全性建模与验证，判定用户定义的规则是否会造成危险，能够解决多种问题。

2) 静态安全分析

Surbatovich 等[15]提出了一套对 IFTTT 规则进行分析的静态方法，定义了一套信息流模型，人工标定 IFTTT 网站上规则中的各种事件信息，检查信息流冲突，发现问题规则。该方法是对 IFTTT 规则进行静态分析，而本文设计了一套自动化的建模与验证方法，支持事件的可信赖度和隐私度建模，分析系统规则的安全性。

3) 系统架构

还有一些工作是从系统架构方面着眼，致力于保证智能家居系统的安全性。Li 等[16]从设计层面提出了一套的智能家居安全结构；Riahi 等[17]则基于系统和网络拓扑提出了一种智能家居系统设计和配置方法，列举了其中多种重要因素。但这些工作主要集中在系统架构设计层面，不涉及用户自定义规则带来的具体行为影响。

4) 平台实现

同样有相关工作致力于家居系统仿真平台的开发[18-19]，来帮助用户对其定义规则进行仿真与展示。然而这类工作只关注家居系统仿真功能的实现，并不涉及智能家居系统安全性的研究。

本文所实现的检验平台门神的技术创新来自于本团队的前期工作[5-6]。在系统模型的基础上，支持事件的可信赖度和隐私度建模，同时定义安全相关问题，实现规约的自动化生成，将整个系统的模型构建和规约生成流程自动化，在此基础上设计了一个智能家居物联网系统仿真和安全验证工具，使用者只需要提供设计规则，就可以使用工具进行一键式安全仿真和验证。

8 结束语

IFTTT 框架使得普通用户可以自定义家居规则，设计自己的智能家居系统，但用户不当的设计往往会引发安全问题。智能家居和人们的生活紧密相关，保证其安全性是至关重要的。

For Research Only

本文设计并实现了第一个面向事件触发智能家居物联网系统的仿真和安全验证工具平台——门神，可帮助用户对其自定义的家居系统进行自主设置并自动化验证其安全性，以保障用户安全。

参考文献：

[1] UR B, MCMANUS E, HO M P Y, et al. Practical trigger-action programming in the smart home[C]//Sigchi Conference on Human Factors in Computing Systems. ACM, 2014: 803-812.

[2] LEELAPRUTE P, NAKAMURA M, TSUCHIYA T, et al. Describing and verifying integrated services of home network systems[C]// Asia-Pacific Software Engineering Conference. IEEE, 2005: 549-558.

[3] CORNO F, SANAULLAH M. Modeling and formal verification of smart environments[J]. Security & Communication Networks, 2015, 7(10): 1582-1598.

[4] LIANG C J M, LANE N D, ZHAO F, et al. SIFT: building an Internet of safe things[C]//International Conference on Information Processing in Sensor Networks. ACM, 2015: 298-309.

[5] LIANG C J M, BU L, LI Z, et al. Systematically debugging IoT control system correctness for building automation[C]//ACM International Conference on Systems for Energy-Efficient Built Environments. ACM, 2016: 133-142.

[6] BU L, XIONG W, LIANG C J M, et al. Systematically ensuring the confidence of real-time home automation IoT systems[J]. ACM Transactions on Cyber-Physical Systems, 2018, 2(3): 1-23.

[7] BAIER C, KATOEN J P. Principles of model checking[M]. Cambridge: The MIT Press, 2008.

[8] HOLZMANN G J. The model checker SPIN[J]. IEEE Transactions on Software Engineering, 1997, 23(5): 279-295.

[9] CIMATTI A, CLARKE E, GIUNCHIGLIA F, et al. NuSMV: a new symbolic model checker[J]. International Journal on Software Tools for Technology Transfer, 2000, 2(4): 410-425.

[10] BU L, LI Y, WANG L, et al. BACH : bounded reachability checker for linear hybrid automata[C]//Formal Methods in Computer-Aided Design. IEEE, 2008: 1-4.

[11] CROCKFORD D. The application/JSON media type for JavaScript object notation (JSON)[J]. RFC, 2006, 13(4)：250-251.

[12] CLARKE E M, EMERSON E A. Design and synthesis of synchronization skeletons using branching time temporal logic[C]//The Workshop on Logic of Programs. Springer, 1981: 52-71.

[13] PNUELI A. The temporal logic of programs[M]. Rehovot: Weizmann Science Press, 1977.

[14] MUNIR S, STANKOVIC J A. Depsys: dependency aware integration of cyber-physical systems for smart homes[C]//ACM/IEEE International Conference on Cyber-Physical Systems. IEEE, 2014: 127-138.

[15] SURBATOVICH M, ALJURAIDAN J, BAUER L, et al. Some recipes can do more than spoil your appetite: analyzing the security and privacy risks of IFTTT recipes[C]//International World Wide Web Conference. ACM, 2017: 1501-1510.

[16] LI H, ZHOU X. Study on security architecture for Internet of things[C]//International Conference on Applied Informatics and Communication. Springer, 2011: 404-411.

[17] RIAHI A, CHALLAL Y, NATALIZIO E, et al. A systemic approach for IoT security[C]//IEEE International Conference on Distributed Computing in Sensor Systems. IEEE, 2013: 351-355.

[18] RIERA B, VIGÁRIO B. HOME I/O and FACTORY I/O: a virtual house and a virtual plant for control education[J]. IFAC-Papers on Line, 2017, 50(1): 9144-9149.

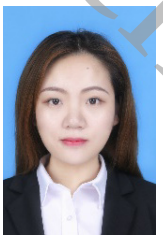
[19] ALSHAMMARI N, ALSHAMMARI T, SEDKY M, et al. OpenSHS: open smart home simulator[J]. Sensors, 2017, 17(5): 1003.

[作者简介]

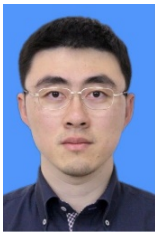
张秋萍（1994− ），女，江苏苏州人，南京大学硕士生，主要研究方向为形式化方法和模型检验。

王熙灶（1995− ），男，江西高安人，南京大学博士生，主要研究方向为程序分析和程序验证。

沈思远（1997− ），男，湖北黄冈人，南京大学硕士生，主要研究方向为以太坊智能合约代码优化、测试和验证。



张时雨（1995− ），女，辽宁阜新人，南京大学硕士生，主要研究方向为形式化方法和模型检验。



卜磊（1983− ），男，江苏东台人，博士，南京大学副教授、博士生导师，主要研究方向为形式化方法、模型检验、实时混成系统和信息物理融合系统。

李宣东（1963− ），男，湖南邵东人，博士，南京大学教授、博士生导师，主要研究方向为软件工程、软件建模与分析、软件测试与验证。

For Research Only