



























额外的工作量。

在智能设备运行时模型基础上,开发者仅声明面向场景的情境知识,配置概念实例与智能设备的映射关系,描述服务对象所处情境的约束条件,就能够自动构建智能家居情境感知服务。

### 5.3.2 智能家居情境感知服务的开发难度比较

表 10 对使用两种方法开发的智能家居情境感知服务的代码行数进行比较,其中,使用 Java 语言开发服务,每个服务独立开发,使用本文方法时,开发者则需要先实现面向场景的相关配置,即基础代码为 115 行。例如, $C_{11}$  是 Jack 的温度调节服务,Java 程序的代码行数为 220 行,其中,接口调用相关代码为 136 行,管理逻辑相关代码为 84 行,本文方法的新增配置行数为 6 行; $C_{24}$  是 Ken 的空气调节服务,Java 程序的代码行数为 140 行,其中,接口调用相关代码为 68 行,管理逻辑相关代码为 72 行,本文方法的新增配置行数为 6 行。Java 程序的平均代码行数为 186 行,而本文方法的平均配置行数为 16 行,其代码减少量超过 90%。

Table 10 Comparison in lines of code of two approaches

表 10 两种方法的服务代码行数比较

	Basic	$C_{11}$	$C_{13}$	$C_{14}$	$C_{21}$	$C_{23}$	$C_{24}$	$C_{31}$	$C_{33}$	$C_{42}$	$C_{43}$	$C_{52}$	$C_{53}$	Average
Java	0	220	198	140	220	198	140	220	197	163	188	163	188	186
Our approach	115	6	6	6	6	6	6	6	6	6	6	6	6	16

图 5 对使用两种方法的服务开发时间进行比较:使用 Java 语言开发单个服务的平均时间是 40min,且每个服务独立开发,开发时间随服务数量的增加而线性增长;使用本文方法时,开发者需要先花费固定时间配置面向场景的情境知识、概念实例与智能设备的映射关系,再根据服务需求,逐个配置服务对象所处情境的约束条件,配置每个约束的平均时间是 5min。因此,对于绝大多数智能家居场景,当智能设备充分发挥作用时(即服务达到一定数量),相对于使用通用语言进行服务开发,本文方法能够大幅度降低其开发时间。

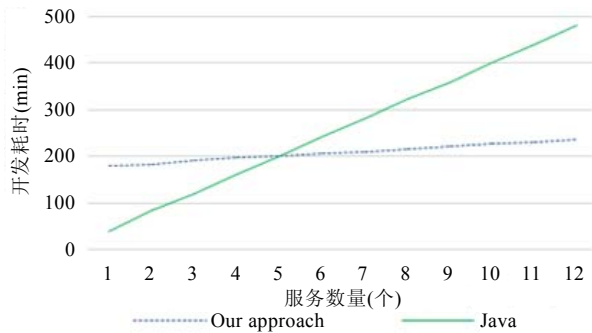


Fig.5 Comparison in development time of two approaches

图 5 两种方法的服务开发时间比较

### 5.3.3 智能家居情境感知服务的执行性能比较

为了比较两种方法的服务执行性能,在配置“CPU:3.1GHz;Memory:4GB”的系统环境下,执行不同数量的智能家居情境感知服务,并统计其平均响应时间,如图 6 所示。在 Java 程序中,每个服务顺序执行,分别执行管理逻辑并调用设备 API,因此,其平均响应时间随服务数量的增加而线性增长。使用本文方法时,知识图谱实例模型通过模型操作转换及设备 API 调用,实现与场景实时信息的双向同步,在此基础上,根据服务需求进行知识推理:一方面,由于需要额外操作来维护实例模型与智能家居场景的运行时同步,当服务数量少时,与 Java 程序相比,本文方法平均响应时间较高,从智能服务的角度看,这种性能上的差异是可以被接受的;另一方面,由于 Java 程序中服务独立执行,当服务达到一定数量时,会出现不同情境感知服务调用相同设备 API 获取场景信息的情况(例如, $C_{11}$ 、 $C_{21}$  和  $C_{31}$  均要监测客厅温度),当存在大量服务时,与 Java 程序相比,本文方法能够有效降低设备 API 重复调用产生的额外开销,平均响应时间较低。

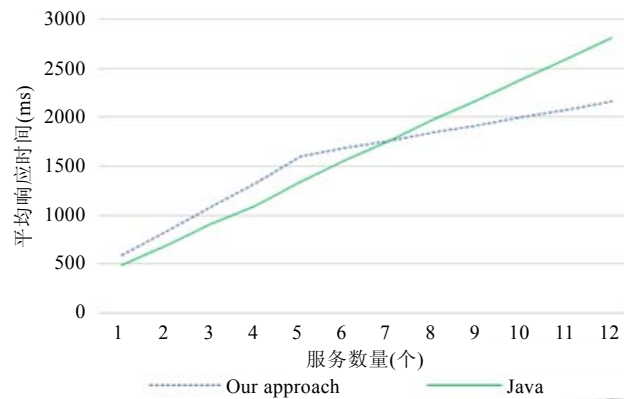


Fig.6 Comparison in execution time of two approaches

图6 两种方法的服务执行性能比较

## 6 相关工作

在当前的物联网应用开发过程中,编程工作一般都接近操作系统级别,这要求程序员对底层系统的相关技术有非常深入的了解,使程序员将注意力集中在底层系统的相关问题上,而不是应用逻辑本身<sup>[11]</sup>.存在一些研究工作,希望在保证效率的前提下,尽可能地简化物联网应用的开发过程.文献[12]提出一种面向物联网的认知管理框架,框架使用虚拟对象(virtual object)表示现实世界的对象,并使用组合虚拟对象(composite virtual object)表示一组可互操作的虚拟对象的聚合体,于是,可以面向组合虚拟对象开发物联网应用.文献[9,13]提出一种基于运行时软件体系结构模型的物联网应用开发方法,将设备能力抽象为运行时模型,通过模型转换实现应用场景模型与设备运行时模型的双向同步,于是,可以面向应用场景模型开发物联网应用.文献[14,15]提出一种基于ECA规则的物联网应用开发方法及支撑系统,给定一组面向应用场景的ECA规则,系统能够对其校验并自动生成面向底层系统的执行代码.文献[16]提出一种自顶向下的物联网应用开发方法及支撑系统,给定平台独立的应用管理逻辑,系统能够自动生成平台相关的具体配置和执行代码.此外,一些研究工作<sup>[17-19]</sup>提出了基于面向服务体系结构的解决方案,提供 RESTful 服务形式的设备访问接口,以屏蔽底层系统的异构性和复杂性.尽管上述工作有效提高了物联网应用开发的抽象层次,但是开发者仍需面向场景手工编写物联网应用的管理逻辑.

为了支持物联网应用的服务组装和知识推理,一些研究工作提出将语义网的概念移植到物联网中,以提供物联网语义服务.文献[20]将注释嵌入到用 XML 表示的观测数据中,以准确地描述数据的语义,采用 OWL<sup>[21]</sup>建立物联网本体,并采用规则语言进行本体推论.文献[22]面向物联网领域提出统一的语义知识基础,包括资源、位置、环境、领域、规则和服务等本体,用于服务发现和组装.文献[23]面向物联网应用提出基于本体的语义建模与演化方法,包括通用上层本体以及一组用于领域本体扩展的柔性接口.上述工作在物联网传感数据、软件服务的基础上进行语义建模,但是缺乏其语义模型与底层数据、服务的联动机制.

北京大学团队在运行时模型理论及构造方法方面进行了研究<sup>[7,8,24-26]</sup>:给定系统元模型与一组管理接口, SM@RT 工具<sup>[10]</sup>就能自动生成代码,在保证性能的前提下实现模型到管理接口的映射;当系统元模型发生变化时, SM@RT 可以自动生成新的映射代码.我们进一步提出了基于运行时模型的物联网应用开发方法<sup>[9,13]</sup>.在上述前期工作基础上,本文面向智能家居场景进行语义建模,并建立语义模型与底层系统的运行时同步机制.

## 7 总结

智能家居情境感知服务需要面向场景进行开发,智能设备的多样性和智能服务的按需性,给其开发带来极大的难度和复杂度.本文提出一种智能家居情境感知服务的运行时建模与执行方法,通过知识图谱概念模型描述智能家居场景中概念和关系等抽象元素,通过运行时知识图谱实例模型表示智能家居情境知识,通过建立在

上述模型上的知识推理自动执行设备功能.于是,开发者仅声明面向场景的情境知识,配置概念实例与智能设备的映射关系,描述服务对象所处情境的约束条件,就能自动构建智能家居情境感知服务.本文方法能够极大地降低开发智能家居情境感知服务的难度和复杂度.

未来工作的重点主要包含以下两个方面:一方面,基于模型检测技术<sup>[27,28]</sup>对智能家居知识图谱实例模型进行深度分析,以保证实例模型和知识推理的正确性及完整性;另一方面,将方法运用到实际的智能家居场景中,并完善特定场景下的支撑机制.

## References:

- [1] Xu K, Wang XL, Wei W, Song HB, Mao B. Toward software defined smart home. *IEEE Communications Magazine*, 2016,54(5): 116–122. [doi: 10.1109/MCOM.2016.7470945]
- [2] Atzori L, Iera A, Morabito G. The Internet of things: A survey. *Computer Networks*, 2010,54(15):2787–2805. [doi: 10.1016/j.comnet.2010.05.010]
- [3] Dixon C, Mahajan R, Agarwal S, Brush AJ, Lee B, Saroiu S, Bahl P. An operating system for the home. In: *Proc. of the Usenix Conf. on Networked Systems Design and Implementation*. Berkeley: USENIX Association, 2012. 25–25.
- [4] Singhal A. Introducing the knowledge graph: Things, not string, official blog, of Google. 2012. <http://goo.gl/zivFV>
- [5] Liu Q, Li Y, Duan H, Liu Y, Qin ZG. Knowledge graph construction techniques. *Journal of Computer Research and Development*, 2016,53(3):582–600 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2016.20148228]
- [6] Guan SP, Jin XL, Jia YT, Wang YZ, Cheng XQ. Knowledge graph oriented knowledge inference method: A survey. *Ruan Jian Xue Bao/Journal of Software*, 2018,29(10):2966–2994 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5551.htm> [doi: 10.13328/j.cnki.jos.005551]
- [7] Huang G, Song H, Mei H. SM@RT: Towards architecture-based runtime management of Internetware systems. In: *Proc. of the Asia-Pacific Symp. on Internetware*. ACM, 2009. 1–10. [doi: 10.1145/1640206.1640215]
- [8] Song H, Huang G, Chauvel F, Xiong YF, Hu ZJ, Sun YC, Mei H. Supporting runtime software architecture: A bidirectional-transformation-based approach. *Journal of Systems & Software*, 2011,84(5):711–723. [doi: 10.1016/j.jss.2010.12.009]
- [9] Chen X, Li AP, Zeng XE, Guo WZ, Huang G. Runtime model based approach to IoT application development. *Frontiers of Computer Science*, 2015,9(4):540–553. [doi: 10.1007/s11704-015-4362-0]
- [10] Peking University. SM@RT: Supporting models at run-time. 2009. <http://code.google.com/p/smatr/>
- [11] Mottola L, Picco GP. Programming wireless sensor networks: Fundamental concepts and state of the art. *ACM Computing Surveys*, 2011,43(3):1–51. [doi: 10.1145/1922649.1922656]
- [12] Vlacheas P, Giaffreda R, Stavroulaki V, Kelaidonis D, Foteinos V, Poullos G, Demestichas P, Somov A, Biswas RB, Moessner K. Enabling smart cities through a cognitive management framework for the internet of things. *IEEE Communications Magazine*, 2013, 51(6):102–111. [doi: 10.1109/MCOM.2013.6525602]
- [13] Chen X, Zhang W, Huang G, Li AP, Guo WZ, Chen GL. Management approach of wireless sensor networks based on runtime model. *Ruan Jian Xue Bao/Journal of Software*, 2014,25(8):1696–1712 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4665.htm> [doi: 10.13328/j.cnki.jos.004665]
- [14] Cano J, Rutten E, Benazzouz Y, Gurgun L. ECA rules for iot environment: A case study in safe design. In: *Proc. of the IEEE 8th Int'l Conf. on Self-adaptive and Self-organizing Systems Workshops*. Piscataway: IEEE, 2014. 116–121. [doi: 10.1109/SASOW.2014.32]
- [15] Chen YT, Chen CC, Chang HY, Lin HS, Chang HT. Constructing ECA rule for IoT application through a novel S2RG process: The exemplary ECA rules for smarter energy applications. In: *Proc. of the Int'l Computer Symp.* Piscataway: IEEE, 2017. 549–554. [doi: 10.1109/ICS.2016.0114]
- [16] Guan GY, Dong W, Gao Y, Fu KB, Chen ZH. TinyLink: A holistic system for rapid development of IoT applications. In: *Proc. of the Int'l Conf. on Mobile Computing and Networking*. New York: ACM Press, 2017. 383–395. [doi: 10.1145/3117811.3117825]
- [17] Spiess P, Karnouskos S, Guinard D, Savio D, Baecker O, Souza LMSD, Trifa V. SOA-based integration of the Internet of things in enterprise services. In: *Proc. of the IEEE Int'l Conf. on Web Services*. Piscataway: IEEE, 2009. 968–975. [doi: 10.1109/ICWS.2009.98]

- [18] Janowicz K, Bröring A, Stasch C, Sachade S, Everding T, Llaves A. A RESTful proxy and data model for linked sensor data. *Int'l Journal of Digital Earth*, 2013,6(3):233–254. [doi: 10.1080/17538947.2011.614698]
- [19] Google physical Web. 2018. <http://google.github.io/physical-web/>
- [20] Sheth A, Henson C, Sahoo SS. Semantic sensor Web. *IEEE Internet Computing*, 2008,12(4):78–83. [doi: 10.1109/MIC.2008.87]
- [21] Web ontology language (OWL). 2004. <http://www.w3.org/TR/owl-ref/>
- [22] Nambi SNAU, Sarkar C, Prasad RV, Rahim A. A unified semantic knowledge base for IoT. In: *Proc. of the Internet of Things*. Piscataway: IEEE, 2014. 575–580. [doi: 10.1109/WF-IoT.2014.6803232]
- [23] Ma M, Wang P, Chu CH. Ontology-based semantic modeling and evaluation for Internet of things applications. In: *Proc. of the Internet of Things*. Piscataway: IEEE, 2014. 24–30. [doi: 10.1109/iThings.2014.13]
- [24] Huang G, Mei H, Yang FQ. Runtime recovery and manipulation of software architecture of component-based systems. *Automated Software Engineering*, 2006,13(2):257–281. [doi: 10.1007/s10515-006-7738-4]
- [25] Mei H, Huang G, Lan L, Li JG. A software architecture centric self-adaptation approach for Internetwork. *Science China Information Sciences*, 2008,51(6):722–742. [doi: 10.1007/s11432-008-0052-y]
- [26] Song H, Xiong YF, Chauvel F, Huang G, Hu ZJ, Mei H. Generating synchronization engines between running systems and their model-based views. In: *Proc. of the Int'l Conf. on MODELS in Software Engineering*. Berlin: Springer-Verlag, 2009. 140–154.
- [27] Zhang PC, Muccini H, Li BX. A classification and comparison of model checking software architecture techniques. *Journal of Systems & Software*, 2010,83(5):723–744. [doi: 10.1016/j.jss.2009.11.709]
- [28] He X, Chen X, Cai S, Zhang Y, Huang G. Testing bidirectional model transformation using metamorphic testing. *Information and Software Technology*, 2018,104:109–129.

#### 附中文参考文献:

- [5] 刘峤,李杨,段宏,刘瑶,秦志光.知识图谱构建技术综述. *计算机研究与发展*,2016,53(3):582–600. [doi: 10.7544/issn1000-1239.2016.20148228]
- [6] 官赛萍,靳小龙,贾岩涛,王元卓,程学旗.面向知识图谱的知识推理研究进展. *软件学报*,2018,29(10):2966–2994. <http://www.jos.org.cn/1000-9825/5551.htm> [doi: 10.13328/j.cnki.jos.005551]
- [13] 陈星,张伟,黄罡,李隘鹏,郭文忠,陈国龙.基于运行时模型的无线传感网管理方法. *软件学报*,2014,25(8):1696–1712. <http://www.jos.org.cn/1000-9825/4665.htm> [doi: 10.13328/j.cnki.jos.004665]



陈星(1985—),男,福建永春人,博士,副教授,博士生导师,CCF 专业会员,主要研究领域为系统软件,软件自适应,云计算.



马郢(1989—),男,博士,CCF 专业会员,主要研究领域为系统软件,Web,移动计算,服务计算.



黄志明(1994—),男,硕士生,CCF 学生会员,主要研究领域为系统软件,软件自适应,软件中间件.



陈芝燕(1995—),女,硕士生,主要研究领域为自然语言处理.



叶心舒(1994—),女,硕士,主要研究领域为分布式系统,软件中间件,软件工程.



郭文忠(1979—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算智能及其在计算机网络中的应用研究.