The 5D Approach to Control and Manage Smart Spaces

Hätönen, Seppo

IEEE
2017

Hätönen , S , Mineraud , J , Rao , A , Flinck , H & Tarkoma , S 2017 , The 5D Approach to Control and Manage Smart Spaces . in 5G - European Roadmap, Global Impact : EUCNC: European Conference on Networks and Communications, Oulu, Finland, 2017 June 12-15 . IEEE , pp. 1-6 , European Conference on Networks and Communications , Oulu , Finland , 12/06/2017 . https://doi.org/10.1109/EuCNC.2017.7980641

http://hdl.handle.net/10138/347909
https://doi.org/10.1109/EuCNC.2017.7980641

The 5D Approach to Control and Manage Smart Spaces

Seppo Hätönen*, Julien Mineraud*, Ashwin Rao*, Hannu Flinck†, and Sasu Tarkoma*
*University of Helsinki, †Nokia Bell Labs
{firstname.lastname}@{cs.helsinki.fi,nokia-bell-labs.com}

***Abstract*—Our homes, offices, and other spaces are expected to evolve into smart spaces served by devices having varying requirements and capabilities. To efficiently control and manage these devices, their controllers needs to be designed using the correct abstraction for the devices. In this paper, we present an abstraction for the devices in smart spaces, and we use this abstraction to present a 5D—*deploy*, *discovery*, *decision*, *dissemination*, and *data*—approach to control and manage smart spaces. We also discuss three approaches to design controllers for smart spaces, and highlight how controllers can leverage recent research in distributed systems. We believe that our abstraction for devices, our 5D approach, and our approaches for designing controllers are building blocks for transforming our spaces to smart spaces.**

***Index Terms*—5D, device abstraction, IoT, controller.**

I. Introduction

Our homes, offices, and other spaces are being transformed into smart spaces with devices that have varying capabilities, requirements, and deployment densities. Furthermore, the capabilities of these devices and their requirements from the communication infrastructure are expected to evolve [1].

The efficient utilization of these devices requires their capabilities and requirements to be turned into abstractions, which can be used to design controllers for smart spaces. These abstractions are important given the high heterogeneity of devices in smart spaces [2], and they provide an ideal vantage point to design controllers for complex systems by distilling the underlying simplicity of complex systems [3].

In this paper, we first abstract the requirements and the capabilities of devices in smart spaces. We then use these abstractions for our 5D approach—*deploy*, *discovery*, *decision*, *dissemination*, and *data*—to control and manage smart spaces. Our approach extends the 4D approach [4] by explicitly including *deploy* as device deployment in smart spaces has constraints, and is expected to be carried out by users with limited knowledge on the smart space infrastructure. The insights from our 5D approach are then leveraged for controller design for smart spaces. In particular we observe that the CAP theorem for distributed systems [5] will play a crucial role in designing the controllers. *We believe that our work can be the building blocks for designing and implementing the control plane of smart spaces.* Our key contributions are as follows.

- We present an abstraction for devices that can be leveraged to design the control plane of smart spaces.
- We present our 5D approach which explicitly includes a *deploy* plane because device deployment is expected to be performed by end-users. We also discuss some key open research problems related to the *deploy* plane. To the best of our knowledge, we are the first to map the 4 D's of the 4D approach to smart spaces.
- We also discuss the benefits and shortcomings of three approaches to design the controllers for smart spaces.

Roadmap. In §II we use an example smart home to highlight issues and challenges in controlling and managing smart spaces. In §III we present an abstraction for devices, their requirements, and capabilities. We then present our 5D approach to control and manage smart spaces in §IV. In §V, we leverage the insights from our 5D approach and the device abstractions to present three different approaches to design a controller for smart spaces. We present our conclusions in §VI.

II. Background and Motivation

We now present an example smart home to highlight the issues in transforming our spaces to smart spaces.

A. An Example Smart Home

As shown in 1, a smart home includes devices that monitor the environment in the house, and automate and control the house-hold appliances. For example, sensors gather and report information such as temperature, luminosity, and presence of people, while smart appliances such as washing machines, allow users to remotely control them. Smart spaces will also include devices such as laptops and phones whose physical location is not restricted to a single smart space.

One or more controllers manage these devices and the resources they use. These controllers can either be in the cloud, or they can be in the smart space [6]. Currently, controllers are vendor specific and they manage devices only from certain vendors (*i.e.*, siloed Internet-of-Things solutions [2]).

B. Key Issues in Controlling and Managing Smart Spaces

Our example of a smart home helps us illustrate the following key issues in managing smart spaces.

1) *Abstractions for Devices.* Smart spaces are expected to contain devices with varying capabilities and constraints. For example, devices can be battery powered and have constraints on communications range and supported protocols. Satisfying these requirements while considering their capabilities is essential for creating smart spaces. Furthermore, systems built

978–1–5386–3873–6/17/$31.00

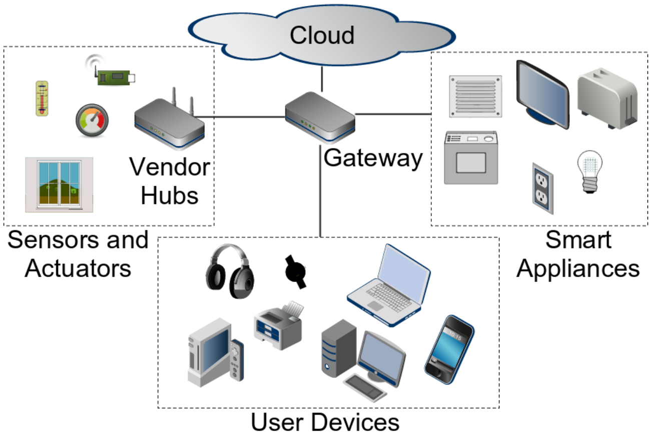

Fig. 1. **An example smart home.** *A smart home is expected to contain a variety of devices. The heterogeneity of devices motivates the need to provide abstractions that can be leveraged to manage and control smart spaces.*

using correct abstractions for devices will ensure that these smart spaces can evolve to support the influx of new devices.

2) *Architecture for the Management and Control Plane.* The heterogeneity of devices, and their requirements and capabilities, makes the management and control plane of smart spaces inherently different from traditional communication networks. An architecture for the management and control plane is vital to efficiently utilize the devices in the smart space.

III. Abstraction for Devices in Smart Spaces

Smart spaces are expected to contain a plethora of devices. To manage these devices and to support an influx of new devices, the control and management plane for smart spaces requires an abstraction for these devices. Devices in smart spaces can be categorized as follows.

1) *Things* include a) devices that sense and monitor the physical environment, b) devices that are expected to perform physical actions, and c) devices whose primary function is something other than converting spaces to smart spaces. Examples include motion sensors, temperature sensors, garage doors, home appliances such as ovens, laptops, *etc.*
2) *Controllers* are responsible for controlling and managing devices in a smart space. Common examples include an IoT hub, and a home gateway that manages the home network.

These devices have constraints for device deployment, device discovery, and device control. Furthermore, device deployment includes constraints on a) energy, b) communication interfaces, communication range and protocols, c) location and naming, and d) physical attributes such as size.

A. Things

Smart spaces are expected to include devices performing a combination of a) sensing and monitoring the physical environment, and b) taking physical actions based on a user-defined policy. Some of these devices may leave a smart space, and a smart space should be usable even in their absence.

Constraints on device deployment. Some key constraints on deployment of monitoring devices are as follows.

1) *Energy.* Things may be powered using batteries, wireless back-scatter [7], or via the mains electricity. For example, garage doors consume significant amounts of energy and are expected to be powered using the mains electricity, while fire alarms are more likely to be powered using batteries. The energy source inherently imposes constraints on the physical location of a device and also the communication range and protocols used by the device.
2) *Communication.* The available energy, physical size, volume of data exchanged, and the physical location drives the choice of communication interfaces and the protocols supported by these devices. For example, protocols such as ZigBee [8] and Bluetooth [9] have a limited communication range, even limited to line of sight in some cases.
3) *Location and Naming.* The device location and name can either be explicitly configured, or it can implicitly inferred using localization techniques [10]. Location and naming is vital to ensure that only the intended set of devices are performing the desired task [11], and is critical for managing networks with a large number of devices.
4) *Physical Attributes.* Physical attributes such as size and appearance depends on the device function. For example, smart devices such as microwaves and washing machines must have similar dimensions as the existing devices they will replace. Similarly, motion detectors and smart lights are expected to be pervasive and must have similar form factors as existing devices.

These deployment constraints play a crucial role in determining how monitoring devices are integrated into a smart space.

Device Discovery. The communication channel used by devices can be explicitly configured using static rules, or the end-points of the communication can discover each other by broadcasting their presence over the communication medium. An example protocol for discovering monitoring devices is *ioDP*, a proprietary discovery protocol used in the *realTime.io* IoT platform (see Platform 27 of [2]). Furthermore, devices such as televisions and projectors which have fully operational TCP/IP stack [2] are capable of using standard network discovery protocols such as DNS Service Discovery [12], UPnP discovery protocols [13], and DHCP [14].

Control and Management. The control and management plane enables the device to be reconfigured with the details on how to a) perform the desired task, b) report the results to the intended recipients, and c) efficiently utilizes the available resources such as energy. For example, a light bulb can be set to a given luminosity and can be configured to report its luminosity.[1] Similarly, when multiple communication technologies such as ZigBee and Bluetooth Low Energy (BLE) are available, a key parameter that impacts energy efficiency is the choice of communication technology [15]. Furthermore, Controllers can minimize the wireless interference by dynamically programming the Things to use one of their available communication technologies. At the same time, devices such as smart-phones can be be used to configure the discovery settings of devices such as garage-doors.

[1] http://www2.meethue.com/

Data plane. The network traffic characteristics such as traffic volume vary with the task and the timeliness of its results [1]. As a consequence, the traffic patterns are expected to vary across multiple devices. However, the traffic volume generated by a bulk of these devices is expected to many orders of magnitude smaller than the traffic volumes generated by activities such as video streaming and gaming [16]; an exception to this are video sensors, such as surveillance cameras.

B. Controllers

Controllers gather information from the Things in a smart space for making and disseminate decision to comply with the user-defined policy. A controller can either manage all the Things in a smart space or a subset of the Things.

Constraints on device deployment. Some key constraints on the deployment of controllers are as follows.

1) *Energy.* Controllers are likely to use mains power because they are expected to be available at all times.
2) *Communication.* A controller's communication interfaces depends on the communication technologies used by the Things under its command, and the Things users use to access the controller. For example, the controller might use Ethernet to communicate with the rest of the Internet, but the Things under its command may use ZigBee.
3) *Location and Naming.* A controller's location is constrained by the communication range of the Things it manages. The controller also requires the name and other meta-data such as the location to uniquely identify itself and the Things.
4) *Physical Attributes.* Some controllers need to be in locations suitable for audio-visual control (e.g., Amazon Echo), while some must be accessible for manual override.

Device Discovery. A controller is responsible for discovering the Things under its control. End-users can manually provide the details of the Things to the controller and the details of the controller to the Things, or the Things and their controller can mutually discover each other by broadcasting their presence.

Control and Management plane. Controllers are the heart of the control and management plane of smart spaces. The controllers collect the data from the Things under their command, take decisions by coordinating with other controllers, and disseminate the decision to the Things under their command. We discuss approaches to design controllers in section V.

Data/user plane. Controllers receive data gathered by the Things. The controller is expected to compile the received data and share this compiled data with users. Data traffic generated by controllers therefore contains data shared with its users.

IV. 5D Approach

We now use our abstraction to present our 5D approach—*deploy*, *discovery*, *decision*, *dissemination*, and *data*—to control and manage smart spaces. We extend the 4D approach [4] for traditional networks by explicitly including *deploy* as the fifth D. We do this because unlike traditional networks where devices are installed by skilled technicians, end-users with limited knowledge of the inner workings of smart spaces are expected to deploy devices in smart spaces. Furthermore, we map the 4 D's of the 4D approach to smart spaces, and to the best of our knowledge we are the first to do so.

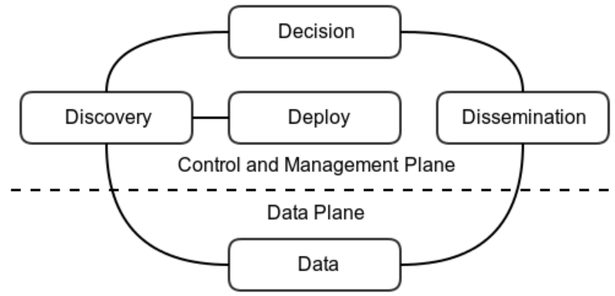

Fig. 2. The 5D approach: *deploy*, *discovery*, *decision*, *dissemination*, and *data*, to capture the life-cycle of devices in smart spaces.

As shown in Figure 2, a device (a Thing or a controller) enters a smart space through the *deploy* plane. In this plane, users deploy the device and perform initial calibration and configuration of the device. Then, this device, its controllers, and other devices discover each other in the *discovery* plane. On successful discovery, users see the deployed device and its capabilities on the management user-interface of the smart space. The device is then included in the *decision* plane. Based on the current state and capabilities of the devices, the controllers decide if any actions need to be performed by the devices. If a new decision is taken, the decision is disseminated to the concerned devices in the *dissemination* plane. *The control and management plane of smart spaces includes the* deploy*,* discovery*,* decision*, and* dissemination *planes.* In the *data* plane, devices in a smart space exchange data among themselves, and some of these devices exchange data with other devices in the Internet. We now detail these five Ds that capture the life-cycles of devices in smart spaces.

A. Deploy Plane

To successfully deploy a device users must address the constraints on device deployment discussed in the previous section. Regardless, these constraints limit the choice of a suitable physical locations for the devices. The device deployment is complete when the user is able to initiate the discovery of the device with a controller and other devices with whom this device needs to communicate. For example, user might press a button on a lamp to initiate the discovery of its controller.

In the 4D approach [4], the *discovery* plane implicitly included the *deploy* plane. This was done because deployment of networking devices such as switches and routers was assumed to be performed by skilled technicians. This is not true for smart spaces because devices are expected to be deployed by end-users who have limited understanding of the working of these devices. The issues that arise during deployment are expected to be addressed by users, and a user might have to re-deploy a device when changing its location or when it is not successfully discovered. We therefore decouple the *deploy* plane from the *discovery* plane. In §IV-F, we discuss some open research problems related to this plane.

B. Discovery Plane

This plane enables the discovery of a) devices, b) the change in device state, and c) the network traffic to and from devices. This information provides a holistic view of the smart space.

A controller discovers devices by either implicitly inferring the device presence [17], or by explicit discovery requests including probes from the devices [12], [13], [14]. A benefit of implicit discovery is that the device is not required to broadcast its presence to the controller. For example, a controller can infer a device's presence by observing traffic to the device [18], [19]. The network traffic flows of such devices may be managed by the controller, however the controller will not be able to send them commands. This shortcoming can be addressed only when the devices explicitly registers themselves at the controller by first broadcasting its presence, followed by mutual authentication with the controller.

A controller also needs to know the current state of devices under its control, which is required for reprogramming the devices according to the user-defined policies [1]. Discovery and subsequent monitoring of the network traffic is important for the controller to ensure that the device's requirements from the communication infrastructure are met, and it can also be used for detecting misbehaving devices.

C. Decision Plane

This plane is at the heart of controlling and managing smart spaces and is largely realized by controllers. The goal of the this plane is to take decisions to ensure that the smart space serves the users' requirements. These requirements are first translated to policies, and the decisions for complying with these policies are taken. Controllers responsible for a decision must form a consensus [20] before taking the decision.

A key task in this plane is taking inputs from users and translating them into policies. For example, IFTTT[2] allows users to create recipes on how the devices should interact with each other. In its simplest form, a recipe is a) a condition, such as detecting a user's presence, and b) the action to be taken when the condition is met, such as dimming lights.

Once the policy is compiled, the next task in this plane is taking decisions to enforce the policy. This includes a) composing decision which account for the current state of the devices and their capabilities, b) identifying tasks to be performed by the devices, and c) creating the commands to be sent to the devices. Enforcing the policy also includes a) optimizing resource usage of devices such as configuring the wireless medium to minimize interference, b) complying with the security constraints and quarantining misbehaving devices, and c) resolving conflicting requirements from users [21].

D. Dissemination Plane

In this plane the outcome of the decisions are disseminated. A key component of this plane is encoding and delivering the desired actions as commands to the devices. The commands can either be *pushed* to the devices, or the devices can *pull* them from their controller depending on the situation.

[2]https://ifttt.com/

TABLE I
5D APPROACH ON THINGS AND CONTROLLERS.

Plane	Things	Controller
deploy	Address constraints on energy, location and naming, communication range, and physical attributes	
discovery	Mutually discover controller	Mutually discover devices to control and communicate
decision	Does not take decisions	Takes decision
dissemination	Receive the decisions	Disseminate decisions
data	Results of tasks, and user dependent data-traffic	Publish compiled version of state of the smart space

The choice of protocols to disseminate decisions is vendor specific. Things in smart spaces can use a wide range of protocols such as HTTP, CoAP [22], *etc.*, to communicate with their controllers. Similarly, forwarding elements such as switches and gateways can be managed using specialized protocols such as CAPWAP [23] or OpenFlow [19]. The above mentioned protocols push the commands from the controller, but devices can also pull the commands. A pull is useful for disseminating decisions that are not time-critical to devices with intermittent connectivity or with energy constraints. For example, when publishing monitoring data, a sensor can pull a command to decrease its data reporting frequency.

E. Data Plane

The goal of this plane is to efficiently deliver the data to the intended recipients. Unlike traditional networks such as campus networks, the forwarding elements in smart spaces are more heterogeneous and include vendor specific IoT hubs and controllers along with traditional switches and routers. Furthermore, the data plane of smart spaces are expected to serve a wide range of communication protocols because of the heterogeneity of the devices. A data traffic flow in smart spaces can be largely categorized as either flow between devices in the same smart space, or between a devices across spaces. A flow between devices in a smart space can be either a) between controllers, b) between a controller and a Thing under its command, or c) between Things. Devices can communicate with devices outside the smart space. For example, a smart phone can stream a video from a streaming service.

The task of forwarding packets to and from devices must consider the capabilities and constraints of the devices. For example, devices with finite energy can demand large sleep cycles, and surveillance video streams can demand a desired quality of service. As a consequence, the forwarding elements must be re-programmable to ensure that the decision plane can compose commands to satisfy the requirements.

F. Discussion and Open Research Problems

The 5 D's discussed in this section encompass the life cycle of a device in the smart space: a quick and smooth device deployment, their automatic discovery, the flexibility to take decisions and compose actions, disseminate the decision, and enable a smooth flow of data between the devices.

5D approach for smart homes, an example. We now present an example on how our 5D approach and our abstraction for devices can be leveraged to convert a home to a smart home.

Consider the smart home of Figure 1 with a programmable window blinder and a programmable air conditioner. A user wishing to automate the cooling purchases battery powered motion and temperature sensors, and their controllers. The controllers needs to be deployed close to mains line and the sensors. Furthermore, the sensors or their controller need to know their physical location and be assigned a unique name for meaningful interpretation of the collected data. The new devices and their controllers need to discover each other, and the controllers need to coordinate with each other to enable users to compose actions. IFTTT exemplifies sample use cases for composing actions such as open window blinds and turn on the cooling when a person enters an empty room. Such decisions require that the controllers have discovered the current state of the devices they manage, received the data for composing the actions, and coordinated the required action to be taken. Once the decision is taken, the desired action must be encoded in commands which are disseminated to the appropriate devices: the window blinds and the air conditioner in our example. This example shows how the 5 Ds can be used to abstract the life cycle of devices in smart spaces.

Importance of *deploy*. A key difference of our approach from the previous 4D approach [4] is that we explicitly include the *deploy* plane. We include *deploy* because unlike traditional communication networks, smart spaces rely on end-users to deploy devices. Consequently, some open research questions related to *deploy* plane are as follows.

1) The physical location of the devices, and in particular the location of the controllers determines the fraction of Things in the smart space which are under the direct control of at least one controller. This in turn determines the overall performance of the system because Things beyond the reach of the controller cannot be dynamically reprogrammed by the controller. As a consequence, it is essential to quantify the impact of the location of the controllers on the system performance.
2) The location of the controllers also determines the efficiency of utilizing the available resources. It is therefore essential to quantify the wireless interference and energy consumed for a given density of controllers and Things.
3) The answers to the above questions are useful only when end-users are aware of the impact of deploying devices at specific locations. Specifically, there is a need for tools that give users get on-demand feedback when they decide to install devices at specific locations.

V. Approaches to Design Controllers

The controllers of smart spaces are the heart of the control and management plane. Of the 5 D's, the controllers are responsible for a) discovering devices, b) taking decisions based on the current state of the devices and the user requirements, and c) dissemination of the decisions to the devices. The controllers achieve these objectives by monitoring and controlling the devices, and managing the resources such as the energy consumed by these devices. We now discuss three approaches for designing controllers of smart spaces; the third approach (see §V-C) addresses the issue of scaling controllers.

A. Independent Controllers

A naive approach to build a smart space is to deploy devices from different vendors. In this approach, the smart space is expected to be fragmented with vendor-specific siloed solutions and no single controller has the complete picture of the entire smart space. Users will be forced to compile a mental image of the smart space by separately interacting with multiple vendor-specific controllers.

Benefits. This approach to convert spaces to smart spaces is easy to deploy because vendors are expected to support plug and play devices. It is particularly useful when the devices are not expected to interact with each other, or when a given vendor provides all devices required for a smart space.

Challenges. The fragmented ecosystem offers limited opportunities to compose actions and manage resources involving devices from multiple vendors. Similarly, the complexity to manage a smart space increases with the number of fragmented ecosystems in it. This approach is thus neither scalable nor user-friendly, and its shortcomings far outnumber its benefits.

B. Single Controller

A single controller can manage a smart space if a) all the devices including the controller are from the same vendor, or b) devices from multiple vendors expose API's which can be used to control them via a third-party controller.

Benefits. This approach offers a single point of control having a fine-grained view on the state of each device, enabling users to compose actions involving multiple devices.

Challenges. A single controller is not a viable approach for large smart spaces which includes devices with a limited communication range. The software and hardware implementation of such a controller must be modular to incorporate new protocols and communication interfaces. These challenges raise serious questions on the feasibility of such a controller.

C. Interconnected Controllers

In this approach, the smart space will be managed by more than one controller, and the controllers communicate and share information with each other. The controllers can be organized either hierarchically or the controllers can be meshed.

1) *Hierarchical Interconnected Controllers.* The controllers at the bottom of the hierarchy control a subset of devices from one or more vendors. These controllers are in turn manages by controllers at the top of the hierarchy.

2) *Meshed Interconnected Controllers.* The smart space is split into sub-spaces, and each sub-space is managed by a controller. The split can be performed in many ways including based on the device type, the communication range of devices, and the communication technologies supported by the controller. The end outcome of splitting the smart space must be that each device controlled by at least one controller.

Regardless of their organization, the benefits and challenges of deploying *Interconnected controllers* are as follows.

Benefits. This approach address the shortcomings of the previous two approaches by providing a unified and comprehensive view of the smart space. Users can leverage this view to compose actions which fully utilize the potential of the devices. Furthermore, this approach is scalable and controllers can be added to support the influx of new devices and also when the physical dimensions of the smart space increases.

Challenges. The Interconnected Controller is by definition a distributed controller, and such controllers must address the challenges of distributed systems. The most important challenge is the trade-off imposed by the CAP theorem, *i.e.*, Consistency, Availability, and Partition Tolerance of the control plane state in the smart space. According to Brewer [5], distributed systems with rare occurrences of network partitions should maximize combinations of consistency and availability. This can be achieved with a well-defined strategy to detect partitions, enter the partition mode and restrict some operations, followed by recovery from the partition. Under this context, the role of the user interface for the controller and the devices under its control is very important; the user must get a clear indication of the partition along with the guidelines how to resolve possible conflicts during the partition recovery.

D. Discussion and Open Research Problems

In this section, we discussed three approaches to design controllers for smart spaces. Independent controllers, though easy to deploy, will not be able to utilize the devices to their full potential. A single controller is viable in smart space with a small physical area. However, this controller must be modular in terms of its software and hardware to support the influx of new devices. Interconnected controllers address the shortcoming of the previous two approaches, however these controllers will have to address the challenge imposed by the CAP theorem. This problem is important because Interconnected Controllers can assume that a part of the controller is in the cloud, as it is with the Amazon Echo. Recent studies [5] have shown that this challenge can be addressed by giving a preference to consistency and availability by assuming network partitions as rare events. By addressing the challenges of distributed systems, the cloud and the Interconnected controllers can leverage on each other for offering a wide range of services including performing complex edge analytics [2]. Another key challenge and opportunity in designing single and interconnected controllers is the interoperability between devices from multiple vendors.

VI. Conclusions

Success of smart spaces depends on the ease of deploying devices and how efficiently they serve user requirements. A key player in transforming our spaces to smart spaces are controllers managing the devices and the resources they consume. Controllers will support an influx of devices only when they are designed using the correct device abstractions.

In this paper, we present an abstraction for devices in smart spaces which offers a vantage point to model their capabilities and constraints. From this vantage point we present our 5D approach to manage smart spaces. We then present three approaches to design controllers, and also discuss their benefits and challenges. In particular, while independent controllers offer the easiest way to setup smart spaces, interconnected controllers can support the influx of devices. We believe that our device abstractions, our 5D approach, and the deployment of interconnected controllers will be the building blocks in evolving our spaces to smart spaces.

Acknowledgment

This work is supported by the Nokia Center for Advanced Research (NCAR) and the Tekes PraNA project, and some other projects to make this three lines.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, 2010.

[2] J. Mineraud, O. Mazhelis, X. Su, and S. Tarkoma, "A gap analysis of internet-of-things platforms," *Computer Communications*, 2016.

[3] S. Shenker, M. Casado, T. Koponen, N. McKeown *et al.*, "The Future of Networking, and the Past of Protocols," *Open Networking Summit*, vol. 20, 2011.

[4] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "A Clean Slate 4D Approach to Network Control and Management," *SIGCOMM CCR.*, Oct. 2005.

[5] E. Brewer, "CAP twelve years later: How the" rules" have changed," *Computer*, vol. 45, no. 2, 2012.

[6] O. Mazhelis, E. Luoma, and H. Warma, "Defining an Internet-of-Things ecosystem," in *Internet of Things, Smart Spaces, and Next Generation Networking*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7469.

[7] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith, "Ambient Backscatter: Wireless Communication out of Thin Air," in *Proc. of ACM SIGCOMM 2013*. New York, NY, USA: ACM, 2013.

[8] M. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. Grieco, G. Boggia, and M. Dohler, "Standardized protocol stack for the internet of (important) things," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, 2013.

[9] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A software defined networking architecture for the internet-of-things," in *IEEE NOMS*, May 2014.

[10] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, 2009.

[11] C. H. Liu, B. Yang, and T. Liu, "Efficient naming, addressing and profile services in Internet-of-Things sensory environments," *Ad Hoc Networks*, 2013.

[12] S. Cheshire and M. Krochmal, "DNS-Based Service Discovery," *Internet RFCs, ISSN 2070-1721*, vol. RFC 6763, February 2013.

[13] A. Presser, L. Farrell, D. Kemp, and W. Lupton, "Upnp Device Architecture 1.1," in *UPnP Forum*, vol. 22, 2008.

[14] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," *Internet RFCs*, vol. RFC 3315, July 2003.

[15] M. Gerasimenko, V. Petrov, O. Galinina, S. Andreev, and Y. Koucheryavy, "Impact of machine-type communications on energy and delay performance of random access channel in lte-advanced," *Transactions on Emerging Telecommunications Technologies*, vol. 24, no. 4, 2013.

[16] "Cisco visual networking index: Forecast and methodology, 2014–2019," Cisco, Tech. Rep., 2015. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf

[17] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, April 2005.

[18] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "Nox: Towards an operating system for networks," *SIGCOMM CCR.*, vol. 38, no. 3, Jul. 2008.

[19] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM CCR.*, Mar. 2008.

[20] L. Lamport, "The part-time parliament," *ACM Transactions on Computer Systems (TOCS)*, vol. 16, no. 2, 1998.

[21] E. C. Lupu and M. Sloman, "Conflicts in policy-based distributed systems management," *IEEE Transactions on Software Engineering*, vol. 25, no. 6, Nov 1999.

[22] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," vol. RFC 7252, no. 7252, June 2014.
[23] P. Calhoun, M. Montemurro, and D. Stanley, "Control And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification," *Internet RFCs*, vol. RFC 5415, March 2009.