Towards Effective Smart Space Application Development: Impediments and Research Challenges

Cristiano di Flora, Jilles Van Gurp, and Christian Prehofer
Nokia Research Center, Finland
{cristiano.di-flora, gilles.van-gurp, christian.prehofer}@nokia.com

***Abstract*— State-of-the-art research and existing commercial off-the-shelf solutions provide several technologies and methods for building Smart spaces. However, developing applications on top of such systems is quite a complex task due to several impediments and limitations of available solutions. This paper provides an overview of such impediments and outlines what are the main research challenges that still need to be solved in order to enable effective development of applications and systems that fully exploit the capabilities of state-of-the-art technologies and methodologies. The paper also outlines a few specific issues and impediments that we, at the Nokia Research Center, faced in this field so far. It also sheds some light on how we are going to tackle some of the mentioned issues in the future.**

I. Impediments to smart spaces applications development

The problem in the field of pervasive or ubiquitous computing application development is not a lack of identified issues and good solutions for those issues. Research in this area has so far yielded many research prototypes, concept development and other tangible results [1]. Generally speaking, the real problem is that most of state-of-the-art solutions never cross the research laboratory door step. This is also observed in [2]. There is a lack of widely adopted, high quality software platforms that may be used to develop smart space applications for a wide range of diverse devices. Anything less than that is inherently doomed to limited adoption due to the fact that real world spaces are filled with a wide variety of devices, most of which unable to become a part of the smart space. This fact has limited the adoption and continued development of many past efforts.

A second problem is that many of the research platforms provide support for a small subset of all the relevant non-functional requirements of an effective Smart space application platform. E.g. the issue of trust seems to be a relatively popular topic. While on the one hand state-of-the-art research has mostly investigated trust issues from security and privacy perspective, on the other hand it did not thoroughly investigate the wider issues of dependability (including reliability and availability) and their impact on development and usability of the resulting applications. Similarly, a lot of state-of-the-art research focuses on semantic issues and automated reasoning. While interesting and important, we argue that for many smart space applications elaborate solutions to those issues are overkill.

Another problem relates to standard means for establishing connectivity and service discovery/delivery between devices. Despite many relevant standards and technologies for interconnecting devices (e.g. Bluetooth, IrDA, RFID, Wi-Fi, UPnP, Zigbee) exist, the basic job of sending simple messages between two devices remains technically challenging in embedded software development, especially if it needs to be done in a cross platform/vendor and technology-independent way. Such an interoperability bottleneck is having also a strong impact on all the software platforms that have been built on top of (or as part of) the mentioned technologies. This may even make them less interoperable then how they could theoretically be. Without these basic issues resolved properly, smart space application development remains limited to walled gardens of selected devices and ideal network conditions not found in the real world.

Smart space research has also been negatively affected by the limited availability of tools for rapid development and evaluation of novel smart space systems. Indeed, to the best of our knowledge, no holistic framework has been proposed so far for the effective simulation, emulation, testing, debugging, and benchmarking of smart space systems. This is especially true for multi-platform/vendor environments characterized by strong heterogeneity of the involved devices. Such an environment is needed to build a wide developer community and to significantly reduce the cost of developing applications.

As far as the adaptation of technology from academic research centers and industry is concerned, standardization processes have so far not had a very positve impact on Smart spaces research. Lengthy standardization and proprietary development technologies can lead to delay and uncertanties, which can make programming Smart spaces less interesting to the wider research community. Such a technological lag should be reduced in the future in order to align novel theories and methods to be applied to really novel technologies and systems.

II. Research Challenges and Vision

We believe a clear progress in the research area is possible by focusing on addressing the basic problems outlined in this position paper. The key challenge is to encourage and enable non-ubiquitous computing researchers and developers and stimulate them to start writing software for pervasive

Smart spaces. The way to this must involve addressing the basic issues summarized in the following:

- easy development: application development should be made easier in order to find a way to get also non ubiquitous computing researchers to write smart space application easily and quickly;
- seamless network interworking: connectivity with existing network technology should be facilitated;
- service-infrastructures interworking: interoperability with existing service-oriented architectures for embedded and general-purpose devices should be guaranteed;
- extensible future-proof platform:
  - integration of future technologies or modifications to existing technologies should be fast and as simple as possible;
  - developed Smart space applications should run not only on the the devices they were tested on but also on new devices (unmodified) when the original devices are no longer on the market;
- versatile support for different application requirements: a good enough solution for supporting a relevant subset of different non-functional requirements needs to be provided (e.g. reliability, security, and trust) in order to make the platform suitable for a wide range of applications.

A pragmatic and hands-on approach to addressing these issues will enable software developers to produce good quality and widely usable Smart space applications.

As we and others [2] have observed, many ubiquitous computing research prototypes have failed because they were neither future proof nor based on standard solutions. Therefore, the platform we envision must be based as much as possible on existing, preferably standard, solutions rather than experimental new solutions and leverage much of the already existing work in the research community.

Eventually, once the platform is comprehensive enough, additional layers of tooling may unlock it to less experienced developers or even end users. However, domain specific languages that are generally required for this have traditionally required a stable domain [3], [4]. Therefore we see this as a long term goal rather than something that is addressable now.

REFERENCES

[1] D. Cook and S. K. Das (eds). *Smart Environments: Technology, Protocols and Applications*. John Wiley, 2005.

[2] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. The gator tech smart house: a programmable pervasive space. *IEEE Computer*, 38(3):50 – 60, March 2005.

[3] Fowler M. Language workbenches: The killer-app for domain specific languages, 2005.

[4] Marjan Mernik, Jan Heering, and Anthony M. Sloane. When and how to develop domain-specific languages. *ACM Comput. Surv.*, 37(4):316–344, 2005.