SDL - the IoT Language

Edel Sherratt, Ileana Ober, Emmanuel Gaudin, Pau Fonseca I Casas, Finn Kristoffersen

SDL - the IoT Language

Edel Sherratt[1], Ileana Ober[2], Emmanuel Gaudin[3], Pau Fonseca i Casas[4], and Finn Kristoffersen[5]

[1] Aberystwyth University
[2] IRIT, Université Paul Sabatier
[3] PragmaDev SARL
[4] Universitat Politècnica de Catalunya
[5] Cinderella ApS

Abstract. Interconnected smart devices constitute a large and rapidly growing element of the contemporary Internet. A smart thing can be as simple as a web-enabled device that collects and transmits sensor data to a repository for analysis, or as complex as a web-enabled system to monitor and manage a smart home. Smart things present marvellous opportunities, but when they participate in complex systems, they challenge our ability to manage risk and ensure reliability.
SDL, the ITU Standard Specification and Description Language, provides many advantages for modelling and simulating communicating agents – such as smart things – before they are deployed. The potential for SDL to enhance reliability and safety is explored with respect to existing smart things below.
But SDL must advance if it is to become the language of choice for developing the next generation of smart things. In particular, it must target emerging IoT platforms, it must support simulation of interactions between pre-existing smart things and new smart things, and it must facilitate deployment of large numbers of similar things. Moreover, awareness of the potential benefits of SDL must be raised if those benefits are to be realized in the current and future Internet of Things.

1 Introduction

Smart things are everywhere, and new smart things are being created and deployed all the time. Together, they form the Internet of Things (IoT), defined by Atzori and others as *'a collection of things that are able to interact with each other and cooperate with their neighbours to reach common goals'* [1]. Smart things present wonderful opportunities, but when they participate in the complex system that is the Internet of Things, they challenge our ability to manage risk and ensure reliability.

Some smart things from the contemporary Internet of Things are described below. The potential for SDL to improve processes for developing things like these is explored, and recommendations are made for more closely aligning SDL [2] with the professional engineering practices and processes that will ensure the quality of the next generation of smart things. This is likely to be of

interest both to those seeking a viable technology that will serve as a backbone for communication within the IoT, and also to the established SDL community to raise awareness and stimulate discussion about future evolution of SDL to meet the needs of the people who are already bringing the IoT into being.

2 SDL and the IoT

The Internet of Things (IoT) is real, current and vulnerable. Novelty and complexity are essential characteristics of the IoT, but novelty and complexity also present direct challenges to reliability and security [3]. However, although the IoT is complex, individual smart things need not be complex, and are often developed by hobbyists or children. That is, even if a new smart thing or network of smart things is not very complex in itself, complexity explodes when new things are deployed amongst all the other things that populate the IoT. Moreover, the behaviour of a new thing in the IoT might well be affected by the pre-existing IoT population, which is likely to include smart things whose behaviour is unpredictable or hostile or both.

Although the challenges of novelty and complexity are difficult, they are not insurmountable. According to the Royal Academy of Engineering and The British Computer Society, the global communications backbone has very high availability – approximately two hours total system downtime in 40 years availability [3]. Much of that success can be attributed to mature engineering processes for adding new devices, protocols etc. to the telecommunications network. Modelling and simulation are central to those processes, as is adherence to standards such as the ITU Z.100 family of standards [2].

A few of the many smart things to be found in the contemporary IoT are briefly considered below, and the potential for SDL to enhance their benefits and limit their vulnerabilities is considered.

2.1 National Plant Phenomics Centre

EPPN, the European Plant Phenotyping Network, is an EU project that offers access to 23 different plant phenotyping facilities at seven different institutions in five countries. One of these seven institutions is the National Plant Phenomics centre in Aberystwyth[6], which provides access to phenotyping platforms that support research involving computer scientists, engineers and biologists to address how genetics and environment bring about the physical characteristics that constitute plant phenotypes. Current projects supported at the centre include predicting responses of food and fuel crops to future climates and development of robust standards for phenotyping experiments and environmental modelling.

At the heart of the National Plant Phenomics centre is a robotic greenhouse that enables controlled temperature, watering and nutrient regimes. Plants are automatically transported through the greenhouse on a conveyor system comprising over 800 individually RFID tagged carriages, and through five imaging

[6] http://www.plant-phenomics.ac.uk/en/

chambers that allow different kinds of images – normal digital photographs, infrared images and fluorescence – to be obtained for further analysis.

This environment collects data that can be accessed via the Internet, but as most of its communications are internal, its vulnerability is limited.

SDL could have been used to good effect to model communications within the robotic greenhouse, thus limiting the potential for error within the system. It could also be used to model how data is made available for external processing, and to simulate – and prevent – situations that might lead to data loss or corruption.

2.2 Walls have Ears!

Internet enabled television sets are a common consumer device, providing access to more content than was available with the previous generation of TV sets. However, with greater interactivity comes greater vulnerability.

For example, concern was raised in *The Daily Beast* [7] regarding privacy of data captured by the voice control facilities available on some Samsung smart TV sets.

In a similar vein, *My friend Cayla* [8] is a doll that provides enhanced interaction by connecting with the public Internet. No data captured by Cayla is currently retained, and no analytics carried out, but nonetheless, the potential to capture and analyse such data is present.

Had SDL been used in the development of these smart things, the potential for data leakage could have been made explicit by simulation and could then have been controlled.

2.3 Field Robotics

In the realm of Field Robotics, sensors are mounted on an autonomous robot that can navigate and survive in an unconstrained environment – sea, land, air, extra-terrestrial – where data is collected and transmitted to a base for further analysis. In other words, it is a typical smart thing that collects and uploads data for analysis and combination with other data to provide useful information.

Challenges include power management, securing the robot and its sensors against potentially hostile physical environments, and ensuring that both hardware and embedded software function reliably and correctly, as illustrated, for example by a glacier-surveying remote-controlled sailing robot [4].

SDL could have supported the development of communications for controlling the boat, and for collecting the sensor data. The effects of physical problems on communications could also have been modelled and simulated using SDL.

Similar applications of SDL include the communications infrastructure of smart cities described in the keynote address [5], and a smart bicycle, for use

[7] http://www.thedailybeast.com/articles/2015/02/05/your-samsung-smarttv-is-spying-on-you-basically.html

[8] http://myfriendcayla.co.uk/

in training, that was developed using SDL [6]. These applications illustrate the benefits of simulation and automated code generation in the development of reliable smart things, as well as the versatility and usefulness of SDL.

SDL has also been used to model and interact with a factory in a virtual reality scenario. This virtual representation is particularly interesting since the IoT will use Virtual Reality to interact with all the smart devices [7].

However, SDL does not directly support the physical engineering processes that result in physical devices, and a complete process involving design and fabrication of physical things as well as programming their behaviour would require fusion of SDL with appropriate support for computer aided design and manufacturing.

2.4 Smart Living

Smart homes promise many things, from intelligent control of lighting and heating, through smart appliances to ambient support for people with disabilities, including age-related disabilities. But this promise is not without corresponding threats. Smart appliances and power management systems threaten privacy, and their failure or unexpected behaviour could lead to the kind of dystopian vision presented by Hecht [8].

In this section we review the Smart Home concept and how SDL can help to provide a holistic solution that encompasses the many different facets and components of the problem.

When we think about a Smart Home, we envisage a house where we can control almost everything, from the windows, to the doors, to the bathroom; or perhaps we think of the various cameras that allow us to see what happens inside or outside our home, with all the legal issues this implies. However, a Smart Home cannot be understood in isolation from the question of sustainability. Usually sustainability is limited to immediate environmental considerations, but, with present-day availability of information, a broader view of sustainability becomes possible. For example, we can take into consideration the origin of the different materials and the environmental and social implications of different construction solutions when we develop a building (or an urban area). This helps us to address not only the environmental aspect of sustainability, but also the social and the economic impacts. Moreover, interconnection of different smart devices enhances the possibility to observe the social effects of urban design with a view to improving future designs.

Since sustainability is one of the main challenges for legislators, ever more clear and specific rules are being applied to new architectural designs. For example, 2010/31/EU European Directive on the Energy Performance of Buildings (EPBD)[9] aims to speed up energy saving policies in the building sector in order to achieve a 20% reduction of energy consumption in the European Union.

[9] http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1434227774810&uri=CELEX:32010L0031

Among many other measures, Article 9 of the directive stipulates that from December 31, 2020 new buildings must be nearly zero in energy consumption, and from December 31, 2018 for occupied buildings and/or public property buildings. In relation to this measure, the board recommends that Member States establish intermediate objectives in 2015 and gradually adopt the goals until 2020 to ensure compliance with the objectives set.

Regulation also applies to renovation and refurbishment of buildings, where all methods must be based on a cost benefit analysis, in order to achieve optimal levels of profitability and sustainability.

The energy that we can use in a building, the origins of this energy and the impacts on the society must all be considered in a Smart Home. In order to obtain this information, a Smart Home includes many devices that enable gathering of a huge amount of information. Differences between devices, methods and protocols make reuse of the devices, protocols and even information difficult. But since reuse is one of the key aspects of sustainability, and sustainability is a key aspect of a Smart Home, this is an issue that must be addressed.

Monitoring the building On one hand, from the point of view of the use of the building, or on the other, from the point of view of the design or refurbishment of the house, the interconnection of different hardware and software makes it difficult to extend and reuse the solutions. Now many different alternatives currently exist; some of the current alternatives that exist of the communication protocols used in this area are:

1. KNX [9], standardized on EN 50090, ISO/IEC 14543. KNX is an OSI-based network communications protocol mainly intended for the continuous monitoring of buildings. KNX was created from the convergence of three previous standards: the European Home Systems Protocol (EHS), BatiBUS, and the European Installation Bus (EIB or Instabus).
2. BACnet [10] is a communications protocol for building automation and control networks. It is an ASHRAE, ANSI, and ISO 16484-5 standard protocol.
3. Modbus [11] is a serial communications protocol originally published by Modicon (now Schneider Electric) in 1979 for use with its programmable logic controllers (PLCs). Simple and robust, it has since become a de facto standard communication protocol, and it is now a commonly available means of connecting industrial electronic devices
4. There are many other alternatives such as LonWorks, Home Automation, BACnet, DOLLx8, EnOcean INSTEON, Z-Wave, Intelligent building, Lighting control system, OpenTherm, Room automation, Smart Environments, Touch panel.

These protocols are used on monitoring devices provided by various different manufacturers; for example, Schneider Electric, elvaco, Carlo Gava, Panasonic, among many others. This represents a huge disparity of elements that must be considered in the needed integration of information on a Smart Home. Also, and

due to the huge opportunities that the Smart Home concept offers to the companies, big corporations are now able to offer complete development platforms to integrate several devices, offering a complete solution for the IoT.

SDL for the Smart Home Because Smart Home technology makes use of different communication protocols, different devices and different development platforms, it is absolutely essential to create an abstraction layer that supports the definition and formalization of the multiple processes that must be taken into account in a smart home, or as an extension of this, in a smart city.

SDL can be used not only to define the communication mechanism between the different actors involved in the Smart Home processes, but also the modelling processes that help in the planning and refurbishment of the house. The Abstract State Machine semantics of SDL means that it can define both the communication architecture of the current Smart house elements, and also the complete behaviour and life cycle assessment (LCA) [12] between all the actors and elements that participate in the life of the building [13]. As a specification language targeted to the description of distributed systems, SDL is very suitable for defining communication elements at many levels of abstraction. This combination makes SDL a perfect language to define, in a holistic way, the main processes that govern the behaviour (both current and future) of a building or residential area.

While it is true that SDL does not directly address security in the sense of attacks on a system of communicating agents, it is perfectly possible to model malicious as well as desired agents in a system, and to simulate the behaviour of desired agents in the presence of malign agents. SDL models involving multiple environmental agents could be used to model threats without cluttering the model of the desired system.

Moreover, the unambiguous nature of the language, its graphical syntax and the clear semantics, make SDL a strong candidate to work in heterogeneous environments, with multidisciplinary teams that want to define a holistic view of a Smart Home.

3 Three Challenges

3.1 Making a smart thing is easy - ensuring it behaves reliably in the wild is not

As a whole, the Internet of Things is characterised by novelty and complexity – factors well known to challenge physical as well as software engineering [3].

However, individual smart things can be created relatively easily by children or young teenagers who are familiar with Minecraft[10], if they have access to a consumer-grade 3-d printer and some basic components. Even without 3-d printing, smart wearables based on Adafruit or Lilypad components are fun to make and use.

[10] https://minecraft.net/

But this means that making safety- or business-critical smart things entails taking account of the other smart things that occupy the environment into which the new things must fit. In other words, the emergence of additive manufacturing and its domestic 3-d printing counterpart, together with the ready availabilty of microcontrollers, single-board computers and other components, has made it possible for any individual or organization to design, fabricate and program new kinds of smart thing, leading to an environment of interconnected smart devices some of which are likely to be badly-constructed or even malicious.

For these reasons, new communicating smart devices are threatened not only by inherent flaws in their own design, but also by flaws in the design of all the other devices that inhabit the Internet of Things. Modelling and simulation with SDL would help expose vulnerabilities before new devices are deployed and so improve security as well as reliability.

So, as we come to depend on ambient interconnected devices, better engineering processes and practices will be essential. SDL supports excellent engineering practices and processes by providing a precise way to specify communications, by enabling automated testing of distributed systems, and by supporting simulation before deployment.

3.2 New IoT Platforms

One of the major events that occurred recently in the IoT world, was the announcement of Brillo and Weave at the 2015 Google I/O conference [14]. Brillo[11], an Android derivate that covers just the lower levels, aims to become the underlying operating system for the Internet of Things, with a developer preview coming in the third quarter of this year.

Brillo represents an alternative to Microsoft's Windows 10, an OS that can be executed virtually on any device (from servers to RaspberryPi devices[12]. Windows 10, viewed as an IoT OS, offers the possibility to connect many different kinds of device such as may be found in a Smart Home.

It is clear that these will not be the only platforms to develop infrastructures for IoT. Canonical is taking a direction similar to that of Microsoft, announcing that in 2016 Unity8 will be integrated in Ubuntu desktop[13].

This brings into focus a trend in IoT platforms war, where two major kinds of IoT related technology are crystallizing.

One one side there are IoT platforms supported by large industrial players such as the emerging Brillo-Weave platform supported by Google and the Intel® IoT. These platforms are addressed to third parties, yet their underlying technology is controlled by a single industrial actor.

On the other side, there are open IoT platforms. The emergence of such platforms is supported by public funding (see H2020 - ICT 30) and facilitated by

[11] https://developers.google.com/brillo/

[12] https://www.raspberrypi.org/windows-10-for-iot/

[13] http://news.softpedia.com/news/Canonical-Details-Plans-for-Unity-8-Integration-in-Ubuntu-Desktop-462117.shtml

the existence of standardised technologies, such as Machine-to-Machine communications (M2M) and Network Functions Virtualisation (NFV) standardised by the European Telecommunications Standards Institute.

Given the opaque nature of the industrial offerings and their projected dissemination, we suggest that efforts towards building open IoT platforms should be supported. Let us consider how SDL fits with respect to the above mentioned standards.

The main M2M assets are

- Unified access to data from any application domain.
- Management of Privacy (Access Rights) adapted to the Application needs.
- Management of security levels adapted to the Application needs.
- Suited for IP networks, yet agnostic to underlying communication technologies.

Although not primarily targeted to the IoT, Network Functions Virtualisation (NFV) is a standard relevant to the IoT. Among its principles the more pertinent with regard to the IoT seem to be

- The need for Inter-Domain Interfaces
- Generic functional blocks
- Multiplicity, Composition, and Decomposition

Originally designed to model telecom systems, SDL has also proved successful in addressing the needs of other application domains [5]. SDL can already support most of the concepts listed above; for example, SDL can deal with data management and with interfaces. However, it would need extensions for dealing with security and it would need adjustments to improve handling of multiplicity and dynamism.

The following section looks in more detail at the challenges of deployment, in particular the challenge of multiplicity, and describes how SDL rises to those challenges.

3.3 Deployment issues

One of the main characteristics of IoT systems is that one of the sub-systems is usually instantiated a huge number of times. For example in a Smart Grid system the meters are sub-systems that are instantiated many times. Literally millions of them are deployed and any problem discovered on site is extremely costly for the operator to solve. Even though it is not always exactly the same sub-system that is instantiated, most of the main characteristics of the meters are the same. In the Smart Grid example, it could be that all meters will transmit the electrical consumption, some meters might do it every day, where some others might do it every month. Another characteristic is that the different sub-systems are physically separated entities that need to communicate with each other. The means of communication between the different sub-system are numerous but they all have the characteristics of a telecommunication protocol.

SDL has from its inception enabled precise description of telecommunication protocols. Static interface description is covered by ASN.1 data types with encoding and decoding rules, and dynamic aspects by the behaviour description of each protocol layer. These static and dynamic elements of an SDL specification are perfectly adapted to the description of exchanges between the different sub-systems in an IoT system. By extension, the application on top of the telecommunication protocols can also be described from a functional point of view. Because SDL is an executable language it is possible to verify the behaviour of the overall IoT system with an SDL simulator independently of any type of implementation. Functional variants of the IoT sub-system can be dealt with using object orientation in SDL. The common behaviour is described in a super-class, and the possible variants are described in different sub-classes.

Building up a system combining a mix of the different variants or several instances of the same sub-system is then very straight forward. Based on this concept, Humboldt University zu Berlin [15] has developed a deployment simulator of an SDL-RT[14] system based on the ns3 network simulator. This work was initially done to simulate an early warning earthquake detection system. The system is composed of hundreds of sensors deployed geographically and the information from the sensors is gathered and analyzed in one hub. PragmaDev has integrated and extended this work to automatically simulate the deployment of SDL or SDL-RT systems on numerous nodes.

The generic SDL architecture is used to define the different sub-systems (Figure 1), an SDL-RT deployment diagram is used to define how many and where the nodes are deployed, and a csv file describes the scenario of events on each distributed node (Figure 2).

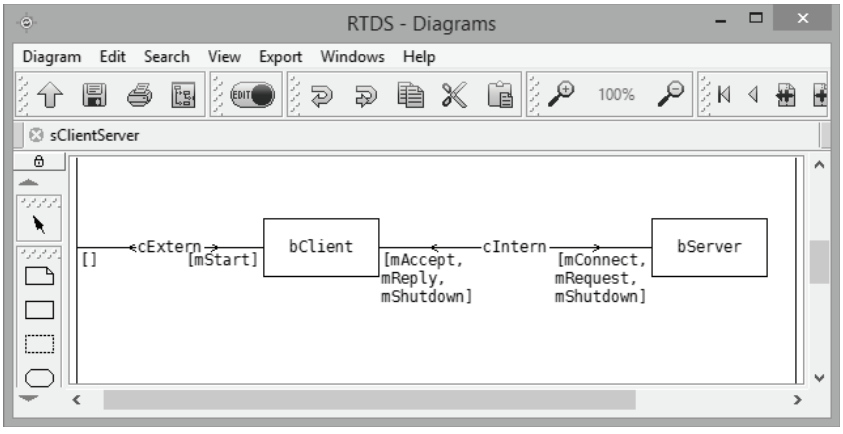

Fig. 1. The SDL architecture describes the different sub-systems: bClient and bServer

[14] http://www.sdl-rt.org

Load testing The deployment simulator described above and illustrated in Figure 3 is perfect from a functional point of view. But one of the most common IoT issues is limitation of the load that can be carried by the means of communication. If too many messages are sent at the same time it is quite possible that some or all of them get lost. These limitations can be described in the underlying network simulator but do not appear directly in the SDL model. It would be interesting and useful to be able to describe in the model a set of characteristics that integrate these limitations and to have the validation tools explore all the possible problems that might occur due to large numbers of deployed instances.

Interleaved test cases The deployment simulator is a very powerful tool that is perfect for the early warning earthquake application. But from a functional point this simulator uses a csv file that describes only one scenario for all the possible nodes. Only a unique sequence among the different nodes is described that way. Usually, before the deployment, a set of test cases is written to test a single node of one of the sub-systems. The issue is to verify that having several instances of this node does not imply alteration of its functionalities. Each test case should be able to pass on one node independently of the execution of another test case on another node. In theory this leads to verification of any combination of the different test cases work on the deployed system and that creates a combinatoric explosion due to the number of possible interleaving combinations.

Work is currently on-going to identify blocks of messages that create interactions between the instances and to run a reduced number of relevant interleaved combination of the test cases.

4 What is needed to make SDL the language of choice for IoT systems?

The examples presented in previous sections highlight typical aspects of IoT systems development, validation and deployment, and illustrate the benefits of SDL. The SDL model of communicating agents fits the IoT systems model, and the formal semantics of SDL, with its associated data modelling capabilities, enables comprehensive validation of IoT sub-systems functionality by simulation and testing. Deployment of IoT systems designed in SDL is also currently well supported.

- the SDL semantics for modelling, simulation and deployment reduces the likelihood of error in deployed systems and also exposes potential data leakage. The use of SDL would have improved the robotic greenhouse system, and the TV and doll applications, through simulation and testing, which would make the potential privacy issues explicit already in the design phase.
- SDL allows to model aspects of power management, insofar as modelling and simulation can expose wasteful communications and handling of unreliable data sensor as described in the field robotics examples. SDL also has the features necessary to specify and develop autonomous self-monitoring adaptive systems.

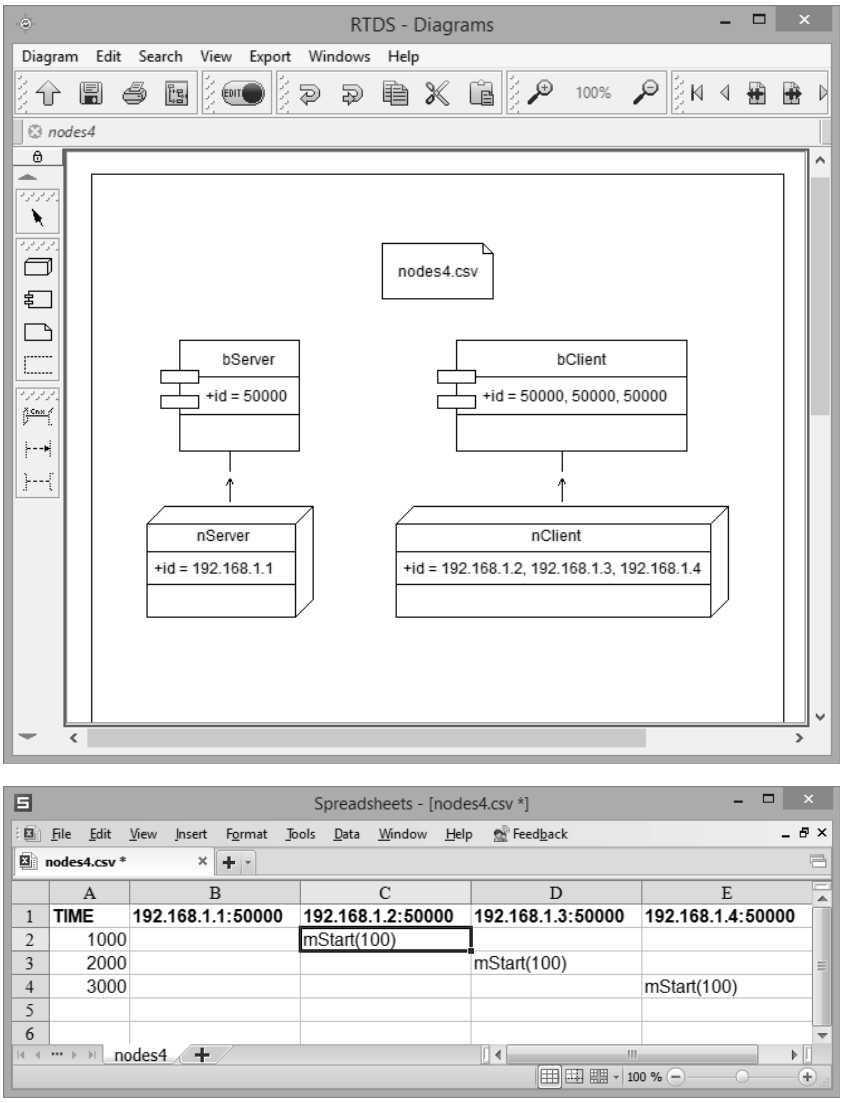

Fig. 2. The deployment diagram describes how many instances of each sub-system is deployed. One instance of bServer and three of bClient.

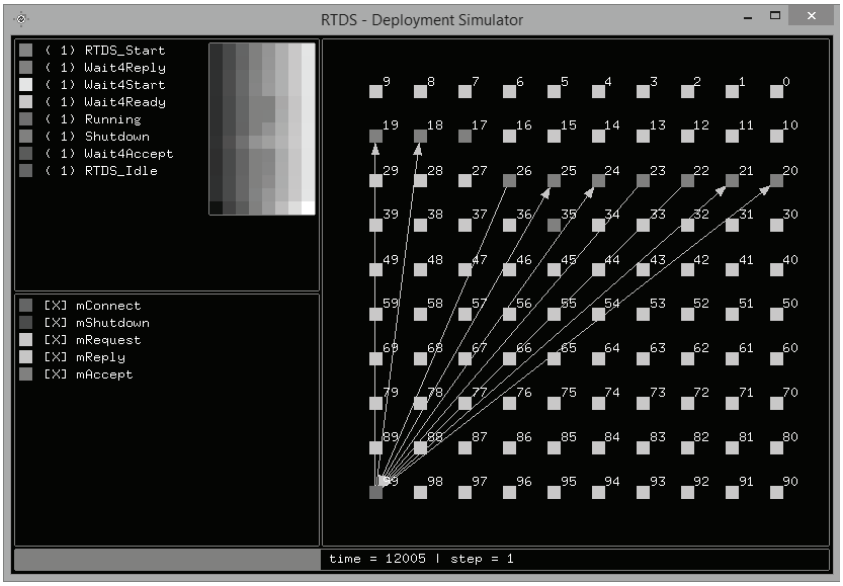

Fig. 3. Deployment simulator interface with live and post-mortem traces.

- SDL allows the system design at different levels of abstractions and hence is useful both for detailed system design of a Smart Home system using several protocols as well as a building maintenance management system. In addition SDL also supports design of IoT systems using more protocols to implement the overall functionality of the system.
- SDL supports development targeting emerging platforms and new devices. It has proved its worth in protocol development and is sufficiently flexible to allow targeting of new devices.
- SDL offers support for well-established good engineering practices, including model simulation, automated testing and deployment. This increases reliability and reduces vulnerability of deployed systems.
- SDL allows simulations that involve large numbers of smart devices, and so addresses the need for scalability.

However, the IoT systems considered above also serve to identify areas where SDL needs improvement in order to fully support the design and development of such systems

- SDL tools are unlikely to compete with the small environments used by hobbyists with arduinos etc., not because these environments provide simulation and modelling, but rather because safety and reliability are not high-priority requirements amongs these developers. So instead of expecting all creators of smart things to use SDL and to follow sound engineering practice, creators of

secure, reliable smart things should use SDL to reduce inherent vulnerability in the things they produce and to increase their resilience to external agents that pose a threat whether because of poor design or malicious intent.
- While SDL supports development targeting emerging platforms, new IoT platforms mean that if SDL is to be used successfully for IoT system development, tools must be developed that map SDL models to these platforms. In particular, SDL should be mapped to open and freely-available platforms.
- The impact of deployment of an IoT system in contexts with numerous of other systems making use of the same communication resources causing potential delays and loss of messages is supported in SDL only to a limited degree. SDL may be extended with features to specify possible loss and delay of signals based on the load on a communication path.
- SDL semantics enables modelling of an environment populated by multiple agents. This provision should be explored and developed to facilitate development of secure networks of smart things, both with respect to privacyof data communicated between parts of an IoT system and protection against attacks to access data or to affect the system functionality.
- An emerging trend in the IoT is to design and 'print' a new device using additive manufacturing (3-d printing) techniques with suitable CAD software. SDL does not interface directly with CAD systems. For the future, it would be good to extract models of the implied SDL agent from designs of new devices. SDL would allow such models could be replicated, simulated and studied in a Virtual Reality populated by other smart devices.

Apart from these technical considerations, there is also an urgent need to raise awareness of the potential benefits of SDL for IoT systems development. For example, SDL is not mentioned in the 2011 survey [16], though this does include ns-2, which was used in conjunction with SDL [17]. Also there is no mention of SDL in [18]. Unless this situation is changed, it is highly likely that much that is already well established SDL will be re-invented, delaying development of the professional practices that will form the core of any effort to create a safe, reliable IoT.

5 Conclusion

We have looked at a variety of elements of the emerging Internet of Things and identified the role of SDL in supporting the engineering practices that will enable creation and deployment of safety and business critical smart things. We have also identified the need to improve SDL's capacity to model the behaviour of smart things when communications channels are heavily loaded, either because many instances of things are deployed in a smart system, or because different systems must co-exist in a crowded IoT environment. Security and privacy are also not adequately expressible in SDL as it stands.

But perhaps the greatest challenge lies in raising awareness of SDL amongst the people who will create and deploy the next generation of smart things.

References

1. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: A survey. Computer Networks 54 27872805, Elsevier (2010)
2. ITU-T: Z.100 series for SDL 2010, International Telecommunications Union 2011-15
3. The Royal Academy of Engineering and The British Computer Society: The Challenges of Complex IT Projects, The report of a working group from The Royal Academy of Engineering and The British Computer Society The Royal Academy of Engineering (2004) Available at http://bcs.org/upload/pdf/complexity.pdf, accessed 4 June 2015
4. Neal, M., Blanchard, T., Hubbard, A., Chauché, N., Bates, R., Woodward, J.: A hardware proof of concept for a remote-controlled glacier-surveying boat, Journal of Field Robotics, Volume 29, Issue 6, Article first published online: March 2012, Wiley Periodicals, 2012
5. Fischer, J., Redlich, J.-P., Scheuermann, B., Schiller, J., Günes, M., Nagel, K., Wagner, P., Scheidgen, M., Zubow, A., Eveslage, I., Sombrutzki, R., Juraschek, F.: From Earthquake Detection to Traffic Surveillance – About Information and Communication Infrastructures for Smart Cities, in Haugen O., Reed, R., Gotzhein, R., (Eds.), System Analysis and Modelling: Theory and Practice, Lecture Notes in Computer Science, Volume 7744, Springer 2013
6. Kuhn, T., Gotzhein, R., Webel , C.: Model-Driven Development with SDL – Process, Tools, and Experiences, in O. Nierstrasz et Al. Eds., Model Driven Engineering Languages and Systems, Lecture Notes in Computer Science Volume 4199, Springer 2006
7. Fonseca Casas, P., Pi Paloms, X., Casanovas Garcia, J., Jov, J. (n.d.). Definition of virtual reality simulation models using specification and description language diagrams. In F.K., Toeroe, M., Gherbi, A., Reed, R. (Eds.) SDL 2013: Model Driven Dependability Engineering, pp 258–274.Lecture Notes in Computer Science, vol. 7916, Springer Berlin Heidelberg (2013) doi:10.1007/978-3-642-38911-5
8. Hecht, J: The Internet of **** things. Nature Physics, 2014 Jul, Vol.10(7), pp.538-538
9. KNX Association. (2014). System Specifications http://www.knx.org/en-us/knx/technology/specifications/index.php accessed 20 July 2015
10. ISO 16484-1:2010, Building automation and control systems (BACS) – Part 1: Project specification and implementation, ISO 2010
11. Pefhany, S. (2000). Modbus Protocol. Penthon Media Inc (Vol. 5).
12. Ortiz, O., Castells, F., Sonnemann, G., Sustainability in the construction industry: A review of recent developments based on LCA, Construction And Building Materials, 2009 Jan, Vol.23(1), pp.28-39
13. Fonseca i Casas, P., Fonseca i Casas, A., Garrido-Soriano, N., Casanovas, J. (2014). Formal simulation model to optimize building sustainability. Advances in Engineering Software, 69, 6274. doi:10.1016/j.advengsoft.2013.12.00
14. Reaper, D: Google IO 2015 Live conference keynote - Brillo & Weave, YouTube video, 11:08 minutes, 28 May 2015, Available at https://www.youtube.com/watch?v=-BSRSCPxiPg, accessed 4 June 2015
15. Brumbulli, M., Fischer, J.: Simulation Configuration Modeling of Distributed Communication Systems. In: Haugen, O., Reed, R., Gotzhein, R. (eds.) System Analysis and Modeling: Theory and Practice, Lecture Notes in Computer Science, vol. 7744, pp. 198-211. Springer Berlin Heidelberg (2013)

16. Baumgartner, T., Chatzigiannakis, I., Fekete, Sndor P., Fischer, S., Koninis, C., Kröller, A., Krger, D., Mylonas, G., Pfisterer, D.: Distributed algorithm engineering for networks of tiny artifacts, Computer Science Review, 2011, Vol.5(1), pp.85-102, Elsevier (Science Direct)
17. Kuhn, T., Geraldy, A., Gotzhein, R., Rothländer, F.: *ns*+SDL - The Network Simulator for SDL Systems, in Prinz., A, Reed, R., Reed, J. (Eds.) SDL 2005: Model Driven, Proc. 12th Int. SDL Forum, Grimstad, Norway, June 2005, Springer 2005
18. Chatzigiannakis, I., Mylonas, G., Vitaletti, A.: Urban pervasive applications: Challenges, scenarios and case studies , Computer Science Review, 2011, Vol.5(1), pp.103-118, Elsevier (SciVerse ScienceDirect Journals)