Foundations for Simulating IoT Control Mechanisms with a Chemical Analogy

Gabor Kecskemeti[1] and Zsolt Nemeth[1]

Institute for Computer Science and Control, Hungarian Academy of Sciences.
1111 Budapest, Kende u. 13-17, Hungary.
`kecskemeti.gabor@sztaki.mta.hu`

Abstract. The emergence of IoT systems introduced new kind of challenges for the designers of such large scale highly distributed systems. The sheer number of participating devices raises a crucial question: how they can be coordinated. Engineers often opt for using a simulator to evaluate new approaches or scenarios in various environments. This raises the second crucial question: how such a large system can be simulated efficiently. Existing simulators (even if they are IoT focused) are often focused on some particular scenarios and not capable to evaluate coordination approaches. In this paper we propose a chemical coordination model and a new extension to the DISSECT-CF cloud simulator. We expect that their combination on one hand ensures a distributed adaptive coordination on the other hand allows the separation of simulation problems into manageable sizes; these enable the analysis of large scale IoT systems with decentralized coordination approaches.

Key words: IaaS, Internet of Things, simulation, actuator, sensor, smart object, chemical coordination

1 Introduction

As the world gets more and more connected, the mere number of devices suggests that decentralized coordination approaches are going to be more prominent in the future. Compared to traditional distributed systems where computing resources are shared and coordinated to rapidly and efficiently reach a shared goal, internet of things (IoT) introduces new kind of resources (e.g, sensors and actuators) and goals [1]. These new resource types have remarkably different properties compared to past resource types. Such new properties demand alternative coordination mechanisms in order to enable the collaboration of these new resources with traditional distributed systems.

Smart objects (e.g., smart-phones, integrated computing facilities in actuators and sensors) in the IoT concept allow computation to be executed as near as possible to the sensing and actuating objects in the system. These smart objects are on the edge of the distributed computing infrastructure and they allow intelligent behavior of many localized IoT systems (e.g., smart homes) even without outsourcing computing activities (like behavior modeling and analysis, data mining of sensor data and decision making based on the results of the data mining

operation). To provide better and more resilient services it was identified that advanced computational intelligence was lacking in current IoT systems [2–4].

This paper introduces a nature inspired, chemical analogy for decentralized coordination of IoT systems. Unfortunately, the nascent state of IoT systems does not allow the real life experimentation for identification of concerns and possible improvements. To overcome this issue this paper also introduces the design of an extension of the DISSECT-CF cloud and distributed systems simulator [5] which would allow its users to freely alter the rules and axioms of the chemical analogy. We expect that utilizing this extension will enable new IoT scenarios to be rapidly evaluated in such large scale distributed environments that was previously not possible.

The next sections of the paper are organized as follows: first, in Section 2, we introduce the related works that already have some success of introducing simulation techniques for IoT systems. Next, in Section 3, we show how can a chemical analogy help us improve the coordination and computational intelligence in IoT systems. The section also provides an overview of the chemical computing models and the ways one can customize such models to better fit to new previously unexplored scenarios. Finally, Section 4 shows how the chemical model can be applied in an advanced simulation toolkit (i.e. DISSECT-CF). The section also provides an overview on the necessary extensions for the simulator and the ways the simulation can be distributed to better model the non-centralized system view available most IoT systems.

2 Related Works

In [6], I. Moschakis and H. Karatza, introduces several simulation concepts for IoT systems. First, they show how the interfacing between the various cloud providers and IoT systems could be modeled in a simulation. Next, they provide a novel model for IoT related workloads where data is processed from sensors taking part in the IoT system. Unfortunately, their work do not consider actuators, and mainly discusses the behavior of cloud systems that support the processing of data originated from the IoT system. In contrast, the our simulation concepts put more emphasis on how sensors actuators and smart objects on the edges of clouds would interface with each other and with the cloud systems.

[7] introduces SimIoT, which is derived from the SimIC simulation framework [8]. SimIoT provides a deeper insight into the behavior of IoT systems, but still restrains itself to the basic sensor-data processing scenario. Compared to the previously introduced work by Moschakis et al., this simulator introduces several techniques (e.g. a broker) that simulates the communication between an IoT sensor and the cloud. Alongside, SimIoT the authors introduce a new – more interactive – user concept as well which is significantly different from the widely accepted big data processing model. Compared to this work, SimIoT is limited by its compute oriented activity modeling.

While the previous simulation approaches were focusing on larger scale systems, [9] addresses IoT simulation from a completely different perspective. Sim-

ulations in that paper are visual, allow the observation of the behavior of the simulated system in real time and are more similar to industrial modeling tools than the previously mentioned simulators. Such simulators allow precise modeling of actuator and smart object operations depending on the sensor's output – because of their detailed models such simulations are more useful for smaller systems (like evaluating IoT behavior in a factory or allowing student experiments with self-built systems). In this paper, we aim at introducing a simulation framework that allows to incorporate as much detail as possible while it still allows to model the size of amount of resources necessary for sufficient evaluation of new IoT coordination models based on the chemical analogy.

Amongst recent works, [10] is one of the few which deals with the dynamic nature of IoT systems namely, it investigates fault behaviors and introduces a fault model for such systems. Although faults are important part of dependability and IoT, the scalability of the introduced fault behaviors and concepts (e.g., the applied use case scenario in their paper is home automation just like in [11]) are not sufficient for modeling large enough systems that would benefit from the decentralized control mechanisms we envisioned in this paper.

Finally, [11, 12] both introduce novel infrastructure coordination techniques that support the use of larger scale IoT systems. In both cases the authors evaluate their approaches with a simulation framework. In [11], CloudSim [13] is used to model a community cloud based on residential infrastructures. On top of CloudSim the authors provide customizations that are tailored for their specific home automation scenarios and therefore limit the applicability of their extensions for evaluating new IoT coordination approaches. These papers are also limited on sensors/smart objects thus not allowing to evaluate a wide range of IoT scenarios that are expected to rise to widespread use in the near future [1].

With respect to the chemical model proposed in this paper, the γ-family (the chemical calculus and the related languages) has already been investigated in various scenarios, like self-organizing systems [14] where a self-healing, self-optimizing and self-protecting mail system is studied; enacting workflows on-the-fly with strong emphasis on dynamicity [15], modeling self-developing secure virtual organizations [16] and dynamic service composition [17] and many others.

3 Chemical Analogy for IoT Control

A brief introduction to chemical computing. Most algorithms are described as a series of computational steps, operations applied to data elements and computing is governed by a control thread. The chemical paradigm radically obliterates this notion rather, a program is conceived as a chemical solution where data and procedures are molecules floating around and computation is a series of reactions between these molecules. This vision of chemical computing is formalized in the γ-calculus [18] as (without the chemical guise) a declarative functional computational model where terms are commutative and associative.

The fundamental data structure in the γ-calculus [18] is the multiset that are affected by so called reactions taking place independently and potentially simul-

taneously, according to local and actual conditions yielding a multiset rewriting system. There is no centralized control, ordering, serialization, rather the computation is carried out in an indeterministic, inherently parallel, self-evolving way. Molecules, i.e. γ-terms are variables, γ-abstractions (procedure definitions) and solutions (separated by membranes). Juxtaposition of γ-terms is commutative and associative realizing the 'Brownian-motion', the free distribution and unspecified reaction order among molecules that is a basic principle in the chemical paradigm. γ-abstractions are the reactive molecules that can take other molecules and transform them by reduction; molecules of a reaction are extracted by pattern matching. The semantics of a γ-reduction is very similar to that of λ-calculus. Reactions may depend on certain conditions and can also capture multiple molecules in a single atomic step. Besides the associativity and commutativity, reactions are governed by: (i) law of locality, i.e. if a reaction can occur, it will occur in the same way irrespectively to the environment; and (ii) membrane law, i.e. reactions can occur in nested solutions, solutions may contain sub-solutions separated by a membrane. The γ-calculus is a *higher order* model, where abstractions – just like any other molecules – can be passed as parameters or yielded as a result of a reduction. The Higher Order Chemical Language (HOCL) [18] is a language realized on the principles of the γ-calculus.

The concept of IoT coordination based on the chemical computing model. An IoT scenario can be seen as a large set of various entities, notably sensors, actuators and smart objects. Sensors and actuators are the gateways between the physical and the virtual world: sensors provide data such as temperature or speed whereas actuators realize some physical transformations according to decisions such as increase heating or apply brakes. Smart objects furthermore, have the capability to compute and communicate. There are some goals predefined to be fulfilled and can be achieved by the interaction of these entities – we are aimed at providing a nature inspired coordination mechanism to facilitate these interactions.

To create such a model we set up the following assumptions and requirements. (i) entities are independent from each other and can perform their activities concurrently. (ii) entities do not have any predefined pattern for their interaction other than their goal and the conditions how to reach it (iii) entities may change their functional and non-functional characteristics of their services any time, even abruptly. (iv) entities can select the cooperating entities (either as consumer or provider) at their discretion. (v) the state of the entity can be changed by their interacting parties only and unaffected by others. (vi) entities must accommodate to local and actual conditions.

To facilitate the flexible and adaptive yet predictable interaction of entities, we envision a chemical metaphor where entities correspond to molecules dissolved in a solution. Molecules also represent data or intermediate control information provided by the entities. Molecules thus, can move freely which, on one hand models their mobility, on the other hand the Brownian motion ensures engaging in reactions in their vicinity. A molecule has its chemical properties that determine which other molecules can react with and in what conditions – this corresponds to the conditions an entity must fulfill before it can perform
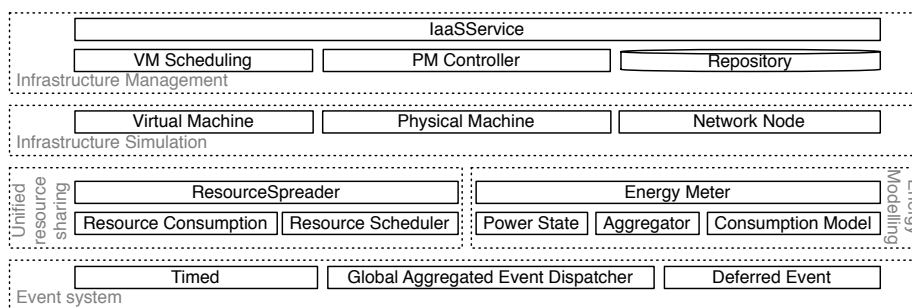
Fig. 1. The architecture of the DISSECT-CF simulator

its activities. The series of reactions transform the molecules and reactions are possible until a molecule becomes inert. At this point the entity modeled by the molecule namely, an actuator or a smart object is triggered accordingly. Each reaction transforms the molecules closer to their inert state yet, there is no pre-defined order or governing of reactions other than the chemical properties of the molecules. Entities could be self-evolving towards a state when they are enabled to perform their tasks, constantly accommodating to local and actual conditions.

In this very simple vision we may express the followings. (i) Molecules precisely define the entity and also define the conditions that must be fulfilled before the goal can be reached. These conditions determine the matching molecules and the types of reactions that can transform the molecule towards its inert state. Some molecules may act as catalysts: they must be present in the reaction (enabling a transformation) but remain intact afterwards. (ii) There can be global conditions that must be kept satisfied, they can model certain boundaries of the IoT environment such as overall energy consumption. (iii) Sensors and smart objects insert molecules into the virtual solution, this way the physical world is converted into the virtual chemical world where the coordination takes place. (iv) In contrast, an inert molecule triggers an action in the physical world via smart objects or actuators. Note, that the chemical model is not applied for problem solving but solely *coordinates* the execution. Reaching an inert state is an exit point form the coordination level. (v) Reactions are affected by local and actual conditions. Since changes in the world are represented in the virtual chemical solution, the coordination mechanism adapts to the changing conditions.

4 Simulation Approach towards the Chemical Analogy

4.1 The DISSECT-CF Cloud Simulator

DISSECT-CF [5, 19] is a compact, highly customizable open source[1] cloud simulator with special focus on the internal organization and behavior of IaaS systems. Figure 1 presents its architecture (as of version 0.9.6). The figure groups

---

[1] available from: https://github.com/kecskemeti/dissect-cf

the major components with dashed lines into subsystems. Each subsystem is implemented as independently from the others as possible. There are five major subsystems each responsible for a particular aspect of internal IaaS functionality: (i) event system – for a primary time reference; (ii) unified resource sharing – to resolve low-level resource bottleneck situations; (iii) energy modeling – for the analysis of energy-usage patterns of individual resources (e.g., network links, CPUs) or their aggregations; (iv) infrastructure simulation – to model physical and virtual machines as well as networked entities; and finally (v) infrastructure management – to provide a real life cloud like API and encapsulate cloud level scheduling (the target of improvements presented here).

4.2 New IoT Related Simulation Constructs

As we aim at supporting the simulation of hundreds of millions of devices participating in previously unforeseen IoT scenarios, the high performance of the simulator's resource sharing mechanism is essential therefore, we have chosen to use the DISSECT-CF simulator because of its unified resource sharing foundation. Building on this foundation it is possible to implement the basic constructs of IoT systems (e.g., smart objects, sensors or actuators) and keep the performance of the past simulator. In the following, we list the four main entities in our extension and we show how they participate in an IoT simulation:

– *Sensors.* These passive entities in the simulated system have four major features: (i) their performance is limited by their connectivity and maximum update frequency; (ii) they can provide historical data (preloaded from a file and offered based on the time and situation of the sensor query); (iii) they can offer dynamic data based on user provided custom algorithms that alter the sensor's reported metrics; and finally (iv) in some cases replicating data from sensors of the real world would help to steer the rest of the simulator's sensors (e.g., a few sensors could bridge with the real world).
– *Actuators.* Although DISSECT-CF is aimed at simulating the properties of computing systems both sensors and actuators could extend these properties further. Therefore, the simulator allows its users to integrate with third party libraries and also allows the dynamic sensor behavior to be updated via the actuator interface (i.e. acting like some of the physical properties around the sensor really changed). Similarly to sensors, these interfaces are limited by their network connectivity and reaction time (e.g., how long does it take to actually perform an actuation action). Actuators also have the unique feature that allow the changing of the locations of others (except the central services).
– *Smart objects* are simulated as localized and significantly limited computing and storage resources that might be coupled with some sensors and actuators and that might perform regular activities arriving from their coupled counterparts. With these objects local goals of the particular IoT system can often be achieved. Also the chemical analogy might coordinate several smart objects in the vicinity to work together in order to achieve the local goals without turning to the central services. These coordination steps might be needed for long delays experienced with central services or because privacy concerns, etc.
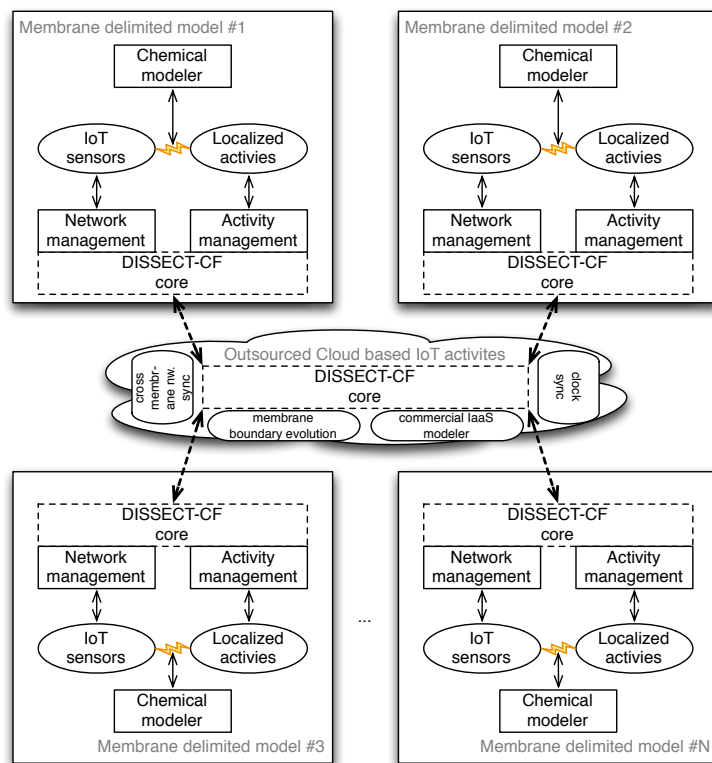
Fig. 2. Multi host IoT simulation for localized chemical solutions and centralized outsourcing capabilities

– *Central computing services.* These computing services provide the large scale background processing and storage capabilities some big data applications building on the other actors might need. In the chemical analogy, these services are expected to be used only if unavoidable.

In all cases, dynamic location information is offered with the actors. Allowing one of the most crucial information and behavior (i.e., the brownian motion and membrane formation) to be simulated for the chemical analogy. Failures of the actors are also modeled on two levels: location based resource availability (e.g., an action cannot be taken because the actuator cannot reach the particular location) and physical device degradation (e.g., an actuator is no longer capable to accomplish its tasks). Users are free to provide failure models for both cases.

4.3 The Role of The Chemical Model in IoT simulations

As we aim at simulating millions of sensors/actuators/smart objects, it can be foreseen that the simulations cannot be run on a single host (e.g. mainly because of memory constraints – e.g., a single sensor might consume only a dozen or

so bytes of memory but modeling messages and activities between them could introduce several orders of magnitude more memory requirements). On the other hand, the chemical model has both the locality principle and the separation of the solutions by a membrane. Therefore, we propose to divide the DISSECT-CF simulations into simulation of localities or sub-solutions – this separation is shown in Figure 2 with separate boxes per membrane (i.e., per simulator host). These membranes on one hand may be permeable and allow the molecules to move across the boundary, on the other hande they significantly limit the number of actors to be simulated at a time.

Although in most cases the time in the various simulations can be kept diverging (because of the lack of chemical control across membrane boundaries), during the contact of the central computing services a clock synchronization must occur between the various simulation cores. This clock synchronization ensures the happened before relations of the various simulation events can be ensured even across membrane boundaries. The clock synchronization mechanism is also ensuring that simulations of network communications between the membranes are possible and incoming/outgoing network links from/to membranes are ensured to be working according to the resource sharing rules in the simulator by the new cross membrane network synchronization component.

Next, the original main simulator functionalities are executed on top of the central DISSECT-CF core. Within this simulation we evaluate the big data processing and storage operations. Here the user of the simulator can define the large scale computing operations as well as the capabilities of the centralized infrastructure. In case of commercial IoT applications this centralized infrastructure could be simulated as a private cloud hosted on the company's premises.

Finally, inside the membranes operations of actors in the simulation will take place as coordinated by the chemical modeler. This component guarantees the Brownian motion based quasi-random activity to sensor/actuator mapping. Thus, the chemical modeler will be the main user of the membrane level DISSECT-CF core components by instructing the actors for their tasks and ensuring that localized goals within the membrane are met according to the chemical ruleset (e.g. possible reactions, available catalysts). The simulator allows the customization of this ruleset thus, allows the evaluation of multiple organization strategies even at the lowest level of the simulations.

5 Conclusions

In this paper we have introduced a new simulation technique for IoT systems. This technique has built on two existing concepts and solutions: (*i*) coordinating distributed systems through a chemical analogy and (*ii*) simulating centralized cloud components and low level IoT concepts with the help of the unified resource sharing foundation of the DISSECT-CF simulator. The chemical model allows the separation of actors in a controlled way and keeping the size of each such partition (sub-solution separated by membranes) proper for efficient simulation. We expect that with the help of these DISSECT-CF extensions future research

will be capable of analyze new decentralized coordination approaches in IoT systems. In our future works, we plan to further investigate the possible chemical to IoT entity mappings and the ways they can alter our thinking about IoT systems and their coordination.

Acknowledgements

References

1. Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.
2. Chun-Wei Tsai, Chin-Feng Lai, and Athanasios V Vasilakos. Future internet of things: open issues and challenges. *Wireless Networks*, 20(8):2201–2217, 2014.
3. Luciano Baresi, Luca Mottola, and Schahram Dustdar. Building software for the internet of things. *IEEE Internet Computing*, (2):6–8, 2015.
4. Helen D. Karatza and Constandinos X. Mavromoustakis. Special issue on simulation-based performance evaluation of infrastructures for the internet of things: Connectivity and resource considerations in the mobility era. *Simulation Modelling Practice and Theory*, 34:157–158, 2013.
5. Gabor Kecskemeti. DISSECT-CF: a simulator to foster energy-aware scheduling in infrastructure clouds. *Simulation Modelling Practice and Theory*, pages 1–31, September 2015 *to appear*.
6. Ioannis A. Moschakis and Helen D. Karatza. Towards scheduling for internet-of-things applications on clouds: a simulated annealing approach. *Concurrency and Computation: Practice and Experience*, 27(8):1886–1899, 2015.
7. Stelios Sotiriadis, Nik Bessis, Eleana Asimakopoulou, and Navonil Mustafee. Towards simulating the internet of things. In *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on*, pages 444–448. IEEE, 2014.
8. Stelios Sotiriadis, Nik Bessis, Nikos Antonopoulos, and Ashiq Anjum. Simic: Designing a new inter-cloud simulation platform for integrating large-scale resource management. In *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*, pages 90–97. IEEE, 2013.
9. Dominique Dhoutaut, Benoît Piranda, and Julien Bourgeois. Efficient simulation of distributed sensing and control environments. In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, pages 452–459. IEEE, 2013.

10. Ivanovitch Silva, Rafael Leandro, Daniel Macedo, and Luiz Affonso Guedes. A dependability evaluation tool for the internet of things. *Computers & Electrical Engineering*, 39(7):2005–2018, 2013.
11. Adnan M Khan, Leandro Navarro, Leila Sharifi, and Luís Veiga. Clouds of small things: Provisioning infrastructure-as-a-service from within community networks. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on*, pages 16–21. IEEE, 2013.
12. Bogdan Manate, Teodor-Florin Fortis, and Viorel Negru. Optimizing cloud resources allocation for an internet of things architecture. *Scalable Computing: Practice and Experience*, 15(4), 2015.
13. Rodrigo N Calheiros, Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1):23–50, January 2011.
14. Jean-Pierre Banâtre, Pascal Fradet, and Yann Radenac. Programming self-organizing systems with the higher-order chemical language. *International Journal of Unconventional Computing*, 3(3):161–177, 2007.
15. Manuel Caeiro, Zsolt Németh, and Thierry Priol. A chemical model for dynamic workflow coordination. In *PDP*, pages 215–222, 2011.
16. Alvaro E. Arenas, Jean-Pierre Banâtre, and Thierry Priol. Developing autonomic and secure virtual organisations with chemical programming. In *SSS*, pages 75–89, 2009.
17. Jean-Pierre Banâtre and Thierry Priol. Chemical programming of future service-oriented architectures. *JSW*, 4(7):738–746, 2009.
18. Jean-Pierre Banâtre, Pascal Fradet, and Yann Radenac. Generalised multisets for chemical programming. *Math. Struct. in Comp. Science*, 16:557–580, 2006.
19. Gabor Kecskemeti, Simon Ostermann, and Radu Prodan. Fostering energy-awareness in simulations behind scientific workflow management systems. In *The 7th IEEE/ACM International Conference on Utility and Cloud Computing (UCC)*, pages 29–38, London, December 2014. IEEE Computer Society.