A Distributed Controller for Organic Computing Applications

Stefan Rudolph
Organic Computing Group
University of Augsburg
stefan.rudolph@informatik.uni-augsburg.de

***Abstract*—In the field of Organic Computing, there is a trend to shift the adaption of systems to its application from design-time to run-time. This makes the presence of some kind of learning component indispensable. The basic assumption for this work is that in systems with multiple entities, the knowledge of dependencies in this system can give a huge advantage in terms of a faster and better learning of the best behavior in many situations. In this work, I propose a working plan for my PhD studies that outlines an approach to solve this issue. The planned work includes the development of a formal framework for the modelling of dependencies, techniques for the automated detection of these dependencies and a distributed controller which integrates this techniques.**

I. Introduction

In the field of complex systems, there is a huge expenditure of time for the developer in order to identify all possibly occurring events and find appropriate reactions to this events. Organic Computing [1] tries to solve this issue by shifting the adaption of systems to its application from design-time to run-time. To reach the goal of an adaption of the system to new situations without the intervention of humans, some kind of learning component has to be integrated in the system. The learning mechanisms available today only implicitly take into account the dependencies between entities in the system. This work presents an approach to make this dependencies an explicit factor in learning techniques in order to fasten the learning of better solutions.

The rest of the paper is structured as follows: In section I-A, the Observer/Controller (O/C) architecture in which the new techniques will be integrated is presented in short. In section I-B, the related work for the modelling and calculation of dependencies is described. In section II, a working plan, including the major issues and approaches to solve them, is outlined.

A. The Observer/Controller Architecture

This O/C architecture has been initially proposed in [2], therefore only a brief discussion follows. The overall goal of the architecture is to give a generic framework as a blueprint for future Cyber-Physical-Systems (CPS), although the architecture has a more general character and can also be applied to other settings besides CPS. This three-layer node architecture is depicted in Fig. 1. A major novel aspect of this architecture is to enable a strong collaboration between individual nodes in the system, in order to increase the speed at which the system as a whole can adapt itself to new situations. The proposed architecture separates different (environmental) scopes of action in order to implement the most suitable methods for each scope. At the first level, the *reaction layer*, local adjustments are made on a single node of the system ensuring fast and robust reactions to recurring, known situations. The second level, the *cognition layer*, extends the system's scope of action on a single node by longer-term observations and the application of machine learning techniques in order to extend the capabilities of the reaction layer towards unconsidered situations and anomalies in the environment. The third level, the *social layer*, enables the cooperation between multiple nodes of the system in order to react to situations that arise in a broader scope than what a single node can address. This includes joint actions of multiple nodes that require coordination in order to be successful.

Besides other details of the architecture, in the context of this work, the assumption that the reaction layer is rule-based is especially interesting. Each rule basically has the form:

$$[\mathit{CONDITION}][\mathit{ACTION}][\mathit{RATING}]$$

In this context, *CONDITION* is a situation model which allows for a determination of the degree of applicability each rule has for a given environmental situation. *ACTION* defines the output of the system that follows if a current input "matches" the given situation. Finally, *RATING* summarizes properties such as "usefulness" of a certain rule in the past, which are used to weight certain rules in certain situations.

Within this architecture, my work will focus on the development of techniques for the controller on the social and cognition layer.

B. Related Work

A starting point for the dependency modelling of parameters can be found in mathematics and Operation Research. In mathematical finance, the idea is to quantify dependencies between random variables, which are interesting for the modelling of credit default risks. A modern approach to this is the use of copulas. Copulas can model dependencies between random variables much more accurately than other measures and give them a huge advantage over simple techniques such as the covariance [3]. Another related approach are risk measures [4] that are heavily used in the field of Operation Research in order to quantify risk in the economic world. Commonly used

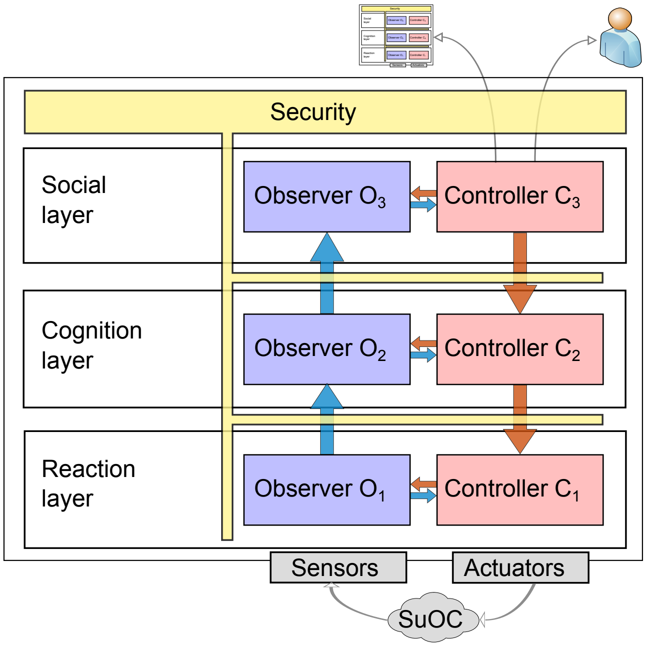

Fig. 1. Three-Layered O/C architecture

is, e.g., the Value-at-Risk approach. In this context, the risk is a value that is aggregated from the probability of a loss and the size of loss. Most of the time, it is necessary to estimate the dependencies between the random variables in order to calculate the risk. The afore described techniques are useful for the detection of dependencies. In order to determine the actual dependencies, including a possibility to accelerate the learning process of the system, there are several interesting approaches that might be adapted such as regression analysis [5], factor analysis [6], or principle component analysis [7].

II. Working Plan

In my PhD studies, I will investigate four major topics that are described in this section. First, I will develop a taxonomy regarding the *ACTION* spaces in OC systems (see section II-A). Second, I will investigate techniques for the modelling of dependencies and methods for the automated retrieval of such dependencies without the restrictions, such as, network bandwidth (see section II-B). The next step is to adapt this techniques to OC systems, such as, systems with local information only (see section II-C). The last step is to integrate the developed framework and techniques in a controller architecture (see section II-D).

A. Development of a taxonomy

Here, the goal is to develop a taxonomy for the *ACTION* spaces in OC systems. The assumption is that recurring patterns in the properties of these applications occur that allow researchers and developers to find generic O/C architectures that enable a proper functionality in the addressed application classes and thereby can be easily reused.

As a starting point for the classification, the large collection of Organic Computing applications that is examined in detail in [1] can be used. There are especially two taxonomies that use a very elaborated methodology: The first is a taxonomy for complex systems [8] and the second is for mobile applications [9] that have to be considered. Both of them use a method that combines the empirical-to-deductive and the deductive-to-empirical approach in order to find an adequate amount of characteristics and classes that are important. There are other classifications, for instance, regarding uncertainty [10], [11] that might be useful as inspiration for such characteristics.

B. Development of modelling approaches for dependencies in the ACTION space of intra-node learning systems

The goal of this work package is to find suitable approaches for modelling dependencies in the *ACTION* space of CPS. In this WP, the focus is on *local* dependencies, i.e., dependencies between atomic *ACTION*s, where no communication overhead is needed to detect these dependencies. Inter-node dependencies are an issue in section II-C.

These objectives have to be seen in the context of the afore introduced 3-layered architecture of the node controller. Within this architecture, the function of the controller in the cognition layer is to create new rules that are applied and evaluated in the reaction layer. These rules will exploit the dependency structures in the application. The task of the controller in the social layer is to gather information about the inter-node dependencies of the system and to forward relevant information to C_2 in order to further improve the rule creation process.

The approach starts with an abstract mathematical description of the requirements for the dependency measures within the *ACTION* space that will serve as a basis for the more concrete modelling of these dependencies. In all applications, the *ACTION* space can be described as a space $K = V_1 \times V_2 \times \ldots \times V_n$ where the V_i are the *ACTION* spaces of a single parameter. This means, the whole *ACTION* space is just a combination of the n elementary subspaces of the *ACTION* space. The objective is here to find a function $\alpha : \mathcal{P}(\{1,\ldots,n\}) \times \mathcal{P}(\{1,\ldots,n\}) \longrightarrow [0,1]$, which quantifies the degree of the dependency between two arbitrary subspaces of the *ACTION* space in a range from 0 (no dependency at all) to 1 (strongest possible dependency). It is possible to formulate abstract requirements for this function α, for example

Self-Independence: The function α fulfills the property of self-independence if $\alpha(P,P) = 0 \quad \forall\ P \subseteq \{1,\ldots,n\}$. This property is natural because every sub$ACTION$ space should be completely independent to itself.

Symmetry: The function α is symmetric, if $\alpha(P_1,P_2) = \alpha(P_2,P_1) \ \ \forall\ P_1,P_2 \subseteq \{1,\ldots,n\}$. This property implies that all dependencies are in effect in both directions.

These are example properties that have to be extended, but are valuable for the development of dependency measures for specific systems.

A starting point for the more concrete modelling within this framework can be found in mathematics and Operation Research. There are also promising approaches in mathematical finance. The concept of dependency measures (in stochastics, cf. section I-B) is especially interesting in systems with only local information. A possibility is to interpret the local fitness and the atomic *ACTION*s as random variables and quantify the dependencies between them. The usefulness of this technique in the machine learning domain will be analyzed. Another approach are risk measures (see section I-B). In this context, the risk is a value that is aggregated from the probability of a loss and the size of loss. This notion of risk can be interpreted as a measure of dependencies in the way that atomic *ACTION*s have a high dependency if and only if there is a high risk. The application of such risk measures to the field of machine learning will be analyzed.

In order to exploit the knowledge of the dependencies, it is not only necessary to quantify them by using the presented techniques but to describe the actual manifestation of the dependencies. Suitable representations for the dependencies could be functions, inequations and propositional or temporal logic. For the creation of such representations, different tools may be used. For example the mathematical techniques regression, principle component analysis or factor analysis can be adapted to the problem or Genetic Algorithms might be used.

C. Development of modelling approaches for dependencies in the ACTION space of distributed learning systems

The goal of this work package is to extend the techniques developed in section II-B for the detection and exploitation of intra-node dependencies for inter-node dependencies. In the preceding work the focus was on intra-node dependencies in the *ACTION* space. This ignores some issues that occur in a distributed OC system, e.g., the negative effects of a huge communication overhead and problems related to a heterogeneous network of nodes, such as different learning strategies and conflicting goals.
Here, the following research issues will be considered. (1) Looking at the inter-node dependencies in the *ACTION* space, it is crucial to find techniques to smartly limit data exchange for the detection of such dependencies. The objective is to find and analyze approaches to keep the communication overhead as low as possible. On the one hand, this can be reached through data selection, i.e., the relevant data will be spread but not unimportant (or less important) data, or data aggregation, for example, by merging multiple values over time to a single one. With these data it will not be possible to analyze the dependencies as well as with the raw data, but it is sufficient to identify the important nodes. Knowing these nodes, the communication with them can be increased and the detailed dependency structure can be discovered. (2) Considering heterogeneous systems, it might be applicable to use different learning techniques and strategies in different nodes. In this case it is necessary to integrate the knowledge about dependencies in all the nodes, i.e., in all the different learning strategies used at the cognition layer. For this purpose I will develop a generic representation of the dependencies based on the ones developed in section II-B and integrate them into different learning techniques.

Hence, here, the focus is on the cooperation between nodes by retrieving information about dependencies and exploiting this knowledge in a distributed system. This will provide the applicability in (heterogeneous) distributed OC systems.

D. Development of distributed controller architectures

The goal of this work package is to develop suitable architectures for the interaction between the nodes. Of course, it is very likely that the optimal architecture differs from one application to another, but general conclusions regarding the general classes of applications should be possible.

There are three basic architecture approaches to consider: P2P systems [12]–[14], hierarchical systems [15] and hybrid systems [16].

The approaches will be analyzed for the use in a distributed controller regarding the applicability to different classes of CPS. An assumption is that it is possible to find patterns in the utility of architectures in different classes of applications.

Besides the basic structure of the architecture of the system, there is another research issue that is the communication paradigm. There are several options for this purpose in general. First, it is possible to use a message-based communication system in which the node communicate by sending messages to each other. Second, a state-based approach is investigated in which the nodes are able to ask for the state of a second node in order to decide appropriately. Furthermore, publish/subscribe approaches will be investigated.

III. Conclusion

The main contributions of this work will be the development of the formal framework for the modelling of dependencies and the investigation of techniques for the automated detection of this dependencies, as well as, the development of a distributed controller and the integration in the presented generic O/C architecture. In order to test the applicability of the developed techniques, the principles will be applied to at least two scenarios, a self-organizing smart camera network and an urban traffic control system.

References

[1] C. Müller-Schloer, H. Schmeck, and T. Ungerer, Eds., *Organic Computing – A Paradigm Shift for Complex Systems*. Birkhäuser Verlag, 2011.

[2] J. Hähner, S. Rudolph, S. Tomforde, D. Fisch, B. Sick, N. Kopal, and A. Wacker, "A Concept for Securing Cyber-Physical Systems with Organic Computing Techniques," in *Proc. of ARCS'13 Works.*, 2013, pp. 1–12.

[3] R. Nelsen, *An Introduction to Copulas*. Springer, 1998.

[4] A. J. McNeil, R. Frey, and P. Embrechts, *Quantitative risk management: concepts, techniques and tools*. Princeton University Press, 2005.

[5] N. R. Draper and H. Smith, *Applied regression analysis*. New York: Wiley, 1966.

[6] R. L. Gorsuch, *Factor Analysis*, 2nd ed. Taylor & Francis Group, 1983.

[7] I. T. Jolliffe, *Principal Component Analysis*. Springer, 2002.

[8] C. L. Magee and O. L. de Weck, "Complex System Classification," in *Proc. of INCOSE'04*, 2004.

[9] R. C. Nickerson, U. Varshney, J. Muntermann, and H. Isaac, "Taxonomy development in information systems: Developing a taxonomy of mobile applications." in *Proc. of ECIS'09*, 2009, pp. 1138–1149.

[10] D. Thunnissen, “Uncertainty classification for the design and development of complex systems,” in *3rd Predictive Methods Conf.*, 2003.
[11] O. L. de Weck and C. Eckert, “A Classification of Uncertainty for Early Product and System Design,” in *Proc. of 16th Int’l Conf. on Engineering Design*, 2007.
[12] E. Cakar, J. Hähner, and C. Müller-Schloer, “Investigation of Generic Observer/Controller Architectures in a Traffic Scenario,” in *Beiträge der 38. Jahrestagung der GI*, 2008, pp. 733–738.
[13] S. Tomforde, B. Hurling, and J. Hähner, “Distributed Network Protocol Parameter Adaptation in Mobile Ad-Hoc Networks,” in *Selected Papers from ICINCO’11*, ser. LNEE. Springer, 2011, vol. 89, pp. 91–104.
[14] H. Prothmann, J. Branke, H. Schmeck, S. Tomforde, F. Rochner, J. Hähner, and C. Müller-Schloer, “Organic Traffic Light Control for Urban Road Networks,” *Int. J. of Autonomous and Adaptive Communications Systems*, vol. 2, no. 3, pp. 203–225, 2009.
[15] U. Richter, H. Prothmann, and H. Schmeck, “Improving XCS Performance by Distribution,” in *Proc. of SEAL’08*, ser. LNCS, vol. 5361. Springer, 2008, Inproceedings, pp. 111–120.
[16] H. H. Dam, H. A. Abbass, and C. Lokan, “DXCS: an XCS system for distributed data mining,” in *Proc. of GECCO ’05*. ACM, 2005, pp. 1883–1890.