

Organic computing for health care systems

Florina Nafz, Frank Ortmeier, Hella Seebach, Wolfgang Reif

Angaben zur Veröffentlichung / Publication details:

Nafz, Florina, Frank Ortmeier, Hella Seebach, and Wolfgang Reif. 2008. "Organic computing for health care systems." In *Proceedings of the First International Conference on Health Informatics - January 28-31, 2008, in Funchal, Madeira, Portugal*, edited by Luis Azevedo and Ana Londral, 286–90. Setúbal: SciTePress.
<https://doi.org/10.5220/0001044402860290>.

ORGANIC COMPUTING FOR HEALTH CARE SYSTEMS

Possible Benefits and Challenges

Florian Nafz, Frank Ortmeier, Hella Seebach and Wolfgang Reif
Lehrstuhl für Softwaretechnik und Programmiersprachen, Universität Augsburg
Universitätsstr. 14, 86135 Universität Augsburg, Germany

Keywords: Organic Computing, multi-agent systems, health-care applications, security.

Abstract: Today's health care institutions will undergo major changes in the next two decades. The reason for this is the change of ageing structure in many industrialized countries. In Germany statistics indicate that the costs for health care systems will at least double per person while the number of contributing, working citizens will significantly lower. At the same time average life expectation will rise above 80 years. To cope up with this development adaptations to organization and process of health care are necessary. Typically tasks in stationary health care can be divided in two groups: task which incorporate direct interaction with the patient (care tasks) and tasks which focus on logistics and organization (background tasks). In health care it is not desirable and feasible to reduce efforts in care tasks. So costs and efforts must be reduced within the second group of tasks. This is possible if new paradigms – both in organization and underlying software architecture – are applied. One such paradigm is organic computing. Organic computing aims at systems, which are self-organizing, self-adapting to new challenges and self-optimize during runtime. Such systems can take away a lot of organizational work from the staff and thus allow for more and better care without rising budgets. The paper outlines the idea of organic computing as well as opportunities and challenges for applying it in the health care context.

1 INTRODUCTION

It is a widely accepted fact, that cost pressure will make changes in today's health care institutions necessary. Reasons for this development are the changing ageing structure and the rising life expectation of many industrial countries' population. In 2005 there were in average 31.7 persons older than 65 years for every 100 persons in working age. This percentage will rise to 54.2 persons in 2030 and to 70.9 persons in the year 2050 (Statistisches Bundesamt, 2006). At the same time the average per year cost for health care for persons older than 65 years is four times the average cost for younger people (Statistisches Bundesamt, 2007). These two facts together show that costs will dramatically rise while the income will stay almost constant.

It is not feasible to simply cut health care benefits to reduce costs. Therefore it is necessary to provide health care more cost efficient. Humans play – and will always play – the most important role in caring for others. So it is not possible and wanted to “remove” human staff from the caring process. To keep the health care systems affordable and survivable it will be of uttermost importance to support the staff in

all tasks, which are not directly related to the patients as much as possible.

This can only be done by providing intelligent infrastructure and background systems, which support the medical staff autonomously whenever possible. A paradigm for construction of such systems is Organic Computing (VDE/ITG/GI, 2003). It allows for constructing flexible systems, which dynamically (re-)configure and autonomously self-adapt to changing tasks and requirements.

In this paper a brief overview on Organic Computing is given in Sect. 2. Sect. 3 shows chances and possibilities of applying Organic Computing in health care on a little example. Open questions for research and challenges are discussed in Sect. 4. Sect. 5 concludes this position paper.

2 ORGANIC COMPUTING TECHNIQUES

The complexity of modern systems has grown immensely in the last years. In particular the trend to replace hardware solutions with software components has made the software become (a) more and more im-

portant for the functionality and (b) more and more complex to develop. A good example is a modern (digi-)cam. Five years ago most cameras were using analogous media to record pictures. Extra functionalities like (visual) noise suppression or image stabilization were implemented by using specific media or by adding extra hardware components. Today such functions are usually implemented in software and can relatively easily be transferred from one camera to another. This led to very complex software and this process is now reaching a saturation level. To further enhance the capabilities of both – the system in general and the controlling software in particular – new approaches are necessary to deal with the growing complexity. One such approach is Organic Computing (Müller-Schloer, 2004).

The Organic Computing Paradigm. Organic Computing (Müller-Schloer et al., 2004; VDE/ITG/GI, 2003) is an extension to Ubiquitous and Autonomic Computing (Kephart and Chess, 2003). The core goal of Organic Computing is, that future (computing) systems should be able to dynamically adapt to changes in requirements, to automatically detect and neutralize component failures and to continuously optimize themselves for better performance. Such capabilities are called self-adaptation, self-healing and self-optimization. The hope is, that systems with self-x capabilities are by far superior to standard systems in terms of possible functionalities, availability, reliability and effort for maintenance. At the same time self-organization mechanisms allow for easier construction. As not every single scenario has to be anticipated at design time. New scenarios and requirements will be detected by the system during runtime and self-adaptation mechanisms will reconfigure the system to this new challenges. So the systems behave from an external point of view very much like living beings. They monitor their environment, reflect upon changes and adapt to new situations. Therefore such systems are called “Organic Computing” systems.

It is often useful and possible to cope mechanisms and techniques from biology/sociology for the design of self-X algorithms and systems – so called bio-inspired algorithms. One example is a (self-)protection algorithm which function like the human immune system (Pietzowski et al., 2006). But there also exists a variety of other algorithms, which are based on more traditional approaches.

Technically, implementations of Organic Computing systems often contain one part which is responsible for delivering the wanted/intended functionality and one part which constantly monitors the environ-

ment and – if necessary – controls/changes the functional part. There already exists a broad variety of methods (Seebach et al., 2007; Richter et al., 2006), middlewares (Trumler et al., 2004; Trumler, 2006) and analysis tools (Güdemann et al., 2007; Güdemann et al., 2006) for design, analysis and construction of such systems.

Application Domains. Organic Computing is typically applied to software-intensive embedded systems. Example domains are large networks of sensors, traffic control systems or production automation. One example could be a vision of the next generation of production cells. Assume a production cell which should process workpieces following a given specification. The functional part of this system may be a set of robots which perform certain tasks – e.g. drilling holes, inserting screws and tightening screws – on workpieces and a set of transport units, which transport workpieces from station to station. If new types of workpiece are to be processed, then the robots must be reprogrammed (or these new workpieces must have already been anticipated during system design). If a single robot/component of the system fails, then production will often come to a stand still.

If in contrast this system is designed in an organic way, then each robot will be enhanced with a supervising (software) component. This software not only monitors the supervised robot, but also decides/plans what tasks the robot will perform and from where workpieces are taken resp. where workpieces are to be placed next. If new workpieces – with a new specification about how they are to be processed – are given to the production cell, then the supervisors try together to find a reconfiguration to achieve the new goal.

Assume for example a cell that consists of three robots, which are all capable of drilling holes, inserting screws and tightening them and a set of autonomous carts for transportation, then an (initial) configuration could state that one robot only drills, one only inserts screw and one only tightens the screws. The workpieces will be transported in batches from robot 1 to robot 2 to robot 3. The supervisors constantly monitor the functionality of the robots and the type of workpieces which are to be processed. If now for example the drill of robot 1 breaks, then the supervisors will search for a new configuration to subsume production. One solution is to let robot 2 do the drilling, assign robot 1 to inserting screws and change the transportation routes accordingly. This is called self-healing. If a new type of workpiece is to be processed; e.g. workpieces now only need holes (and no

screws at all). Then the supervisor will recognize this and reconfigure all robots to drill (self-adaptation)¹.

The technical basis for such effects is (i) the capability of the supervisors to reflect their controlled subsystems capabilities, (ii) to interact/communicate with each other and to (iii) use a common language about goals and capabilities. In the last years there have been big advances in building Organic Computing systems. There exists now a number of working middlewares, organic architectures and intelligent re-configuration algorithms. They have also been successfully applied to various technical scenarios. But there exist hardly any application to systems in which privacy and individual user-trust play an important role.

3 ORGANIC COMPUTING IN HEALTH CARE

How can Organic Computing help for health care? What are the challenges, which have to be met to allow for using Organic Computing in health care? These questions can be best answered, if an example scenario is taken into account.

A possible example scenario may be support and comfort functionality in future rehabilitation centers. This scenario is only one selected example of a broad class of background support systems both inside and outside the domain of health care.

An Example. The scenario assumes, that during the next decade patients as well as medical staff and doctors will be equipped with mobile devices, which can form ad-hoc-networks. The topology of these networks will be continuously changing as the devices are carried around by their possessors. Stationary devices like ergo-meters or central services will also be equipped with wireless communication and can also participate in the network.

Typically patients in a rehabilitation clinic arrive with a fixed treatment concept. This concept states a number of treatments (e.g. physio, massage, etc.) which have to be applied to the patient in certain intervals (e.g. twice a week). Periodically – often every two weeks – these concepts may be changed or updated according to the physical state of the patient. Using the treatment concept and the capacities of the staff and installations as input data, a central administration service creates a treatment plan for the patients. This plan schedules treatments to certain

¹A more detailed description of this example may be found in (Güdemann et al., 2006).

time slots and staff/installations. Similarly work plans for the staff are created. This process is relatively complex and requires intensive planning. It is usually done on a weekly basis. The plans are typically printed and handed out to patients and staff.

This process is performing adequately while there are no unpredicted events. In reality there typically occur a lot of disturbances – like patients coming to late to their treatments, illness of staff or malfunction of some equipment. The consequence is that the plans must be adapted. This is often not possible or only if the disturbance can be foreseen (like illness of staff). As a consequence in most cases treatments will either be canceled or at least postponed for several days. The root of this problem lies in the static nature of this process and the central planning.

Benefits of Organic Computing in Health Care.

Better results can be achieved if an Organic Computing system is implemented for this task. In an organic approach, the treatment plan will not be calculated a priori by a central administration. It will rather evolve by interaction of all agents with each others. This allows for dynamic reconfiguration and continuous adaptation for optimal solutions. The agents can much better locally decide if a schedule for treatments is possible, impossible or borderline. Agents are in this context the mobile devices of all relevant participants. An automatic replanning algorithm could for example first collect localization information of the patient and the next treatment's position. It will then consider the physical state of patients or delays of the treatment staff (for example because of meetings or emergencies). Together this information will often allow for dynamic re-organization of treatment schedules. The consequence is that many disturbances can be compensated by a little reorganization. The big benefit is, that this can be done "on-the-fly" without the need of a central coordination.

An example: assume patient A will not make it to his next treatment in time, while patient B, who is scheduled right after A, is already close to the treatment. It is then possible to inform A, B and the staff and to dynamically switch the patients' treatment. Note, that this simple scenario could also be possible without a computer aided planning system. But it is clear that the same method can also be used to make more complex re-schedulings, which may involve multiple changes. This is better than traditional approaches, because firstly re-planning starts as soon as a disturbance is detected and secondly it is not necessary for all agents to physically meet at one place but rather share and exchange information through the ad-hoc network. Similar algorithms can compen-

sate for delays of staff, broken equipment and many other disturbances. Furthermore the system could be used to *automatically* integrate new patients into the system by calculating treatment schedules for them. As a result of later automatic optimization some re-scheduling of existing schedules could make sense to improve the overall quality of the planning system.

All the properties described above are typical properties of Organic Computing systems. In Organic Computing these effects are called self-organizing (e.g. autonomously integrating new patients and calculating schedules), self-optimizing (e.g. minimizing waiting times), self-healing (e.g. compensating for broken equipment or illness of staff) and self-adapting (e.g. adapting to changing treatment requirements and health situation of a patient). So its only a logical step to apply techniques, tools and middlewares of Organic Computing to the domain of health care.

4 DOMAIN SPECIFIC CHALLENGES

What are the specific challenges to implementing an Organic Computing system in health care? The sad point is: traditional Organic Computing techniques are not directly applicable to this domain. The problem is, that most Organic Computing approaches make two assumptions of the systems and its components agents. The first assumption is, that all participants/agents share a common goal. The second one is, that there are no “malicious” parties involved. This means, all agents work for common interests and no agent tries to gain only benefits for himself while causing major drawbacks for all others. This arises a number of challenge, which must be mastered for successfully integrating Organic Computing in the domain of health care:

1) Individual Goals

Whenever agents of an Organic Computing system are personalized to individual persons, they in general won't share a common goal. In the example described above, agents owned by patients, medical staff and administration will have different goals. Patients are interested in treatment schedules, which are compatible with their leisure times, medical employees are interested in compact working times and the clinic's administration is interested in maximum profits i.e. no canceled treatments. These conflicting goals have to be balanced and weighted in a meaningful manner. It must also be assured, that security mechanisms prohibit individual persons from getting benefits on the cost of drawbacks for multiple others. So it will be

a challenge for the next generation of Organic Computing algorithms to be able to cope with individual goals and requirements.

2) Sensitive Data

Personal data is sensitive. Medical data is even more sensitive. Therefore privacy and security is a prime requirement for every system, which handles data in a medical care scenario. On the other hand, the Organic Computing paradigm strongly relies on interaction and exchange of data between system components/agents. New and optimal configurations are computed at runtime jointly by all agents. This is only possible, because of information exchange and reflection of the exchanged data. For health care it will be a challenge to adapt Organic Computing algorithms such that only for reconfiguration and planning necessary data is exchanged and that sensitive data can be kept private wherever and whenever possible.

3) Behavioral Guarantees

Health care is a highly critical domain. Whenever a system is allowed to autonomously make decisions, which have effects on the medical treatment of a patient, it must be assured that these decisions may never pose a threat to the health of the patient. This problem can be solved by applying formal methods for analyzing the systems. They allow for rigorously proving, that the system will always fulfill some behavioral guarantees. For Organic Computing such analysis is much more difficult, because it can often not be anticipated at design time in which environments/scenarios the system will eventually be asked to (self-)adapt to. Therefore new analysis methods must be developed, which allow for giving behavioral guarantees for Organic Computing systems.

4) User Trust

This is possibly the most important challenge. Organic Computing systems can achieve a lot of benefits from a global point of view. The benefits result from their ability to self-organize and self-adapt. Improvements become typically visible on system wide performance metrics. For local agents, (re-) configuration is not necessarily beneficial or even traceable. This will also affect the user, who own the agents. If users are often confronted with decisions/plannings of the system, which they can not understand (and from which they don't benefit), then they will likely not accept the system. This must not happen. So it will be a challenge to provide algorithms, which provide users with enough information to understand and accept the decision of the system (without violating privacy constraints).

5 CONCLUSIONS

Organic Computing is a very promising new method for construction of modern software systems. It has proven to be an superior architecture in many domains like traffic control, sensor networks or production automation. Organic Computing is also very promising for a variety of user-intensive scenarios, where individual users own individual agents. The purpose of these agents can be to assist, support and/or guide their users. One scenario in health care is a planning/scheduling system in a rehabilitation clinic. An organic system can perform in this context much better than traditional approaches. On the other hand constraints, which arise in this domain – like concurrent/individual goals of users, privacy, behavioral guarantees and user-trust – require new Organic Computing algorithms and techniques. Developing such algorithms and methods can have a significant impact on many domains and open new opportunities and functionalities.

REFERENCES

- Güdemann, M., Ortmeier, F., and Reif, W. (2007). Using deductive cause consequence analysis (DCCA) with scade. In *Proceedings of SAFECOMP 2007*. Springer LNCS. accepted at SAFECOMP07.
- Güdemann, M., Ortmeier, F., and Reif, W. (2006). Formal modeling and verification of systems with self-x properties. In Yang, L. T., Jin, H., Ma, J., and Ungerer, T., editors, *Proceedings of the Third International Conference on Autonomic and Trusted Computing (ATC-06)*, volume 4158 of *Lecture Notes in Computer Science*, pages 38–47, Berlin/Heidelberg. Springer.
- Güdemann, M., Ortmeier, F., and Reif, W. (2006). Safety and dependability analysis of self-adaptive systems. In *Proceedings of ISoLA 2006, 2nd Symposium on Leveraging Applications of Formal Methods, Verification and Validation*. IEEE CS Press.
- Kephart, J. O. and Chess, D. M. (2003). The vision of autonomic computing. *Computer*, 36(1):41–50.
- Müller-Schloer, C. (2004). Organic computing: on the feasibility of controlled emergence. In *CODES+ISSS '04*, NY, USA. ACM Press.
- Müller-Schloer, C., von der Malsburg, C., and Würtz, R. P. (2004). Organic computing. *Informatik Spektrum*, 27(4):332–336.
- Pietzowski, A., Satzger, B., Trumler, W., and Ungerer, T., editors (2006). *A Bio-Inspired Approach for Self-Protecting an Organic Middleware with Artificial Antibodies*, volume LNCS 4124. Passau, Germany.
- Richter, U., Mnif, M., Branke, J., Müller-Schloer, C., and Schmeck, H. (2006). Towards a generic observer/controller architecture for organic computing. In Hochberger, C. and Liskowsky, R., editors, *INFORMATIK 2006 – Informatik für Menschen*, GI-Edition – Lecture Notes in Informatics, pages 112–119, Bonn, Germany. Köllen Verlag.
- Seebach, H., Ortmeier, F., and Reif, W. (2007). Design and Construction of Organic Computing Systems. In *Proceedings of the IEEE Congress on Evolutionary Computation 2007*. IEEE Computer Society Press. accepted for publication.
- Statistisches Bundesamt (2006). 11. koordinierte bevölkerungsvorausberechnung – annahmen und ergebnisse.
- Statistisches Bundesamt (2007). Gesundheit – ausgaben 2005.
- Trumler, W. (2006). *Organic Ubiquitous Middleware*. PhD thesis, Universität Augsburg, Eichleitnerstr. 30.
- Trumler, W., Bagci, F., Petzold, J., and Ungerer, T. (September 20–24 2004). Towards an Organic Middleware for the Smart Doorplate Project. In *Workshop on Organic Computing, INFORMATIK 2004 - Informatik verbindet*, pages 626–630, Ulm, Germany.
- VDE/ITG/GI (2003). Organic Computing: Computer- und Systemarchitektur im Jahr 2010. [http://www.gi-ev.de/download/VDE-ITG-GI-Positionspapier Organic Computing.pdf](http://www.gi-ev.de/download/VDE-ITG-GI-Positionspapier%20Organic%20Computing.pdf).