A Monitoring Infrastructure for the Digital on-demand Computing Organism (DodOrg)

Rainer Buchty, Wolfgang Karl*

* *Universität Karlsruhe (TH), Institut für Technische Informatik*
76128 Karlsruhe, Germany
E-Mail: `{buchty|karl}@ira.uka.de`

ABSTRACT

The Digital on-demand Computing Organism (DodOrg) is a novel system concept based on biological concepts. Major part of DodOrg is a sophisticated monitoring infrastructure spanning all system layers from hardware to application, providing necessary information for system surveillance and the adaptive processes triggered by an organic middleware and the low-power planning.

KEYWORDS: Adaptive Computing, Reconfiguration, Self-X, Self-Optimization

1 Introduction and Motivation

The Digital on-demand Organism (DodOrg) [BBB+06] is an ambitious project funded by the *Deutsche Forschungsgesellschaft (DFG)* within the priority program 1183 (SPP1183) "Organic Computing". The idea behind DodOrg is to adopt biological concepts for use within digital architectures.

Following biology terminology, the DodOrg architecture divides into brain-, organ- and cell-level. On brain level an organic robot control will be developed with emphasis on Self-X features. Together with the middleware and low-power control (organ level), and reconfigurable hardware (cell level) an organic computing architecture is formed. This is illustrated by Figure 1. Work on this project is pursued by five research groups, each being responsible for a distinct topic within the DodOrg project as shown in Figure 2.

Any organic architecture requires a comprehensive, flexible, and adaptive monitoring approach, therefore system monitoring is the important issue with self-organizing systems. In particular, the entire system must be constantly evaluated, hence monitoring within DodOrg spans all system levels, i.e. individual monitors are distributed across the system and are connected to different components and layers.

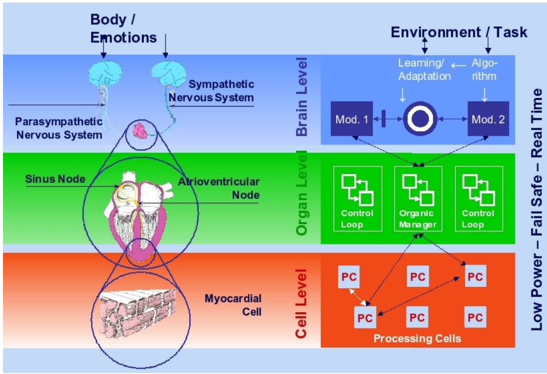

Figure 1: The DodOrg Concept

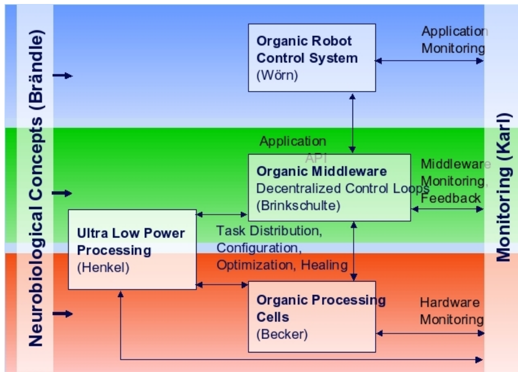

Figure 2: DodOrg Working Groups

In this paper we will outline the current status of the DodOrg monitoring infrastructure: Section 2 contains a detailed presentation of the monitoring concept to be used within the DodOrg implementation. The paper is concluded with Section 3.

2 Monitoring Architecture

The DodOrg hardware is a heterogeneous, adaptive multicore architecture comprising of several processing elements, so-called *cells*, connected through a peer-to-peer network as illustrated by Figure 3. For this reason, each cell does not only contain its core functionality but also dedicated router functionality supporting communication in this network.

As outlined in the introduction, monitoring is distributed over the entire system architecture. No centralized monitoring instance exists and each monitoring instance is per se independent. This is achieved by embedding monitoring functionality into each part of the system. On hardware-level, monitoring is part of every cell including basic preprocessing on this level to achieve semantic compression of the collected monitoring data minimizing the communication overhead. Higher-level, aggregate monitoring functions can be realized using dedicated monitoring cells or, if necessary, monitoring organs created from arbitrary processing elements.

Because a peer-to-peer network is used as a shared transport medium for all communication taking place in the system, two basic monitoring functions, need to be derived from each cell's router which are network load and communication errors. Network load statistics are used locally to optimize packet forwarding within the peer-to-peer network and is provided to the middleware and low-power planning to enable organ formation and reconfiguration. In addition to determine network load, also communication errors must be detected. Depending on their severity or persistance, this data is not only used locally, but actively forwarded to the planning instances.

In addition to those general statistics, additional information needs to be collected depending on the processing cell's type such as performance counters for CPU and DSP cells, or usage statistics for memory cells known from cache and data locality optimizations. Similarly, FPGAs need to report their configuration state and usage statistics. If possible to be implemented directly into hardware, also power usage and heat dissipation need to be measured.

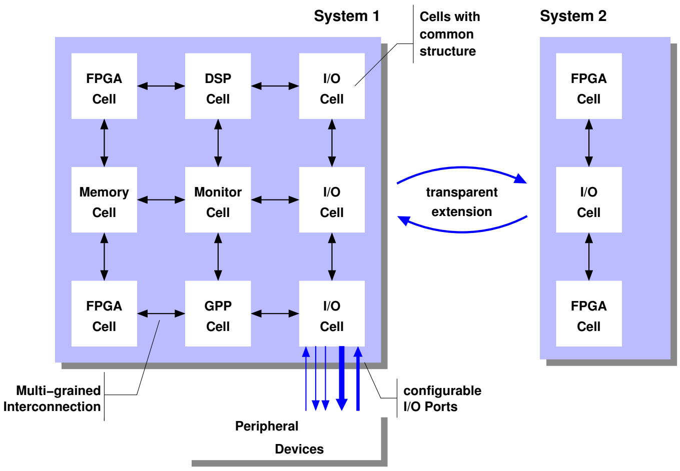

Figure 3: DodOrg Heterogeneous Multicore Architecture

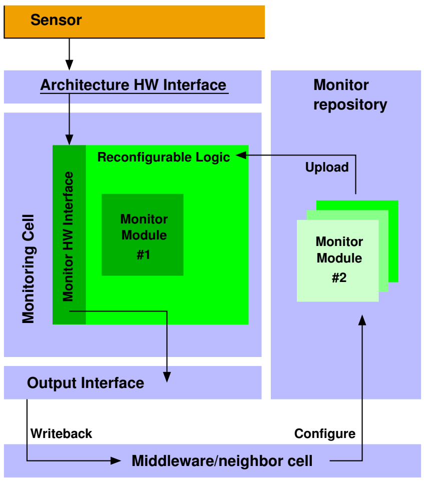

Figure 4: Monitoring Infrastructure

2.1 From Basic to Complex Monitoring

Based on this low-level monitoring more complex monitoring functions are derived by using either dedicated monitoring cells, or mini-organs e.g. consisting of CPU, FPGA logic, and memory. In such configuration, monitoring from individual cells is routed into the dedicated monitoring fabric consisting of one or more monitoring cells, where the data gets pre-processed and interpreted. This reduces monitoring-related communication overhead, but also contributes to rather light-weight middleware and low-power planning, because data interpretation can be "outsourced" to the monitoring infrastructure. In addition, it introduces some level of fault-tolerance if – necessary system resources provided – this interpretation is done redundantly.

2.2 Monitoring Communication

Monitoring must closely interface with the planning infrastructure, i.e. organic middleware and low-power planning. Hence, we defined message and communication types as well as communication modes necessary for such a setup. We will first deal with the **message types** defining the type of payload transported: **Configuration** is required to set-up and alter the monitoring infrastructure. Through such configuration cycles, desired monitoring functionality is selected. Such configuration may either take place from internal, e.g. by a monitoring organ parametrizing its cells, or by higher system levels such as the middleware. To ease reconfiguration, monitoring functionality is strictly divided into the physical or logical interface, the *monitor capsule* (MC), and the requested function, called *monitor module* (MM). Reconfiguration beyond simple parametrization of existent monitoring functionality will therefore be typically realized by loading MMs into the corresponding MCs. This is illustrated by Figure 4. Once monitoring is set up, **information** will be transmitted either autonomously, or by request. For the latter, another communication type needs to be introduced, the **information request**.

2.3 Transmission of Monitoring Events

Certain types of events need to be transmitted instantaneously and autonomously by the monitoring infrastructure. In addition, certain messages need to be sent periodically such as heartbeats or status updates. Finally, information is gathered which needs only to be transmitted on demand. For this reason, three communication types were defined: **autonomous & aperiodical** messages are typically required to signal events which require immediate reaction. Such events will typically lead to system reconfiguration to e.g counter detected faults or accomplish for a changed mode of power supply.

In certain cases, the proper functioning of a system unit is signalled by periodical heartbeats or detected by periodic sampling of certain system parameters. To accomplish for such events, monitoring needs to support **autonomous & periodical** messages. This takes pressure from higher system levels which instruct monitoring to periodically collect and, if required, preprocess the collected data. Such preprocessed data might not require immediate transmission but should be polled on demand. We therefore define the third communication type as **non-autonomous**.

Finally, the **communication mode** defines how many recipients a monitoring message has. These may be directed to a single recipient, e.g. from individual monitoring cells to the coordinating cell within a monitoring organ. This we call a **unicast** message. In case a message needs to be distributed among a defined group of cells, we call this a **multicast** message. Finally, a **broadcast** message will reach all cells within a defined radius, i.e. within an organ, a chip, or the entire system.

2.4 Monitoring Prototype

As of now, a prototypic implementation of the DodOrg monitoring infrastructure is actively developed. With the help of this prototype the suitability of the overall concept will be evaluated and eventually fine-tuned. This prototype is developed in close cooperation with the other project groups to ease integration into a common demonstrating prototype at a later project stage.

The monitor prototype consists of a configurable monitoring service resembling the proposed monitoring organs or monitoring cells. This monitor service can be instructed to invoke certain monitoring primitives (monitor modules) providing configurable low-level monitoring functions which can be combined to accumulate monitoring functions on higher level as described in the introductory part of Section 2.

3 Conclusion

In this paper we presented a first sketch of the monitoring infrastructure to be used within the Digital on-demand Computing Organism (DodOrg). Due to DodOrg's self-reconfiguring nature, monitoring plays a vital role in information retrieval and processing. Because of the communication method chosen for DodOrg, it must be ensured that no communication bandwidth is wasted by unnecessary message exchange. Thus, monitoring has to accomplish for certain communication types and modes, as well as support distinct message types.

References

[BBB+06] Jürgen Becker, Kurt Brändle, Uwe Brinkschulte, Jörg Henkel, Wolfgang Karl, Thorsten Köster, Michael Wenz, and Heinz Wörn. Digital On-Demand Computing Organism for Real-Time Systems. In Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, and Erik Maehle, editors, *ARCS'06 Workshop Proceedings*, pages 230–245. Gesellschaft für Informatik e.V., March 2006. Lecture Notes in Informatics (LNI) P-81, ISBN 3-88579-175-7.