# Towards Autonomic Workflow Management Systems

Markus Strohmaier[1], Eric Yu[2]

[1]Department of Computer Science, University of Toronto

*mstrohm@cs.toronto.edu*

[2]Faculty of Information Studies, University of Toronto

*yu@ fis.utoronto.ca*

## 1 Motivation

In a world of dynamic and discontinuous change, systems constantly need to adapt to new conditions so that they can survive and flourish in their environment. *Autonomic computing* emerged as a research field that takes up this challenge and aims to build systems that are capable *of adapting automatically to dynamically changing environments* (Self-configuring), *discovering, diagnosing and reacting to disruptions* (Self-healing), *monitoring and tuning resources automatically* (Self-optimizing) and *anticipating, detecting, identifying and protecting themselves from attacks* (Self-protecting) [3]. A major application area for autonomic computing is intended to be *system administration*, aiming to free system administrators from the details of system operation and maintenance [8], improving robustness of systems and decreasing total cost of ownership. However, the vision of autonomic computing does not need to be restricted to the area of system administration. For example, much research has been done in the area of *process-aware information systems* [2] such as Workflow Management, Enterprise Resource Planning, Business-to-Business and Customer Relationship systems to effectively and efficiently deal with change on different levels and scales. Frequent questions in these domains include: How can changes to workflows be accommodated? How can flexibility and adaptability of running workflow instances be increased? How can workflow management systems themselves optimize workflow definitions? The type of questions raised here seems to address issues that are similarly addressed by research in autonomic computing, where dealing with change represents a major concern. However, little research has been done on the intersection between these two domains [4]. Based on this observation, this contribution aims to tackle the question: "*Can the principles of autonomic computing be applied to workflow management – and if so, how?*"

According to [11], change in the context of Workflow Management Systems (WfMSs) can take place on two levels: the *workflow type* level and the *workflow instance* level. Change on a workflow type level potentially affects running instances on the workflow instance level. Strategies to deal with such issues include a) creating a *new version* of the modified workflow type while leaving running instances untouched or b) migrating a potentially large number of running workflow instances from an old workflow type to a new one (which is referred to as *change propagation*).

On a workflow instance level, change has different implications. Here, a*nticipated changes* can be facilitated by adaptive workflow management systems through for example the provision of some "intelligence" that analyzes situations and deduces *automatic workflow adaptations* at run-time. This can result in workflow instances where workflow activities that potentially fail are automatically substituted by other activities (representing *predictive adaptation* [9]). Another example is *automatic repair* of logical failures (or *reactive adaptation* [9]) in workflow instances through the utilization of planning and problem solving techniques.

In order to achieve such kind of behavior, *adaptive* WfMSs need to be able to 1) detect failures and change stimuli 2) determine necessary adaptations 3) identify the workflow instances to be adapted 4) introduce corresponding changes to them and 5) notify respective users [11]. Existing approaches to realizing such behavior include *rule-based*, *goal-based* and *process-driven* approaches [11]. In contrast to anticipated changes, *unanticipated changes* to workflow instances require adaptive WfMSs to *involve users* for resolving situations. In such situations, typical modifications include the *addition, deletion* or *reconfiguration* of elements within running workflow instances, but also *skipping* or *re-iterating* through parts of the workflow instance. Available approaches addressing this issue typically engage users in some sort of graphical dialogue in order to make communication of changes between the user and the WfMS easier. Ensuring correctness of the workflow instances after the changes have been applied represents a critical factor for such concepts. Other strategies for dealing with unanticipated changes on a workflow instance level include concepts such as *ad-hoc planning* [6], *late modeling* or *late-binding* of workflows. Further strategies for dealing with the propagation of changes from type to instance levels include *lazy, eager, selective* or *upward* propagation as well as merging [7].

In addition, changes on the workflow instance level can be used to feed change information back to the type level, thereby exhibiting *self-learning* and *self-optimization* abilities through pattern matching and reuse of knowledge. *Process mining* [13] for example represents a technique that aims to analyze data from both, past workflow *executions* and *adaptations* to automatically construct or improve workflow type definitions. Analyzing past workflow adaptations takes place by analyzing change logs that comprise information such as *reasons for change, types of change* and *context of change* [14].

To a certain extent, such adaptive WfMSs can be regarded as exhibiting autonomic behavior. As noted in [3], autonomic systems represent "*an evolution, not a revolution*" that increases the level of autonomy in autonomic systems *incrementally*, thereby combining manual and autonomic behavior on its way to achieving the vision of fully autonomic systems. *But how can these different degrees of autonomy be conceptualized in the context of WfMSs?* To address this question, we will introduce an incremental model for autonomic WfMSs.

# 2 Towards Autonomic WfMSs

Based on the five levels of autonomic computing introduced by [10], Table 1 gives an introduction to five distinct levels representing different degrees of autonomy in WfMSs. In the following, we will describe each of the levels in some greater detail.

**Basic Level:** This level represents the starting point for WfMSs. Definition, analysis and adaptation rely solely on the workflow designer's abilities and is only marginally supported by for example syntax checkers that ensure syntactical correctness of defined workflows.

**Managed Level:** On a managed level, the WfMS monitors workflow instances and provides runtime statistics. This aids workflow designers in exploring the need for workflow adaptations and in optimizing certain aspects of workflows.

**Predictive Level:** The predictive level provides techniques and tools for process mining and simulation. This aids workflow designers in deciding upon change by having a set of workflow definition alternatives readily available (e.g. workflow mining) and being able to predict the effects of different alternatives. Thereby, workflow designers can optimize their workflows in terms of their goals.

**Adaptive Level:** On this level, "some intelligence" provided by the autonomic system (such as self repair or self-optimization behavior) supports workflow designers in acting upon change. Certain actions are executed autonomously, while other actions still are in control of the workflow designer (representing a combined directive agent). This in turn aids in aligning the WfMS to the business goals of the organization.

| | Autonomic Workflow Management Systems | | | | |
|---|---|---|---|---|---|
| | **Basic Level 1** | **Managed Level 2** | **Predictive Level 3** | **Adaptive Level 4** | **Autonomic Level 5** |
| Key aspects | Manual workflow definition, analysis and adaptation | Automated workflow analyses | Automated workflow simulation | Automated workflow adaptation | Goal-oriented workflow governance |
| WfMS techniques and tools | Syntax checker | Workflow statistics and monitoring | Workflow mining and -prediction | Workflow repair and -optimization | Goal-based WfMS configuration |
| System support for | Syntactical correctness of change | Exploration of change | Decision upon change | Action upon change | Goal evolution |
| Aligned with | Workflow specification | Local optimizations | Workflow goals | Business goals | Strategy |
| Directive Agent | Workflow designer | Workflow designer | Workflow designer | Combined | Autonomic WfMS |

Table 1 Five Levels of Autonomic Workflow Management Systems

**Autonomic Level:** On the autonomic level, the WfMS autonomously adapts its workflow types and instances according to its assigned goals and its knowledge about the environment and itself. Interaction with the WfMS takes place on a goal level. This high level of abstraction aids in aligning workflow definitions and -instances with strategies of organizations.

When reflecting upon this categorization, traditional workflow management systems can largely be regarded to represent level 1, 2 and (occasionally) 3 autonomic WfMSs. Current research on workflow mining and adaptive workflows powerfully demonstrate the prospects of predictive (level 3) and adaptive (level 4) WfMSs through the availability of prototypes such as the *ProM Framework* [12] or *AgentWork* [9]. In addition to these, first prototypical implementations of self-tuning, self-configuration and self-healing behavior in WfMSs are already on the way (as introduced by e.g. [4]). The emergence of dedicated conferences on autonomic computing fuels expectations of further research in this area.

In analogy to autonomic system administration, it can be expected that having autonomic WfMSs available would *decrease perceived complexity* of workflow management for workflow users and *lower the work burden* for workflow designers. It could *shift human efforts away* from low-level workflow design towards the definition of and reasoning about goals that represent criteria for the selection of a set of autonomously generated workflow alternatives.

# 3 Conclusions

This paper explored the question "*Can the principles of autonomic computing be applied to workflow management – and if so, how?*"

To the best of our knowledge, this contribution represents a first attempt towards a roadmap to autonomic WfMSs by introducing a conceptualization of different levels of autonomy in such systems. Even if the developed vision of autonomic WfMSs may seem to be too ambitious at first, we aimed to demonstrate that a stepwise approach seems to be feasible by illustrating how certain levels of autonomy can already be achieved with available techniques.

We expect the conceptualization of autonomic WfMSs to support the assessment of different levels and degrees of autonomy in available workflow management systems. Therefore, further work should focus on applying the framework to existing workflow management systems (such as IBM's WebSphere MQ Workflow [1]) and prototypes. In order to be practically applicable, a finer grained characterization of each level might need to be necessary, relating the different levels and capabilities to the elements of, for example, the WfMC reference model [5].


## Acknowledgements

The research of this contribution is funded in part by the Know-Center Graz (within the Austrian Competence Center program Kplus), the FWF Austrian Science Fund (within an Erwin Schrödinger Fellowship Program) and the Natural Sciences and Engineering Research Council of Canada. We would like to thank Alexei Lapouchnian for early fruitful discussions regarding this contribution.


## About the Authors

**Markus Strohmaier** is a postdoctoral fellow at the Department of Computer Science at the University of Toronto. He holds a Ph.D. in Computer Science from Graz University of Technology, Austria. His research interests include business process- and agent oriented modeling, information systems design and knowledge management.

**Eric Yu** is an associate professor at the Faculty of Information Studies at the University of Toronto. He received his Ph.D. in Computer Science from the University of Toronto in 1995. His interests are in the areas of information systems design, requirements engineering, knowledge management, enterprise architecture, and software engineering.

## References


[1] IBM WebSphere MQ Workflow Software, http://www-306.ibm.com/software/integration/wmqwf/, last accessed on May 31, 2006.

[2] M. Dumas and W. M. van der Aalst and A. H. ter Hofstede (eds). Process-aware information systems: bridging people and software through process technology. John Wiley & Sons, Inc., New York, NY, USA, 2005.

[3] A.G. Ganek and T.A. Corbi. The dawning of the autonomic computing era. IBM Systems Journal, 42(1):5–18, 2003.

[4] T. Heinis, C. Pautasso, and G. Alonso. Design and Evaluation of an Autonomic Workflow Engine. In Proceedings of the Second International Conference on Autonomic Computing, 2005. ICAC 2005., pages 27–38, 2005.

[5] D. Hollingsworth. Workflow Management Coalition (WfMC) - The Workflow Reference Model. Technical report, Workflow Management Coalition, Jan 1995.

[6] C. Huth, S. Smolnik, and L. Nastansky. Applying Topic Maps to Ad Hoc Workflows for Semantic Associative Navigation in Process Networks. In Proceedings of the Seventh International Workshop on Groupware (CRIWG'01), 2001.

[7] G. Joeris and O. Herzog. Managing Evolving Workflow Specifications. In Conference on Cooperative Information Systems, pages 310–321, 1998.

[8] J. O. Kephart and D. M. Chess. The Vision of Autonomic Computing. Computer, 36(1):41–50, 2003.

[9] R. Mueller. Event-Oriented Dynamic Adaptation of Workflows: Model, Architecture and Implementation. PhD thesis, University of Leipzig, 2002.

[10] R. Murch. Autonomic Computing. Information Management. IBM Press, 2004.

[11] S. Rinderle, M. Reichert, and P. Dadam. Correctness criteria for dynamic changes in workflow systems: a survey. Data & Knowledge Engineering, 50(1):9–34, July 2004.

[12] Open Source. ProM - Framework for Process Mining. http://sourceforge.net/projects/prom, last accessed on May 31st, 2006.

[13] W. M. P. van der Aalst, B. F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. J. M. M. Weijters. Workflow mining: a survey of issues and approaches. Data & Knowledge Engineering, 47(2):237–267, 2003.

[14] W.M.P. van der Aalst, C. Günther, J. Recker, and M. Reichert. Using Process Mining to Analyze and Improve Process Flexibility. In Proceedings of CAISE'06 - The 18th Conference on Advanced Information Systems Engineering, 2006.