

A Research Agenda for Business-Driven Information Technology

Jeffrey O. Kephart and Steve R. White
IBM Thomas J. Watson Research Center
Hawthorne, New York 10532, USA
{kephart, srwhite}@us.ibm.com

Edith Stern
IBM Software Group, Tivoli
Somers, New York 10589
estern@us.ibm.com

Abstract

On Demand Computing is a popular vision of the future in which businesses will respond nimbly to new opportunities and threats. Unfortunately, the ever-growing complexity of IT is a key inhibitor of this vision, as it raises the cost and risk of altering systems, rendering them ever more ponderous. Since the purpose of autonomic computing is to reverse the trend of increasing IT complexity, it is a critically important enabler for On Demand Computing, or business-driven IT. In this paper, we situate autonomic computing within the broader context of business-driven IT, and use the resulting picture to motivate and discuss a research agenda that we and our colleagues at IBM have begun to pursue.

1 Introduction

On Demand Computing [17] envisions a world in which businesses will be much more flexible and responsive in the face of ever-changing customer demand, business opportunities and external threats. Armed with the right tools, business people will be able to redesign business processes and redefine business priorities, and their directives will quickly (and to a large extent, automatically) percolate down to the IT system infrastructure, driving appropriate system reconfigurations, application (re-)deployments, IT policy changes, and IT component adaptations. Vendors are moving towards this vision of business-driven IT by offering products that provide improved business performance monitoring and decision support. However, the ever-growing complexity of IT makes changing systems more costly, time-consuming, and risky, and is therefore a major impediment.

Clearly, autonomic computing, which seeks to enable systems and their components to manage themselves in accordance with high-level objectives, is a critically important enabler for On Demand Computing. In this paper, we present a simple picture of how we believe autonomic computing will support this vision, and use that picture to assess

the research community's current state of progress and lay out a research agenda for autonomic, business-driven IT.

2 The Big Picture of Business-Driven IT

Figure 1 presents our high-level view of business-driven IT, representing functions and flows that will enable business objectives and business process models to be driven through a series of automated transformation steps, culminating in a fully-deployed IT system that carries out the business intent. The top box represents the business level, the bottom box represents the IT level, and the middle box represents a set of transformations between the business and IT levels. The arrows represent flows of information.

At the very top level (Business Process Tools and Transforms), users with knowledge of business process models and business objectives will use various tools to create representations of those business process models and objectives, and they will receive a high-level view of the status of their business in terms of those models and objectives. Business processes such as accounts receivable or order entry will be modeled abstractly, without specifying how they are to be implemented in the IT infrastructure. They will be expressed as discrete steps (which may be atomic, or compositions of finer-grained business sub-processes) that flow from one to the next sequentially [6]. Dependencies among the various steps and among the various sub-processes and processes, represented as Petri Nets [21] or in other forms, will be an important part of the abstract representation. Business objectives (e.g. "Be capable of processing 1000 insurance claims per day") will be captured at this level, and will be converted into expressions that will be monitored, with violations or other high-level health indicators made clearly visible via a business dashboard.

At the next level down (Business-to-IT Tools and Transforms) will be mechanisms that transform the platform-independent models into explicit plans for deployment of applications that drive the business processes onto a chosen set of resources. There will also be mechanisms that convert the business-level objectives into a) a form that guides

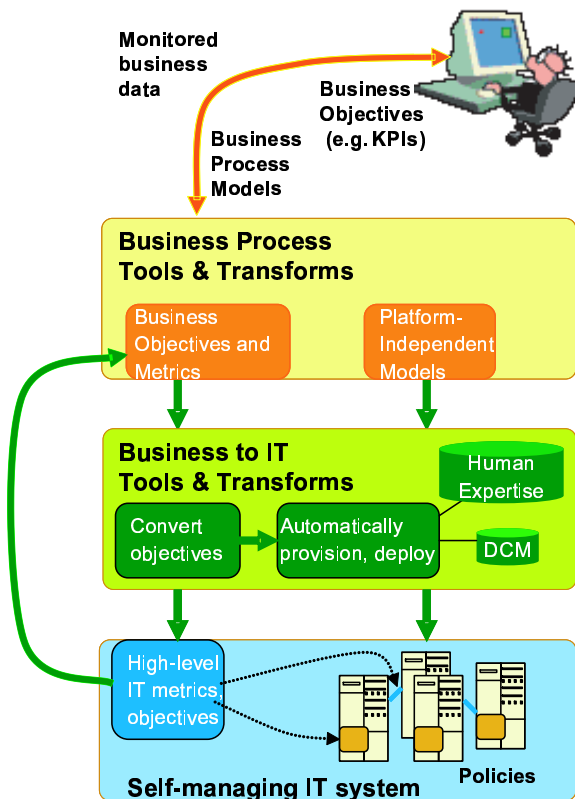


Figure 1. Business-driven IT: the big picture.

the initial provisioning and deployment of resources and b) high-level IT objectives that will guide the run-time operation of the deployed business applications.

At the bottom level (Self-Managing IT System), the deployment plan derived at the middle level is executed, and the set of applications associated with the business process is installed on the designated platforms, along with any required operating system, middleware, etc. Then, as the system runs, its behavior will be guided by the high-level IT objectives that were derived in the “Convert Objectives” step in the middle layer. The high-level IT objectives will be oriented around distinct management disciplines, such as performance, power, availability, and security. They may be transformed in a subsequent step into lower-level, resource-specific and/or platform-specific policies.

The bottom level is what we traditionally think of as an autonomic computing system: one that behaves in accordance with high-level objectives specified by humans [12]. However, Fig. 1 clarifies that autonomic computing is not merely an end unto itself; it is also a means towards an even loftier goal—making businesses much more flexible and resilient than they are presently. Another point brought out by Fig. 1 is that the high-level objectives referred to in the standard definition of autonomic computing are not necessarily

specified *directly* by humans. Instead, they may be derived indirectly from higher-level (business-level) objectives.

3 Assessment of progress to date

How close are we to attaining the vision illustrated in Fig. 1? Let us start from the top. Business process integration and management (BPIM) is already an active field of investigation within academia and industry [21, 6]. In June, 2005, a new OMG standards effort, the Business Modeling and Integration Domain Task Force (BM&I DTF), was formed [19, 3], with roughly two dozen companies participating. Among the participants are IBM and BEA Systems, Inc., which offer tools to help elicit and express business processes. BM&I DTF is also developing specifications in support of business performance monitoring and measurement, intended to introduce some coherency and consistency across the many tools that vendors are developing¹ to capture business objectives (sometimes referred to as *Key Performance Indicators*, or KPIs) and monitor the degree to which those objectives are being met. In summary, while much work remains to develop the necessary tools and transforms required to capture, represent, and use business process models and objectives, this problem is receiving so much attention from academia, industry, and standards groups that we expect to see significant progress in this emerging area over the next few years.

Within the middle level of Fig. 1 are technologies that are critical to automated provisioning and deployment. Tools such as these are responsible for mapping abstract business process models into their realizations in a given IT infrastructure [14]. Over the last few years, our colleagues have explored several technologies that occupy the “Automatically provision, deploy” box in Fig. 1 [14, 10, 5]. They take as input a description of available resources, application characteristics and constraints, and captured human expertise, and through various planning and scheduling techniques they produce a description of the type of resources needed and how they should be connected, and an explicit workflow for deploying these resources. A provisioning manager² then executes the workflow, resulting in a deployed system. Much remains to be done, but we anticipate good progress in this space in the near term.

However, surveying the current state of the art, we identify several important things that are presently missing from the middle level. These include:

- The arrow flowing from platform-independent models to business-to-IT tools and transforms,
- The “Convert objectives” box,

¹IBM’s entry is WebSphere Business Integration Monitor.

²Tivoli Provisioning Manager, in IBM’s case.

- The arrow flowing from “Business Objectives and Metrics” to “Convert objectives”, and
- The two arrows flowing out of the “Convert objectives” box to the “Automatically provision, deploy” and “High-level IT Metrics and Objectives” boxes.

As discussed previously, the bottom level is the autonomic computing level. Somewhat detailed discussions of the current state and research challenges of autonomic computing appear elsewhere [12, 11]. Here, we focus narrowly on the one aspect that is featured in Fig. 1: high-level IT metrics and objectives.

One significant trend is that some products (mainly workload managers) are acquiring the capability to manage themselves in accordance with higher-level objectives, e.g. response-time goals or utility functions. Examples of products that manage to response-time goals include the IBM Enterprise Workload Manager [1] and the HP-UX Workload Manager [16]. IBM’s WebSphere Extended Deployment middleware product seeks to maximize utility functions based on average- or percentile-response time metrics [15]. This is a significant advance over the common practice of imbedding low-level if-then policies or rules in specific components (e.g. “If average CPU utilization rises above 80%, add another server to the tier”). We anticipate that, over time, products will branch out from response-time goals to other types of goals in the realms of availability, etc.

A parallel trend is a movement from resource- and platform-specific management towards management by discipline (e.g. performance, availability, security, etc.), as reflected in the ITIL (IT Infrastructure Library), a series of documents that constitute the world’s most widely accepted best-practices approach to IT service management [20]. It seems clear that this shift can only be successful if there is a concomitant shift from platform- and resource-specific policies (which govern platform- and resource-specific management) to policies that are oriented around management disciplines. Taking a management-discipline-oriented approach to policy necessitates elevating the treatment of policy above the level of if-then, and up towards goal-oriented or utility-based methods of specifying objectives.

Putting these two trends together, we anticipate that present-day low-level, resource-specific policies will be phased out in favor of high-level objectives expressed in terms of performance, availability, power, and security concerns. These several objectives will be brought together in one place, where they can be compared, reasoned about, argued about explicitly in a rational and informed way by IT users, and combined into a single set of objectives with explicit tradeoffs expressed in some form. This set of objectives will drive the adaptive, self-managing behavior of the system in the face of workload fluctuations, faults, and other environmental conditions and variations. Especially

in cases where efficiency is important and the cases are enumerable, they will be compiled automatically into lower-level, resource-specific policies.

Thus far, there has been little work on eliciting, representing and using high-level objectives that are oriented around multiple management disciplines such as performance, availability, and security³. Therefore, we add to our research agenda the need to develop such technologies. Moreover, the feedback loop from the high-level IT objectives back up to the business-level objectives, which ensures that the business-to-IT metric transformation is working properly, is receiving very little attention at present.

4 A research agenda for business-driven IT

According to the assessment just presented, technologies supporting some of the functions depicted in Fig. 1 are making good progress, while others are either in their infancy or not being attended to at all. In this section we focus on some of the gaps that surfaced from our analysis.

4.1 Self-managing IT level

First, let us begin at the bottom level. In section 3 we asserted that the IT Services Management paradigm represented by ITIL requires a shift from resource-specific policies to management-discipline-specific objectives. Among the science challenges driven by this shift are:

- How can we best elicit high-level objectives and tradeoffs among them from IT administrators, who are completely unaccustomed to interacting with systems at this level? Solving this problem is likely to entail leading-edge work that combines algorithms with innovative human-interface studies and designs.
- How can we use these objectives to govern runtime behavior of systems? Most likely we will need to capture or learn models of system behavior—what tools or methods are most effective?
- How can we best build trust between adaptive systems driven by high-level IT objectives and administrators, and ensure that systems can operate in a semi-automated mode?

To make these challenges somewhat more concrete, consider the following scenario, which stays purely at the IT level. Suppose an IT administrator is assembling an eBrokerage transactions application. A plausible set of informal objectives existing inside the mind of the administrator might look like the following:

³There *has* however been some work on managing to multiple high-level *performance* objectives [18].

I'd like a 1-second average response time for Gold customers. 0.75 sec would be a little better, but any faster won't make much difference. A response time of 2.0 sec is tolerable, but more than 3.0 sec is not. Silver class is just the same, except the time scales are a factor of two slower.

I can live with the system being down for an hour per month, but I'd really like 10 minutes or less. More than 2 hours per month is intolerable.

Given a choice between good performance and mediocre downtime, or mediocre performance and good downtime, I'd pick the latter.

The first challenge is to convert these informal notions of desire into a concrete representation. One plausible approach is to adapt preference elicitation techniques that have been developed for e-commerce [7, 9] to infer a utility function. In this case, the utility function $U(R_G, R_S, D)$ would represent, for each possible triple of (gold response time, silver response time, monthly down time), a value on some chosen scale (perhaps monetary). Examples of such utility functions are provided in references [13] and [18].

The second challenge is to use that multi-objective representation to drive system behavior. For example, suppose the multi-objective representation is $U(R_G, R_S, D)$. One plausible approach to driving system behavior is to use models of how the IT metrics like response time and down time depend on any system or workload variables that can either be controlled or observed. For example, control variables might include the amount of CPU and memory that can be devoted to gold and silver classes, the frequency with which backups are run or checkpoints are taken, the number of concurrent threads, the number of hot standbys, etc. Observable (non-controllable) variables may include the total number of servers and the number of requests per second. A model can be thought of as a simple functional relationship between a service level variable and the control and observed system variables, e.g. $R_G(\text{CPU}_G, b; \lambda)$ might represent the dependence of the average gold response time on the number of CPU cycles per second devoted to gold class, the backup frequency b , and the average number of transactions per second, λ . The model could represent knowledge captured from a human expert, or it could be derived automatically by fitting queuing models to observations, or via machine learning techniques. The models can be substituted into the expression $U(R_G, R_S, D)$ to obtain an expression for U in terms of the system variables $\text{CPU}_G, \text{CPU}_S, b$, etc. Then, an appropriate optimization routine (e.g. a derivative-free nonlinear optimizer) can be used to determine the optimal setting of the control variables—the one that maximizes U . As the environment changes (e.g., the workload λ varies) the optimal setting of control variables will change, so the optimizer will run pe-

riodically, resetting the control variables as needed.

These two challenges establish a neat and necessary separation between *articulating* objectives and *meeting* them. While there may be other approaches that satisfy this property, the utility-based mechanisms proposed here support that separation by first establishing an objective function (the utility function) and then using a combination of modeling (possibly augmented by learning) and optimization to maximize that objective function. Lest this seem trivial, note that the prevalent resource-specific policy approach, which frequently relies on if-then rules like “IF CPU utilization exceeds 90% THEN add server”, does not clearly articulate what is desired (good response time), and fixes on a particular way of executing that non-articulated objective that becomes outdated whenever objectives or technologies change, and is not reusable across deployments. Capturing human expertise in the form of models rather than rules renders that knowledge more generally useful, longer-lasting, and easier to refine via machine learning.

The third challenge is to support a natural interplay between automated and manual management until the user is willing to turn over management completely to the automated management system. For example, the system might simply *recommend* a setting of control variables to a human system administrator, allowing the administrator to permit the proposed change or not. In a more sophisticated version, the system could propose multiple settings and allow the administrator to select one, or input a different one.

4.2 Business process and business transformation levels

The challenges become even greater as we move up towards the business level. In general, the main challenges can be inferred from the bulleted list in section 3, which center on the most of the arrows connecting the three levels to one another (except the one pointing from “Automatically provision, deploy” to the IT level, which is under much study), plus the box that converts from business-level to IT metrics. Developing the feedback loop from IT level metrics up to business level metrics constitutes another challenge. We summarize all of these challenges as follows:

- To what extent can we transform and/or augment the platform-independent models so as to be consumed directly by automated provisioning and deployment technologies? Today, the inputs to the automatic provisioning box are created manually via a number of different tools, greatly slowing down the process of creating or modifying the IT configuration.
- How do we convert business objectives to IT-level objectives in such a form that they can a) help determine the original configuration of the deployed sys-

tem and b) guide the run-time operation of the system? Promising early work in this area includes work by Bartolini [2] at Hewlett Packard. Statistical learning techniques such as those described in [4, 8] may play an important role.

- How do we use feedback from the IT level to improve the transformations from business to IT objectives? If statistical learning underlies the converter, then this feedback would naturally be used to drive the learning.

5 Conclusions

Business-driven IT would bring great rewards. Businesses would be able to deploy and configure IT systems much more quickly and automatically than they can be today, and more flexibly reconfigure IT systems as business needs and opportunities arise. The time required to implement significant new business processes would drop from months to days, dramatically increasing the responsiveness of businesses to changes in their tactical direction. By closing the loop and permitting business people to monitor the operation of their business in business terms rather than IT terms, the business impact of such things as changes in order volume can be assessed immediately. Businesses that can detect changing business conditions quickly and respond to new opportunities quickly are likely to have a distinct competitive advantage over those that cannot.

We hope that the framework presented in this paper will help nucleate and inspire development of technologies that are necessary to support autonomic, business-driven IT, and ensure that they are not developed piecemeal, but within a coherent framework, which we believe is essential to the ultimate success of this undertaking.

6 Acknowledgments

The authors thank many colleagues at IBM for useful discussions that have influenced their thinking on these topics, especially Pamela Durham and Dinesh Verma, who worked closely with us in developing these ideas.

References

- [1] J. Aman, C. Eilert, D. Emmes, P. Yocom, and D. Dillenberg. Adaptive algorithms for managing a distributed data processing workload. *IBM Systems Journal of Research and Development*, 36(2), 1997.
- [2] C. Bartolini, M. Salleé, and D. Trastour. IT service management driven by business objectives - an application to incident management. In *Proc. 10th Network Operations Management Symposium*, 2006.
- [3] bmi.omg.org.
- [4] D. Breitgand, E. Henis, and O. Shehory. Automated and adaptive threshold setting: Enabling technology for autonomy and self-management. In *Proc. 2nd Int'l Conference on Autonomic Computing*, 2005.
- [5] T. Eilam, M. Kalantar, A. Konstantinou, and G. Pacifici. Model-based automation of service deployment in a constrained environment. In *Proc. Int'l Symposium on Integrated Network Management*, 2005.
- [6] Y. Huang, S. Kumaran, and J.-Y. Chung. A model-driven framework for enterprise service management. *Information Systems and E-Business Management*, 3(2):201–217, 2005.
- [7] V. Iyengar, J. Lee, and M. Campbell. Evaluating multiple attribute items using queries. In *Proc. 3rd ACM Conference on Electronic Commerce*, pages 144–153, 2001.
- [8] M. Karlsson and M. Covell. Dynamic black-box performance model estimation for self-tuning regulators. In *Proc. 2nd Int'l Conference on Autonomic Computing*, 2005.
- [9] R. L. Keeney and H. Raiffa. *Decision with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press, 1993.
- [10] A. Keller, J. Hellerstein, J. Wolf, K. Wu, and V. Krishnan. The CHAMPS system: Change management with planning and scheduling. In *Proc. 9th Network Operations and Management Symposium*, 2004.
- [11] J. O. Kephart. Research challenges of autonomic computing. In *Proceedings of the 27th International Conference on Software Engineering*, pages 15–22, 2005.
- [12] J. O. Kephart and D. M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–52, 2003.
- [13] J. O. Kephart and W. E. Walsh. An artificial intelligence perspective on autonomic computing policies. In *Proc. 5th IEEE Int'l Workshop on Policies for Distributed Systems and Networks*, pages 3–12. IEEE Computer Society, 2004.
- [14] J. Koehler, R. Hauser, S. Kapoor, F. Y. Wu, and S. Kumaran. A model-driven transformation method. In *Proc. 7th Int'l Enterprise Distributed Object Computing Conference (EDOC'03)*, pages 186–197. IEEE Press, 2003.
- [15] G. Pacifici, M. Spreitzer, A. Tantawi, and A. Youssef. Performance management of cluster based web services. *IEEE Journal on Selected Areas in Communications*, 23:2333–2343, 2005.
- [16] H. Packard. HP-UX workload manager user's guide, part no. b8844-90010. Technical report, Hewlett Packard, 2006.
- [17] J. G. Spooner and S. Junnarkar. IBM talks up 'computing on demand'. *ZDNet*, 2002.
- [18] W. E. Walsh, G. Tesauro, J. O. Kephart, and R. Das. Utility functions in autonomic systems. In *First International Conference on Autonomic Computing*, 2004.
- [19] www.bpmi.org.
- [20] www.itil.co.uk.
- [21] J. Zhu, Z. Tian, T. Li, W. Sun, S. Ye, W. Ding, C. Wang, G. Wu, L. Weng, S. Huang, B. Liu, and D. Chou. Model-driven business process integration and management: A case study with the Bank SinoPac regional service platform. *IBM Systems Journal of Research and Development*, 48(5/6):649–669, 2004.