RZ 3463 (# 93564) 11/18/02 (Revised 01/19/03)
Computer Science 15 pages

Research Report

Autonomic Economics

Why Self-Managed e-Business Systems Will Talk Money

Giorgos Cheliotis and Chris Kenyon

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland
{gic,chk}@zurich.ibm.com

Autonomic Economics
Why Self-Managed e-Business Systems Will Talk Money

Giorgos Cheliotis and Chris Kenyon
IBM Research, Zürich Research Laboratory
8803 Rüschlikon
Switzerland.
{gic|chk}@zurich.ibm.com

19 Jan 2003. Draft: 1.2

Abstract

Autonomic systems take decisions independent of human administrators and hide much system complexity when they do signal for intervention. Furthermore, they are expected to interact with their environment in an autonomous way, even beyond the boundaries of an individual organization. Autonomic systems will offer their capabilities as services partially based on service elements they acquire from their environment. In a commercial environment autonomic decisions, e.g. (re-)configuration, healing and anticipation will not be taken "at any cost". Financial factors will be combined with technical feasibility to obtain optimal outcomes. In effect, the maximum financial value of a system will be the result of an optimal operating policy and vice-versa. We term this Autonomic Economics. Here we interpret the Autonomic Manifesto [IBM03] from an economic point of view demonstrating how financial criteria complement and enrich other technical aspects of autonomic systems. We further argue that the inclusion of economic criteria in the autonomic decision process will not only support self-management but also facilitate the communication of decisions and their trade-offs between the system and the administrator.

Keywords: Autonomic, self-managed, economics, optimization, valuation, resource management, grid computing

1 Introduction

Web technologies and the emerging service-oriented Grid middleware are simplifying access to information and computing resources respectively. However the IT systems which support our daily computational tasks are constantly increasing in complexity, often making their management a very costly, time-consuming task. This complexity is amplified in the case of heterogeneous multi-tier distributed systems commonly used in industry today. Autonomic systems hold the promise of being self-aware systems that configure and reconfigure themselves when faced with anticipated and unanticipated changes in themselves, their tasks, and in their environment.

Configuration, in the autonomic sense, includes relations with other systems as well as a system's own internal state. In short, autonomic systems take decisions. We argue that economics will form a major part of many of these decisions, especially for systems which will support/offer commercial services: system optimization and configuration is not "at any cost" but will be optimized in dollar terms as well, relative to the system's knowledge of its objectives.

In this article we interpret IBM's Autonomic Manifesto [IBM03] — the beginning of an effort to create a commercial framework and common research agenda for the development of self-managed systems — from an economic point of view, demonstrating how financial criteria complement and enrich other technical aspects of autonomic systems.

At these early stages of development we do not aim to provide any definite recipes for the design of self-managed systems with economic reasoning. Our purpose is to challenge current beliefs of what a system is and what it can or should do as well as to motivate research and development work targeting the inclusion of economic criteria in a self-managed system's decisions. We envision a fundamental change in how we view IT systems: in the future a system will not be simply an assembly of its parts, but rather a software entity endowed with the capability to evolve and acquire or release components as it sees fit, in order to meet high-level user objectives. E-commerce technologies will play a strategic role in realizing this vision, as they enable the automated exchange of assets between not only human users but also between systems.

The Autonomic Manifesto [IBM03] gives seven characteristics of autonomic systems that we can group into three that deal with market context and identity (identity, environment, standards) and four others that consist of situations and decisions (self-optimization, re-configuration, healing, anticipation). In the following we first discuss the economic primitives of autonomic systems, i.e. identify the market agents, the goods they can trade, their initial endowments, etc. Then we iterate through required properties of an autonomic system, often quoting from the autonomic manifesto, and in every case explain how to apply economics to support a system's decision process. We then present two examples illustrating potential applications of autonomic economics. We conclude with a summary of the main benefits of applying economics within autonomic systems.

2 Background

The need for self-managed systems has been recognized by academia, and in part by industry, with several initiatives underway (see [IBM03] for a list). In some of these efforts, as well as in the Autonomic Manifesto itself parallels are drawn to electrical utilities, often implying that IT services will also be priced and traded as utilities, by users and systems alike. However, to the best of our knowledge the analogy has not been sufficiently exploited (if at all) by the self-managed systems community.

However, the use of economic mechanisms for distributed resource allocation has been the main subject of study of another — not so remote — research community for more than three decades now, in what is sometimes termed "market-based control". This particular community has been focusing mostly on the design of artificial markets for regulating resource contention, see for example [BGA01, CDS00, DGBS00, Sut68, RN98, SDK+94]. One of the motivations for this has been the development of adaptive and resilient IT systems. These efforts have gone so far as to prove that the economic paradigm is a valid one for the allocation of IT resources. However, hav-

ing focused mostly on auction design and resource exchange protocols, they have not yet provided the sophisticated decision-making machinery required for autonomic systems to behave truly independently. Also, the aim in many cases has been to contain such marketplaces to the limited world of an artificial economy, built solely for matching tasks to resources or for congestion control, not for trade. This limitation has been used both as a justification and an excuse for the limited market functionality and most notably the software agents' simplistic (in economic decision terms) design in most prototypes to date. Nevertheless, there has been some solid theoretical work on mechanism design and network pricing [Sai97, San00, Sem99, Var, WMMR].

Some of the problems associated with short-sighted agents have already been uncovered and documented [GK99], but one just needs to look at the sophistication of commodity and financial markets to understand that current efforts in the Computer Science field fall short of providing us with IT systems that can partake intelligently in a real economy. Our aim is to show that future efforts need to build the appropriate level of sophisticated economic reasoning into autonomic systems that will allow them to go beyond the mostly solved issues of how to participate in an auction. To act independently and intelligently in a real e-business environment autonomic systems will need to learn about asset-liability management, investment planning and risk management.

This work is partially based on previous work of the authors on the valuation of network services [CK01, KC01], as well as recent insights on how to build commercial services on top of Grid middleware technology [CK03, KC02].

3 Economic Primitives

To set the stage for later discussions we first introduce the basic economic primitives for autonomic economics, i.e. how we envision the market players in this environment, the resources to be exchanged, etc.

3.1 Market Agents

The market players will be the IT systems themselves. More precisely, a kind of planning and trading agent will be used for every autonomic system. This agent will be in charge of acquiring or releasing IT assets depending on the system's needs, available budget and the actions of other such agents. Human administrators of these systems will endow them with a budget which the systems themselves will manage. Generally agent-based trading is a rather old idea and so is the idea of artificial economies, but the novelty lies in the proposal to use such agents in the heart of self-managed systems, endowing them with real money (or a "token"-type internal currency for which an exchange rate can be defined).

3.2 Assets

The traded assets, i.e. the commodity space for autonomic systems will comprise:

- access services (computing, storage and networking)
- higher-level services, e.g. data replication, intrusion detection, web services
- utility services (better than plain access services, on-demand and with quality of service)
- spot (immediate delivery) and forward contracts (future delivery), as well as derivative products, e.g. options to acquire extra storage units at a fixed premium

Some assets may be fully commoditized and this will greatly ease the execution of commercial transactions because then only quantity and price matters for exchange. For non-standard assets some technical features will need to be published or negotiated before any exchange takes place, but in effect this does not change much for the autonomic systems that needs to acquire certain resources. It will just have to check an asset's specification against its own requirements before deciding on quantity and price.

3.3 Market Organization

In a market populated by autonomous agents any conceivable market organization model can be considered. Typically for facilitating exchange intermediaries are needed (brokers, exchanges, market-makers) which results in a centralized hierarchical market-structure. But for autonomic systems we could also imagine a peer-to-peer, i.e. flat market structure which perhaps better suits the nature of such systems. Each system is responsible for its own resources and by adapting its configuration can act as both seller and buyer without the need for intermediaries. What it does need is a "community", i.e. a mechanism for the automatic creation of peering neighborhoods where exchange can take place. Such neighborhoods can also be defined by administrators as common pools of computational resources and services. Autonomic peering communities can then be defined per group, department, division, geographical area, etc. inside an organization as well as across organizations.

We will now start to iterate through the Autonomic Economics "wish list" of the basic properties a self-managed system should possess and examine how economic thinking and financial methods can contribute.

4 Identity and Environment

Figure 1 shows the basic components that comprise an industry-grade autonomic system in a service-oriented architecture. An autonomic system can be of any scale, from individual self-managing components to enterprise-wide service delivery platforms. The system is self-managed and continuously optimized with respect to the generated value of services it offers versus the costs of consumed services. The administrator defines directives for system behaviour and only occasionally gets notified by the system in the case of an extraordinary event.

4.1 Identity

> *[An autonomic system] needs to "know itself" and comprise components that also possess a system identity.*

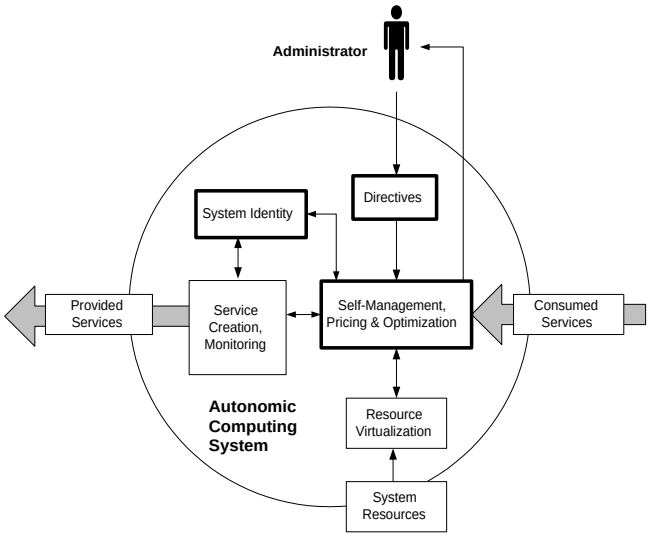

Figure 1: Service-oriented autonomic system architecture with identity and control points. Emphasized boxes denote core autonomic elements. Administrator input is provided in a set of directives (rules). Some of these rules will specify in which cases the administrator wants to be notified of changing conditions or of decision points in the system. Surrounding the administrator and autonomic core are functions which enable systems operation and cooperation in a service-oriented context.

4.1.1 Elements and Awareness

An autonomic system's identity is founded on its awareness of the following basic elements:

1. A *Budget* that the system should manage. Budgets can be used to constrain the decision space of a system, making some courses of action infeasible. Of course a self-managed system can increase that limit by trading profitably.

2. *Directives* imposed by the owner or administrator of the system which guide or constrain the system's behaviour. *Goals* are (optimization) criteria guiding the system's process of adapting to internal or external changes. *Policies* are rules of behaviour, defining what the system can or cannot do as it evolves. An example of a goal is: "maintain service reliability levels at 99.99% while

minimizing the cost of third-party services". A policy may be: "don't store this data with a service offered by a company that has a credit rating of less than AAA".

3. *Rights* and *commitments* which encapsulate options resp. obligations with respect to the system's interaction with its environment. For example, the system may always have an option to tap into free cycle-scavenged resources, or may have a commitment to provide a service in the future after accepting an advance reservation request. Rights and commitments refer always to actions that may or will be taken in the future as opposed to the following identity item, *services*, which refers to the present, i.e. currently provided/consumed services.

4. *Services* used or offered by the system. Every procured service has a contract associated with it which describes ownership and a service-level agreement which specifies a service profile, quality characteristics and the (contingent) payment (and penalty) structure associated with it. Services can be made available using a form of Web Services technology which is computer-interpretable and supports long-lived sessions (based on SOAP and WSDL descriptions plus OGSA extensions [TCF+02]).

5. *Internal Resources* owned by the system. These are the IT resources (hardware, software licenses, etc) endowed to the system by its administrator. Using these resources and third-party services the autonomic system is in a position to support or develop the range of services it offers. Assuming that a limited set of resources is for internal-use only is very similar to the assumption that a company needs some capital to setup and support its own operations.

6. *Internal Constraints* of the system. These are not imposed by an administrator, but are inherent to the system's design. Such a constraint may be the inability to self-manage parts of the system, which can only be changed through a user-initiated software/hardware upgrade, when this becomes available.

In autonomic terms a system has a singular identity in that there is a single logical point of control for the whole system. However, the implementation of control policies and decisions may be distributed and localized. A proposed structure for system identity is given in Figure 2.

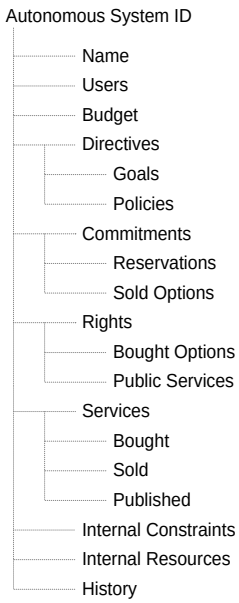

Figure 2: Proposed structure for autonomic system identity. Assets and liabilities, i.e. rights, commitments, bought/sold contracts and internal resources form an integral part of this identity. It follows that the system must know the value of these elements and be in a position to protect/maximize this value.

4.1.2 It's a business

Just as a business can be viewed as a nexus of contracts so autonomic systems can be understood in the same way. Businesses may have complex ownership and control structures and the same will be true of autonomic systems. However these will be more dynamic because decisions can be both made and carried out at electronic speed. Making the analogy between a business and an autonomic system is useful because we can then apply to these systems the same well-defined quantita-

tive criteria used for evaluating the performance of a company.

Identity, context, directives and decision possibilities are intrinsically linked to financial value. Changes in any of these will affect the value of an autonomic system. Although there are many different ways in the literature to value a going concern [CKM00, Dam01] we take a pragmatic and limited view: the value of an economically-driven autonomic system is the sum of the contingent discounted cash-flow that its services can be sold for [DP94]. We emphasize the contingent nature of this value because the value is contingent upon the decisions made by the autonomic system and its administrators (when they are called to act)[1].

From an economic point of view the job of an autonomic system is to maximize its own value within a particular context given the management directives provided using the scope contained in its decision possibilities. Thus it must be aware of its component values and the value of the system as a whole. Self-value is a fundamental concept for autonomic systems — especially given their ability to influence this via the decisions that they take.

How do you value an autonomic system? This is not simply the hardware together with software licenses just as a company is not simply plant and equipment. The value of an autonomic system lies in its ability to support business processes and the value attached to these processes. The best valuation method will always depend on the context, but as we can infer from Figure 2, value can be defined with respect to the offered services and owned resources. Resources not used for services today can be valued as option instruments since they provide the system the option to support more service requests when demanded [DP94].

It is also possible to take the view of asset-liability management. For a given autonomic system what are its assets and what are its liabilities? Again these can be considered from a service point of view. From a resource point of view ownership of hardware and software licenses, as well as subscriptions to third-party services are assets, contracts to provide services are liabilities. A key aspect for autonomic systems from this point of view is to avoid bankruptcy. This is true whether or not assets and liabilities are also denominated in monetary terms. The owner of an autonomic system can add new liabilities and new assets. The autonomic system itself must manage these responsibilities.

The inverse view from the one expressed above is also possible. From a financial point of view bought hardware, software and services are liabilities that must be paid for (e.g. they generate depreciation) whereas contracts for provided services generate income.

[1]This could also be termed a Real Options valuation of the system in that the decision possibilities, or options, are included in the valuation.

4.2 Environment and Context

> *[An autonomic system] will tap available resources, even negotiate the use by other systems of its underutilized elements, changing both itself and its environment...*

How is the identity defined with respect to the environment? This will be expressed in terms of contracts between the "system identity" and users or providers of services.

Borrowing and lending an autonomic system's own assets (hardware, software and services) is only possible from knowledge of self-value: the value of those assets to itself and to others in its environment. IT assets come in many flavors: e.g. the value of storage is actually the value of response-time, capacity, throughput and reliability, for a given set of content and under the constraints of processing and port-capacity. Remember that value is not a constant and interacts continually with the environment.

In a commercial service-oriented architecture, such as the one shown in Figure 1, interaction between systems will be more and more on a price basis. Information about the environment will then include not only technical aspects (available resources, service descriptions, etc) but also pricing information (for spot and reservation contracts).

A system with a static configuration has only limited knowledge of its surroundings, e.g. it may be aware of the IP address of a DNS server in the network or the users with access rights to the system. An adaptive system on the other hand affects its environment even when it is making only

internal changes since these changes may impact the services it provides to others. The interdependence of system and environment is even stronger in the autonomic paradigm because such a system is not only internally adaptive but also buys and sells contracts for elements beyond its own resources. This form of interdependence will lead to virtual structures very similar to the economic dependencies formed by international trade relationships.

When allowing systems a greater degree of freedom in how they shape their environment, we must engineer their interactions in such a way that potential negative effects of interdependence and complexity are mitigated. Sophisticated demand and pricing simulations are needed for this purpose at the system design stage. An example of such an effect are the distributed consequences of local failures. We have studied this in the context of network failures with the help of our Network Market Simulator [KC01] which yielded surprising and counter-intuitive insights [CK01]. Such insights will not reveal themselves to the designer who only studies isolated systems.

Another issue is how to ensure that such transactions between autonomic systems are fair. When several systems compete for limited service offerings, who should get what? A technically sound solution can be constructed with the use of auction mechanisms [WWW01], which has two benefits: firstly desired properties of the exchange can be guaranteed by the auction design, e.g. fairness and incentive-compatibility; and secondly resource pools will be managed on a transparent (price) basis providing direct input to the self-optimization procedures.

4.3 Heterogeneity and Standards

> *... an autonomic system must function in a heterogeneous world and implement open standards ...*

Coexistence and interdependence are unavoidable. Autonomic systems will build on open Internet (W3C, IETF) and Grid (GGF, Globus, OGSA) standards across a heterogeneous landscape of computing services.

4.3.1 Standardization before automation

Consider scheduling algorithms. There are many different ways to schedule actual use of resources. These may emphasize different aspects: for example one scheduler may schedule jobs by using a simple FIFO queue; another could use a priority queue with defined policies for changing priorities (e.g. earliest deadline first); yet another may perform a strict time multiplexing scheme (resources dedicated to only one application/task at a time), granting access based on price irrespective of whether the resources are actually used.

The first is a best-effort scheduler, the second is supporting differentiated quality levels and the third is purely based on forward contracts (pay for reservation). The last option is preferred from an economics perspective because it entails a clear definition of the provided service. However, the first two schedulers are often used in practice today.

Fortunately all three scheduling methods can be encapsulated in Web Services and described with XML-based (WSDL) descriptions. This allows any system to query their status and capabilities, also through an open directory service such as UDDI or the Globus MDS.

This trend of standardization of interfaces and interoperability has many parallels to the commoditization of energy assets (e.g. electricity, crude oil, etc) and IT (home PCs, network bandwidth). In the former case the goal is horizontal integration and systems automation, whereas in the latter it is market efficiency. Both aspects will be needed to some extent in order to build commercial autonomic systems which can decide on acquiring or releasing resources from/to other commercial systems.

4.3.2 Barter

Barter is perhaps possible for some autonomic systems but economic interchange is enormously facilitated by the most open standard of exchange in the world for resources and services: money. Autonomic systems will offer their services according to contracts (containing service level agreements) for money. In the same way, when they require additional resources (actually service level agreements on capabilities) to meet actual or an-

ticipated demands these will be obtained on a financial basis. Open standards for systems management, service creation, monitoring and billing will facilitate this.

Even with pure barter exchange rates are required. In fact, for many practical cases of changing service requirements, the most important exchange rate is between different times. Suppose one autonomic system has a peak demand now but expects very low demand later. That system will want to sell later underused service capacity in order to buy service capacity to meet present needs. However it does this, the dynamic environment that autonomic systems deal with requires exchangeability across time. These exchange rates will be market driven and highly dynamic.

4.3.3 Choice in a heterogeneous world

An autonomic system must be able to do two things with respect to heterogeneous services (either for acquiring them or for offering them): first it must be able to decide if the service is feasible to offer or appropriate for its own needs; second it must be able to work out an appropriate price. If it is offering different services based on its resources it must be able to map these onto each other financially as well as computationally. Failure to do so at an economic level will have just as serious consequences for the business as failure at the systems level.

Different services will not be simply comparable in financial terms. Many will require mapping multi-attribute metrics to each other and to their financial equivalence. This has been studied for some time in multi-attribute utility theory [KR76]. One of the simplest examples are the original risk/reward trade-offs of portfolios. Given metrics for risk and reward it is then possible to find the curve that dominating portfolios lie on. Depending on the autonomic system's directives (e.g. minimize risk for a given reward level) the optimal service portfolio can be automatically computed.

5 Decisions

In the first section we identified what an autonomic system consisted of and its context for action. In this section we describe how an autonomic system acts in economic terms.

5.1 Self-Optimization

> *An autonomic computing system ... always looks for ways to optimize its workings ... also considering supplemental external resources ... similar to the way power utilities buy and sell excess power in today's electricity markets.*

The system is required to (possibly continuously) optimize its resource usage and economics (cost, revenue, risk) under complex, conflicting and changing IT demands.

5.1.1 Unified control

All the elements of an autonomic system must be controllable in a unified manner. The elements of optimization are the control points together with the feedback from the service metrics (monitoring points). These system metrics may have requirements or constraints imposed by contractual arrangements for service delivery (SLA's) or by administrative policies. However, changes made through the control points must have clearly identifiable costs. Likewise the benefits and penalties for the service quality delivered must be identified.

In practice the service quality delivered may be constrained by business policies, e.g. "never breach a service level agreement" although the actual contract may only provide for, e.g. payment in kind for service outage (the current telecom service model). Within the spectrum of service delivery options the autonomic system will optimize by using the cheapest-to-deliver service support.

Costs in service delivery are not necessarily associated with the direct sum of fixed and variable costs of an installation. A much more important cost will probably be the opportunity cost of missed new business or expansion of existing services.

5.1.2 Techniques

There is a large literature on how power utilities, energy sources and physical resources are designed and run, e.g. [Lun97, SM99, TKW00, KT01]. These are often posed as valuation problems. The maximum value of a system is the result of an optimal operating policy and vice-versa. There has also been considerable industrial activity, as might be expected, with a range of firms offering commercial software for optimally running physical (electricity) generation assets (e.g. from e-Acumen, Caminus, Lukens Energy Group, FEA). This software takes into account management directives, service contracts, resources and the spot and forward prices. They offer a variety of different financial objectives that can be optimized.

The literature and commercial products in the energy and related fields offer a starting point and useful analogies for how to construct autonomic optimization software. All the current techniques take as a basic assumption that the environment is stochastic. The specifics of the future are unknown but it can be characterized statistically and stochastically. The main methods are based on stochastic dynamic programming (SDP) [BT97] and stochastic optimization [BL97]. Although stochastic control has been proposed, e.g. for control of web-server resources, in current commercial settings it appears that the scale and complexity of real control problems is beyond the techniques available with this approach. In fact even SDP experiences scaling problems and this has led to the emergence of neuro-dynamic programming (optimization) as one way to handle the dimensionality problems.

The main difference between previous work with utilities (energy, water, etc) and autonomic systems is the complexity of services offered. Previous utilities were very simple offerings whereas autonomic systems will support and create a large variety of different services. The optimization challenges for autonomic systems will also be formidable because of the range of ways to construct services. Additionally the link between underlying resources at the hardware level and service offerings is not direct or simple. However the speed and frequency with which autonomic systems can alter their configurations may help in problem formulation and solution. In any case the existing academic work and commercial software offer an excellent starting point.

5.2 (Re-)configuration

> *An autonomic computing system must configure and reconfigure itself under varying and unpredictable conditions. System configuration must occur automatically ... to best handle changing environments.*

In a commercial environment companies (and autonomic systems) will get what they have contracted (paid) for according to the SLAs they have for delivery of services. A direct consequence of commercial service offerings is a requirement for a high degree of precision in contract (or SLA) language. When IT services are not provided as a utility, as is currently common then there are often no guarantees and best effort is common. The precision and detail of current outsourcing contracts suggests that this mode of best effort working will not survive in a commercial environment.

Autonomic systems must adapt appropriately to varying and unpredictable conditions to the extent specified by the relevant contracts and directives. Now from an economic perspective we have the paired questions of what it will cost to adapt appropriately and what it is worth to do so, i.e. the market price. If a contract has not included consideration of varying load, for example, but specifies a constant load then the optimal response from the system may be to do nothing but generate an alert that a new condition is present and the interested parties may want to create a new contract to deal with it.

Re-configuration under unpredictable conditions leads to a consideration of risk and so to the price of risk and the possibility of risk transfer. Suppose a user or autonomic system wants a contract with another autonomic system to handle a varying unpredictable load. How much should they expect to pay? Let us consider four cases with decreasing degrees of predictability:

1. **Known variation, no uncertainty:** the user knows exactly how the load will vary and is certain that this knowledge is exact to the degree that he does not want to pay up front for any coverage for changes, nor does he want to be covered for any upside or downside potential. In short this presents a load versus time curve for pricing.

2. **Statistically characterized uncertainty:** the user is confident in his specification of how the load will vary and also in his specification of uncertainty. The user presents the probability distribution of the load versus time for pricing and specifies a desired QoS relative to it.

3. **Limits:** here, whilst the user is not confident of a statistical characterization of uncertainty, he is comfortable with specifying limits (versus time) within which he wants the system to handle load.

4. **No idea:** here the user wants any load handled and is not capable of any characterization.

The higher the degree of uncertainty transferred to the system providing a service the higher the risk premium for that service will be. In general, given a set of services with the same derived utility and quality characteristics, a service with a predefined (deterministic) load profile will be cheaper than one with a statistically defined load profile.

The fourth case implies unlimited service expectations and is not acceptable in a commercial environment. One way of understanding why it is not acceptable is to seek a way of pricing such a service. Since a contract with no guarantees is generally disliked by users assume the most simple case of a guaranteed service rate. The only reasonable way of covering for excess unexpected load while maintaining the same service rate is by allocating extra resources on the fly. This in turn means pooling resources indefinitely from the spot market, with each new purchase raising the market price. How much will this cost? No idea! Starting from a blank service characterization we are forced to end up with a dump pricing answer: transfer the spot price, however high. Unless, of course, the user is prepared to give up on the idea of QoS, not a very attractive proposition indeed.

Autonomic systems provide value by their ability to reconfigure as needed. This re-configuration to deal with varying and unpredictable loads has direct and indirect (opportunity) costs which depends on the degree to which this unpredictability can be (statistically) characterized. Contracts will develop the appropriate level of precision for specifying who bears what risk and how the different parties are compensated for providing specified QoS levels. Autonomic systems will have to be able to evaluate and exchange such contracts.

5.3 Self-Healing

An autonomic computing system must be able to recover from events that might cause some of its parts to malfunction.

Ideally autonomic systems should anticipate future events and that holds for failures too. Even if a particular failure event is generally unexpected, the probability of failure for a particular component can be modelled. Reliability theory is a well developed area of probability modelling [Ros00]. The reaction to a failure could be simple replacement as in RAID storage systems. For example, by maintaining multiple independently stored replicas of critical data an autonomic system will be able to continue operation smoothly when a single storage device fails. This kind of pro-active system protection is left for the discussion of anticipation later in this article in Section 5.4 and we deal here with those cases where injury has occurred and the main goal is to rapidly restore operation.

5.3.1 The healing process

Since IT systems will most probably not be fitted robotic arms to repair themselves in the near future, the natural IT-equivalent of self-healing is to let the system substitute failed components with spare resources of equivalent functionality — without involving a human in the process — at least until the failed component can be replaced. Assuming that the system has already identified

the failed component or service, such a healing process requires:

- identification of compatible alternatives, their availability and their price
- on-demand service provisioning
- a method for making optimal substitutions

The system must be aware of its environment and in particular of the availability and prices of resources and services that it could acquire if needed. On-demand provisioning is still a major challenge in IT even if a great deal of work has been devoted to it. The main difficulty lies in the insufficient degree of integration in today's distributed systems, fuelled by a shift away from vertical integration, an increase in competition and the fast pace of new technology adoption. Open standards and horizontal integration with end-to-end service quality are necessary to bring on-demand services in a fragmented world. Challenging as the first two issues may be, the main technical obstacle for an autonomic system is the third item, i.e. given a set of alternatives how can the system make a sound recovery decision?

5.3.2 Substitution

One way of handling a failure is to switch over to the system's own back-up service for the duration of the failure. But this assumes anticipation of the failure. What if we are dealing with a failure that was not anticipated in this way? In this case healing requires knowledge of what is damaged by the failure. From the economic perspective this means knowing the value of the service(s) which suffered from the failure.

In finance prices which are consistent with market expectations are often computed using substitution arguments between assets or combinations of assets. Equivalent services that autonomic systems offer will have equivalent prices. In practice the cheapest equivalent service sets the price up to the limit of the, say, capacity for which it is offered. This is known in finance terms as the "no-arbitrage" argument, because if the above statement would *not* hold for an extended period of time an arbitrary amount of risk-less profit could be possible [Hul00, Nef00]. No-arbitrage has been used extensively in the development of pricing models for commodities and financial instruments. We expect utility computing services to be no exception, although there is still much more work needed in this direction.

As explained, an autonomic system will have to find the cheapest equivalent in the case of component failure. In a market situation it would suffice to look up the market price for that component, equivalent service, or combination of services that could be used to construct a replacement for the impaired service.

5.4 Anticipation

> *When faced with a potentially dangerous or urgent situation ... our autonomic nervous system ... optimizes our bodies for a selection of appropriate responses ... Autonomic systems will deliver essential information to users with a system optimized and ready to implement user decisions.*

Imagine that an autonomic system is running applications within an intranet and internal demand for these is increasing. When this demand reaches a trigger level it must take action to prevent overload. It can either increase its service prices internally to throttle demand or alternatively buy the capability to increase its service levels externally. The system checks the price and availability of external services and then puts the choice before the business person responsible. Increasing prices internally or buying external services will both permit the system to maintain service levels. Now the business person can decide and has only that decision to make, not the technicalities of the tradeoffs involved or how to implement them.

5.4.1 Degrees of anticipation

How can an autonomic system be prepared for action? What is the cost of preparation for the benefit in terms of decision possibilities offered to decision makers? We can identify a progression of autonomic capability:

1. **Delegation:** this is the usual case today, something needs doing but no help given on pos-

sible actions and no preparations made for implementing those actions.

2. **Interactive adaptation:** the anticipated problem is presented together with set of control points where user can take action. This is an interactive problem solving process involving the user and the system.

3. **Anticipation and interactive adaptation:** problem, control points and costs, potential outcomes and benefits presented to the administrator.

4. **Anticipation and automatic adaptation:** problem, control points and costs, potential outcomes and benefits computed, price or service adaptation made. No human intervention is required.

5.4.2 Techniques

How much is it worth to prepare? This is a robust optimization problem: different decisions should be implementable with little cost. This is also a Real Options [DP94] problem: what is the cost of buying options that cover the appropriate decision scope — and is it necessary?

Decision makers must define policies which describe how risky they want their decisions to be in terms of the accuracy and timeliness of the information presented to them for the decisions. Information has a price, so does accuracy and certainty. In conventional dynamic and stochastic environments mechanisms have been devised to artificially construct accuracy and certainty in information about the future. The two most common methods are forward (futures) contracts and options contracts.

A forward contract gives the rights to a service starting at a defined time in the future (and for a defined period) for a sum which is also predefined. Thus it provides certainty about price and availability once bought. Will the actual price of the service turn out to be the price of the forward contract? Almost certainly not, in general it is not even the expected value of the future price because it includes various utility premia as well as a convenience premium which may be positive or negative at different times depending on market conditions and expectations [Hul00]. An option (on a forward contract) provides the same artificial certainty about future price and availability to the decision maker but without the commitment of actually having to buy. There is, of course, a price for this but it is usually much less than the price of the forward contract itself.

It is conceivable that autonomic systems will trade such contracts with little or no help on the part of the user/decision-maker. This activity will greatly assist in controlling uncertainty about future events. An autonomic system can also construct its own estimates of price and availability as well as estimates of the risk of not reserving in advance.

6 Examples

We will demonstrate the benefits of including economic considerations in the decision process of autonomic systems through two examples, on failure recovery and on the management of differentiated service levels.

6.1 Failure Recovery

This simple example demonstrates how decision making based on economic criteria provides a sound and flexible basis for failure recovery actions. Suppose that D is the (random) duration of a storage component failure. Should the system acquire a backup solution? How can it reach a decision without user input? The answer to the first question depends on the valuation of the data to be stored. A simple proxy of the market value of the data is the sum of the prices of all service contracts which are impacted by the failure. However, typically in cases of failure a compensation is paid back in cash or most commonly in free service time.

Let $c(x)$ be the compensation fee per time unit for interruption of service, where x is the duration the interruption (specified in the SLA). Suppose that from historical data there is an estimate of the probability of failure $P[X = down]$, where X is the operational state of a storage component. Suppose also that failures are Poisson arrivals with rate λ and that failure duration follows a Pareto distribution $f(x; a, k)$ with shape param-

eter $\alpha > 1$ and a scale factor k. The system will choose to acquire at time $t = 0$ a backup service of duration T at the spot price $S(0)$ when the following condition holds:

$$\lambda T \int_{x=k}^{\infty} f(x;\alpha,k)c(x)dx + \phi(\lambda T, \frac{k\alpha}{\alpha-1}) - S(0) \geq 0$$

A fudge factor $\phi(n, \mu)$ is used to capture the (cumulative) "reputation cost" of service outage during period T, where n the number of failures during that period and μ is the first moment of the duration distribution. We want $\phi()$ to be sensitive to one or more moments of the duration distribution so as to be able to specify through $\phi()$ the administrator's risk sensitivity. Generally failures of unusually long duration will have a disproportionately negative effect on reputation. [2]

In the case above we use a spot backup service, but more interesting is the case of option contracts for the anticipation of failure events since these give the system the option to make use of a service only when needed. By purchasing call options on third-party on-demand services a system is able to manage the risk of component failure (but not of data loss) in a better way than with plain redundancy. The reason is that the call is only a right to a service which is only provided upon exercising the option (reduced management costs, better resource allocation). In particular, American calls (exercise at any time point until expiration) are more appropriate here since the date of failure is naturally not possible to determine a priori. In order to make decisions regarding American derivative contracts a system needs to possess stochastic models of price movements for the respective services. Such models are currently under development.

6.2 Service Level Management

In this example we will examine the workings of an autonomic system in charge of selecting and continuously adapting service levels for a large user population, under budget constraints. This example is based on a real case where the charges made for an outsourced service were tied to the sizes of user email accounts. In this actual case costs were managed manually, by forcing users to switch to one or the other service levels manually, one by one as required. This is an ideal case for the introduction of a self-managed system with economic reasoning, and the problem can be formulated as a binary linear optimization problem. Some constraints are economic whilst others are technical or operational.

Assume that there is a discrete number of service levels and for every user k and level i there is a known utility $u(k, i)$ that the user derives from being served at the particular level. The cost for supporting one user at level i is $c(i)$. Charges for the entire service are based on the service levels of all the users combined, so depending on the size of the budget some or all the users may need to be moved to a lower service level or may be able to benefit from higher service levels. Managing this process manually is a painstaking process, especially given the fact that not only budgets, but also pricing schemes of the outsourcing provider, and user utilities, change with time. This service level management problem can be formulated as a simple binary linear optimization problem: select user service levels to maximize the total utility of all users combined, subject to budget and resource availability constraints.

It is also common to allow for the creation of privileged groups whose service levels must be kept high irrespective of the other users. This is included in the formulation below.

We further need to consider the fact that autonomic systems will be in a position to take decisions much faster than humans and thus the rate of change of service levels could be unexpectedly, and undesirably, high. For example, what if the optimal decision is to degrade some users by two levels at once? Such abrupt changes may generate unnecessary complaints on the side of the users who could not foresee such a development. For this and other reasons we include constraints to throttle the system's rate of change.

This leads to the following formulation of the Service Level Management problem [SLM].

[2] Even when we talk about autonomous systems we should remember that when those systems are used to support commercial services the reputation of the service provider will be influenced by some system-level self-adaptation decisions.

Sets
K: set of users
P: subset of privileged users (in K)
I: ordered set of service levels

Data
B: budget
$u(k,i)$: utility of level i for user k
$h(k)$: historical (previous) level for user k
$w(k)$: minimum level for user in privileged group P
$c(i)$: cost of level i
$r(i)$: resource requirements for level i
A: available resources
x: maximum level change

Decision Variables
$l(k,i)$: (binary) level of user k

Objective
$\max \sum_{k \in K} \sum_{i \in I} l(k,i)\ u(k,i)$

Constraints
$\sum_{i \in I} l(k,i) = 1$: a user is in exactly one level
$\sum_{k \in K} \sum_{i \in I} l(k,i)\ c(i) \leq B$: budget
$\sum_{k \in K} \sum_{i \in I} l(k,i)\ r(i) \leq A$: resources
$\sum_{i \in I} 10^{|i|} l(k,i) \geq 10^{w(k)}$: privileged user constraint

$10^{h(k)} \leq \sum_{i \in I} l(k,i)\ 10^{|i|}\ 10^{x}$: level change
$\sum_{i \in I} l(k,i)\ 10^{|i|} \leq 10^{h(k)}\ 10^{x}$: level change

The last three lines of constraints exploit the nature of binary variables and the fact that we have used an *ordered* set of service levels so we can take the ordinal value of a member of that set ($|\ |$).

This is a basic formulation of the [SLM] problem. Utility for a given service level for a given user is an input to the problem. This utility can be the output of a forecasting system that monitors user service usage (email activity in the example mentioned). This formulation can be extended to take account of multiple planning periods and (statistically characterized) errors in utility forecasting.

This example illustrates how a system can continuously adapt and optimize its configuration in response to changes in its self (user utility levels) and environment (price levels and budget).

7 Conclusion

In this paper we have defined and characterized Autonomic Economics, that is, the inclusion of economics within the decision making processes of autonomic systems. We gave two examples of how autonomic systems equipped with economic reasoning will be able to respond to an environment where both technological and business conditions change.

Identity and context give rise to value. Optimization identifies the way to maximize this value under the dynamic and unpredictable circumstances autonomic systems will face in commercial reality. The challenge for autonomic computer system designers is to adequately characterize these circumstances and enable a sufficient range of actions for the system to take without exposing this complexity to human administrators.

Figures 3 and 4 provide some intuition of the links and correspondences between autonomic systems characteristics and economic concepts and methods. Summarizing, we list the tangible benefits of including economic criteria in autonomic systems:

- Economics provides a method to choose among technically feasible alternative strategies, given that adaptation and optimization will not be at any cost.
- Borrowing and lending of resources between autonomic systems will be greatly facilitated by a common standard of comparison: equivalent monetary value.
- Self-management activity is tied straightforwardly to budgets and business objectives: the costs of supporting business strategies are made explicit for varying and unpredictable conditions.
- System analysts can easily audit autonomic actions in terms of cost, revenue and risk.

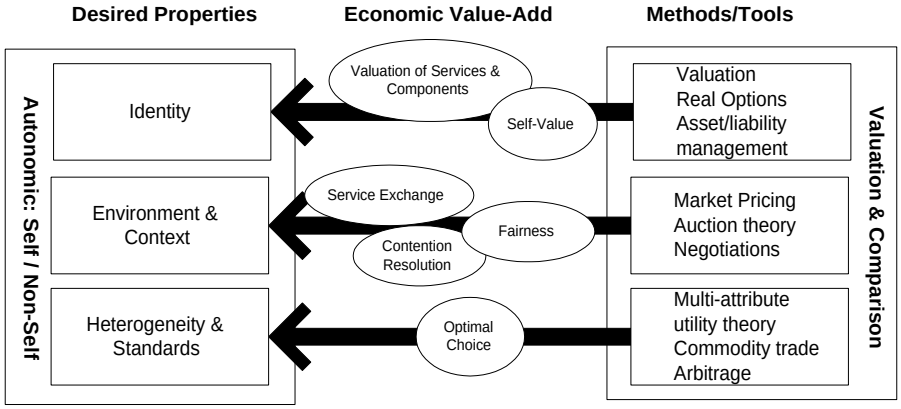

Figure 3: Overview of Autonomic Economics (Part 1) showing connections between Autonomic characteristics and economic concepts and methods.

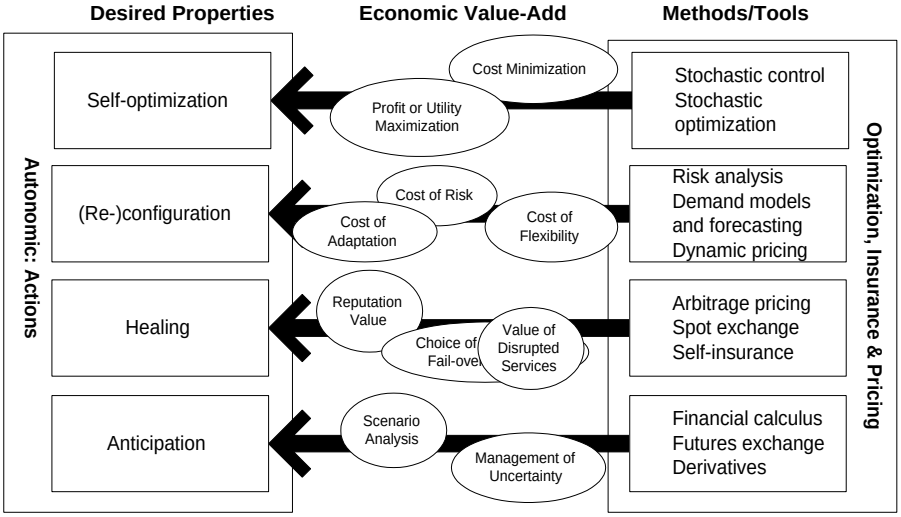

Figure 4: Overview of Autonomic Economics, Part 2.

References

[BGA01] R. Buyya, J. Giddy, and D. Abramson. A Case for Economy Grid Architecture for Service-Oriented Grid Computing. 10th IEEE International Heterogeneous Computing Workshop, April 2001. San Francisco, California, USA.

[BL97] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, 1997.

[BT97] D. Bertsimas and N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA, 1997.

[CDS00] C. Courcoubetis, M. Dramatinos, and G. Stamoulis. An Auction Mechanism for Bandwidth Allocation over Paths. Available at http://www.m3i.org, 2000.

[CK01] G. Cheliotis and C. Kenyon. Failure Contagion and QoS Elasticity in Bandwidth Markets. In *Proceedings of the 10th IEEE International Conference on Computer Communications and Networks*, pages 495–499. IEEE, 2001.

[CK03] G. Cheliotis and C. Kenyon. Grid Economics: 10 Lessons from Finance. available at http://www.zurich.ibm.com/grideconomics. Submitted to IEEE Internet Computing, 2003.

[CKM00] T. Copeland, T. Koller, and J. Murrin. *Valuation: Measuring and Managing the Value of Companies*. McKinsey Inc., John Wiley & Sons, New York, 3rd edition, 2000.

[Dam01] A. Damodaran. *The Dark Side of Valuation*. Financial Times Prentice Hall, London, 2001.

[DGBS00] G. Dermler, M. Güenter, T. Braun, and B. Stiller. Towards a scalable system for per-flow charging in the internet. In *Proceedings of Appllied Telecommiunication Symposium, Washington D.C., U.S.A., April 17-19, 2000*, pages 155–168, 2000.

[DP94] A. Dixit and R. Pindyck. *Investment Under Uncertainty*. Princeton University Press, 1994.

[GK99] A. Greenwald and J. Kephart. Shopbots and Pricebots. In *IJCAI '99. Stockholm, July 31 - August 6*, 1999.

[Hul00] J. Hull. *Options, Futures, and Other Derivatives, Fourth Edition*. Prentice Hall, 2000.

[IBM03] IBM. Autonomic Computing: IBM's Perspective on the State of Information Techonology. http://www.ibm.com/research/autonomic, 2003.

[KC01] C. Kenyon and G. Cheliotis. Stochastic Models for Telecom Commodity Prices. *Computer Networks Vol.36, Issue 5-6, Theme Issue on Network Economics*, pages 533–555, 2001.

[KC02] C. Kenyon and G. Cheliotis. Architecture Requirements for Grid Resource Commercialization. In *Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing, Edinburgh, Scotland, July 24-26, 2002*. IEEE, 2002.

[KR76] R. Keeney and H. Raiffa. *Decisions with Multiple Objetives*. John Wiley and Sons, New York NY, 1976.

[KT01] C. Kenyon and S. Tompaidis. Real options in leasing: the effect of idle time. *Oper. Res.*, 49(5):675–689, 2001.

[Lun97] M.W. Lund. *The Value of Flexibility in Offshore Oilfield Development Projects*. PhD thesis, Norwegian University of Science and Technology, Trondheim, 1997.

[Nef00] S. Neftci. *An Introduction to the Mathematics of Financial Derivatives, 2nd Edition*. Academic Press, 2000.

[RN98] O. Regev and N. Nisan. The Popcorn Market Online Markets for Computational Resources. First International Conference On Information and Computation Economies. Charleston SC, 1998.

[Ros00] S.M. Ross. *Introduction to Probability Models*. Academic Press, New York, 7th edition, 2000. Chapter 9. Reliability Theory.

[Sai97] J. Sairamesh. *Economic Paradigms for Information Systems and Networks, Ph.D. Dissertation*. Columbia University, New York, NY, 1997.

[San00] T. Sandholm. Approaches to Winner Determination in Combinatorial Auctions. *Decision Support Systems 28(1-2)*, pages 165–176, 2000.

[SDK+94] M. Stonebraker, R. Devine, M. Kornacker, W. Litwin, A. Pfeffer, A. Sah, and C. Staelin. An Economic Paradigm for Query Processing and Data Migration in Mariposa. Proceedings of 3rd International Conference on Parallel and Distributed Information Systems, Austin, TX, USA, Sept 1994.

[Sem99] N. Semret. *Market Mechanisms for Network Resource Sharing, Ph.D. Dissertation*. Columbia University, New York, NY, 1999.

[SM99] J.E. Smith and K.F. McCardle. Options in the real world: lessons learned in evaluating oil and gas investments. *Oper. Res.*, 47:1–15, 1999.

[Sut68] L. Sutherland. A Futures Market in Computer Time. *Communications of the ACM*, 11(6), June 1968.

[TCF+02] S. Tuecke, K. Cjaikowski, I. Foster, J. Frey, S. Graham, and C. Kesselman. Grid Service Specification. http://www.globus.org, February 2002.

[TKW00] S. Takriti, B. Krasenbrink, and L. S.-Y. Wu. Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem. *Oper. Res.*, 48(2):281–293, 2000.

[Var] H. Varian. Economic Mechanism Design for Computerized Agents. prepared for the USENIX Workshop on Electronic Commerce, July 11-12, 1995, New York, NY.

[WMMR] M. Wellman, J. MacKie-Mason, and D. Reeves. Exploring Bidding Strategies for Market-Based Scheduling. Available at http://ai.eecs.umich.edu/people/wellman/pubs.

[WWW01] P. Wurman, M. Wellman, and W. Walsh. A Parametrization of the Auction Design Space. *Games and Economic Behaviour 35*, pages 304–338, 2001.