Modernizing DB2 for z/OS with Autonomics

The clear trend these days is to automate and enable computerized tasks to streamline and optimize administrative and maintenance tasks. Many database management tasks that today require oversight and hand-holding by DBAs can, over time, be turned over to intelligently automated software to manage.

Unfortunately, automation sometimes gets a bad reputation. It seems like many IT folks believe they can automate everyone else's job but not their own. Let's face it, if you work in IT, your job involves automating somebody else's job at your company. As the experts on technology you'd think those of us in IT would be able to see the fallacy of this belief.

By developing computerized applications to support business processes, we automate just about every job in our organizations. But try to tell a DBA to automate their utilities or to use advanced autonomics to direct their actions and you'd think you just insulted their mother. Technology folks tend to resist automation for fear of losing control or perhaps, losing our job. These fears are understandable, but for many reasons, this position is not really justifiable.

First of all, there is an IT skills shortage and companies want to hire more IT professionals than are available. In the mainframe world this is even more troublesome as professionals with decades of skills and experience retire to be replaced by less-skilled individuals. Furthermore, many IT folks work long hours. But who really wants to work a 12-hour day or manage maintenance tasks during off-hour shifts?

Mullins Consulting, Inc.

15 Coventry Ct, Sugar Land, TX 77479

Phone: (281) 494-6153

www.MullinsConsulting.com

The truth is, most IT tasks and procedures can be streamlined and made more efficient using automation: automated systems management, database administration and tuning, and autonomics, a newer discipline that will be explained in this white paper.

But keep in mind, automation and autonomics are not designed to make IT professional extinct. It is not possible to completely replace DBAs, nor will it be possible any time soon. However, it is important as organizations struggle to cope with a shortage of skilled IT professionals to look for means of optimizing and streamlining tasks without the need for constant administrative handholding. By turning some of the work over to the computer, IT can become more efficient, more effective, and provide a higher ROI to the business.

So read on for a definition and description of autonomics, including what it is, how it works, and the benefits you can glean from implemented autonomics.

Then we'll dig into some of the recent advancements being made by IBM in autonomics for DB2 on z/OS. The primary focus will be on IBM DB2 utility management and how autonomics can be driven with the DB2 Utility Solution Pack for z/OS. We'll also look at how you can more effectively maintain your DB2 databases and environment with IBM's new Management Console for IMS and DB2 for z/OS and how it can be used with the IBM DB2 Utilities Solution Pack.

Autonomics

So what are autonomics? Autonomic computing refers to the self-managing characteristics of distributed computing resources, adapting to unpredictable changes while hiding intrinsic complexity to operators and users. Autonomics implies much more than simple automation.

The goal of autonomics is to make enable computer systems and software capable of managing themselves. It requires an awareness of the environment, knowledge of changing usage patterns and resources, and the ability to adapt to shifting requirements. The primary benefit is to optimize manual tasks and eliminate the growing levels of complexity.

Components of an Autonomic System

At a high-level, there are four aspects of autonomic computing:

1. It is *automatic,* meaning it can make decisions on its own, using accumulated performance and usage metrics and high-level policies developed by administrators.
2. It is *adaptive*, meaning it can automatically adjust to changing conditions.
3. It is *aware,* meaning the system can monitor (or sense) its operational context as well as its current state to determine if it is reaching its specific purpose.
4. It is *self-managing*, meaning it can optimize and administer itself without human interaction being required.

Autonomics Battles Complexity

And let's face it, complexity is a significant driving issue in organizations today. Some forecasts show that the number of computing devices in use will grow at 38% per year. And the average complexity of each device is increasing – our devices and software are doing more and more with each new version and release. That adds complexity!

Complexity is also increasing because the number of systems that interoperate with each other continues to grow. Connections are made between systems that were never originally intended to share data or processing. Legacy systems rapidly are being transformed to work with modern, distributed applications adding layers of complexity behind the scenes, even as the user interface becomes easier. When you look at all of these trends, you have to admit that it is next to impossible to master all of the interconnected technologies at your shop today, let alone into the future as the trends continue and pick up speed.

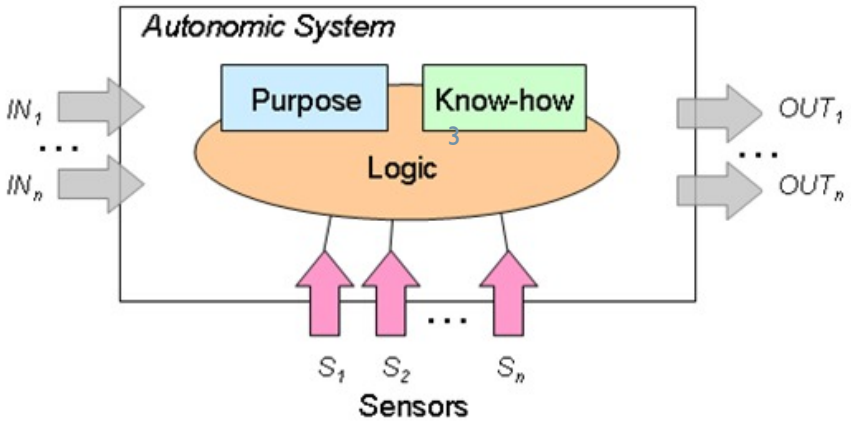

Figure 1. An Autonomic System
(Source: Wikipedia, Sept 2014, http://en.wikipedia.org/wiki/Autonomic_com puting)

The Components of an Autonomic System

So what makes up an autonomic system? At a high-level, an autonomic system is *automatic, adaptive*, *aware,* and *self-managing.* Of course, there are many aspects to "self" managing, and autonomic systems can support varying degrees of self-management capabilities, including:

- Self-configuration: Automatic configuration of the system and its components;
- Self-healing: Automatic discovery, and correction of faults;
- Self-optimization: Automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements;
- Self-protection: Proactive identification and protection from arbitrary attacks.
- Self-inspection: Understands itself and its interactions with other systems in order to make intelligent decisions;
- Self-organization: Proactive modification of data structures and organization to optimize access.

Figure 1 offers a high-level depiction of the components of an autonomic system.

A fundamental building block of an autonomic system is a sensing capability, which enables the system to observe its external operational context. Inherent to an autonomic system is the knowledge of the *Purpose* (intention) and the *Know-how* to operate itself without external intervention. The actual operation of the autonomic system is dictated by the *Logic*, which is responsible for making the right decisions to serve its *Purpose*, and influence by the observation of the operational context (based on the sensor input). The operation of an autonomic system is purpose-driven. This includes its mission (e.g., the service it is supposed to offer), the policies (e.g., defining the basic behavior), and the "survival instinct."

Autonomic Computing Levels

As with most things, it is not reasonable to expect organizations to jump right into a full-blown implementation of autonomics. There are 5 levels of autonomic computing that can be attained, as outlined in Figure 2.

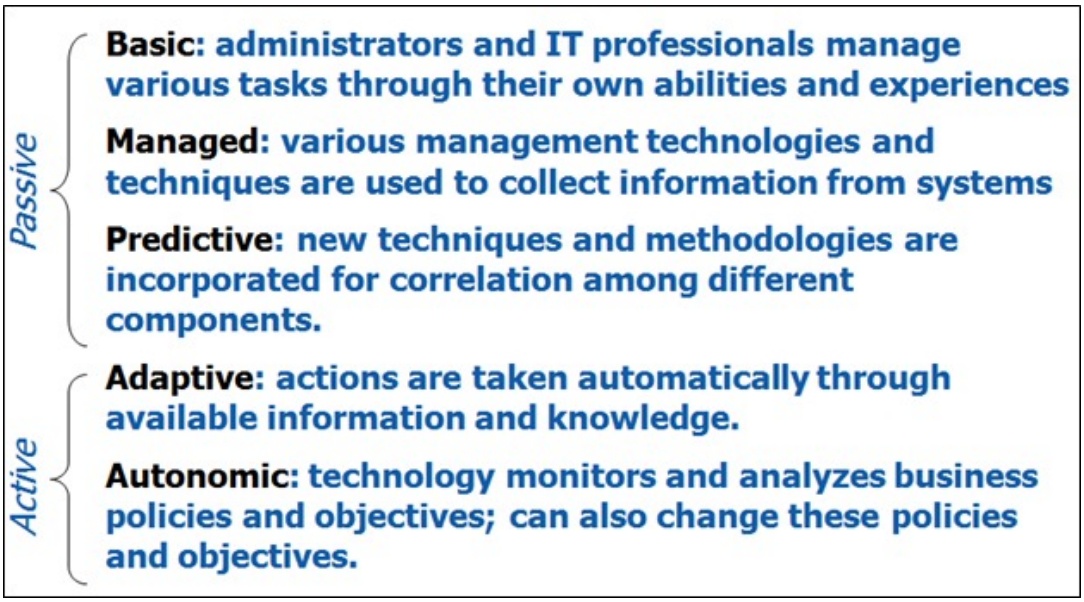

Figure 2. Autonomic Computing Levels

The first three levels are passive, meaning that corrective actions may be advised, but are not automatically taken. The Basic level is essentially through human effort, IT professionals are used to manage the appropriate tasks through their own abilities and experiences. The Managed level augments human effort with management technologies and techniques. And the third level is where we really start to see a more autonomics-focused approach. This Predictive level introduces new techniques and methodologies for correlation among different components.

Then we move into active autonomics, where corrective actions are taken automatically. The Adaptive level uses the information at hand to automatically take actions as needed. And the final level, the Autonomic level, actively monitors and analyzes business policies and objectives – and can even change these policies and objectives based on new observations. Basically, it all boils down to identification of issues vs. not just identifying the issues but also taking action to remediate the issues.

Trust, But Verify

An evaluation of the accuracy of the autonomic capabilities by trusted advisors is an important part of the continuum of moving to autonomics. It never fails that when a technician is first exposed to autonomics, skepticism is justly aroused.

For example, a DBA who is told that some piece of software is going to make automatic adjustments to parameters, jobs or data will usually say something like "Wait-a-minute! First tell me what the problem is, along with what you suggest I should do."

The DBA wants to look at the recommendations and develop a level of trust in the technology. Well, this is the first phase for autonomic software. You can break problem solving down into three high-level steps:

1. Find the problem
2. Analyze the situation for a solution
3. Implement the solution

Automating the first two steps are easily accepted. Every DBA will readily accept help finding problems, analyzing details and even recommending a solution. But the last step takes time.

DBAs will begin to trust the autonomic solution if they aregiven the opportunity to review the proposed actions, and those actions are deemed to be appropriate and accurate. At that point, autonomics can be turned on as fully autonomic… not just finding the problem, and performing analysis to find a solution, but also automatically fixing the problem.

Autonomics and IBM DB2 for z/OS Utilities

Now that we understand what is meant by autonomics, let's take a look at the new autonomics solutions that IBM is introducing for managing IBM DB2 utilities as delivered in the IBM DB2 Utilities Solution Pack.

The first new technology comes in the form of The IBM Management Console for IMS and DB2. This no charge, web-based platform offers a consolidated view of your mainframe database systems. It simplifies your view by making information from a variety of different tools available within a single interface. See Figure 3.

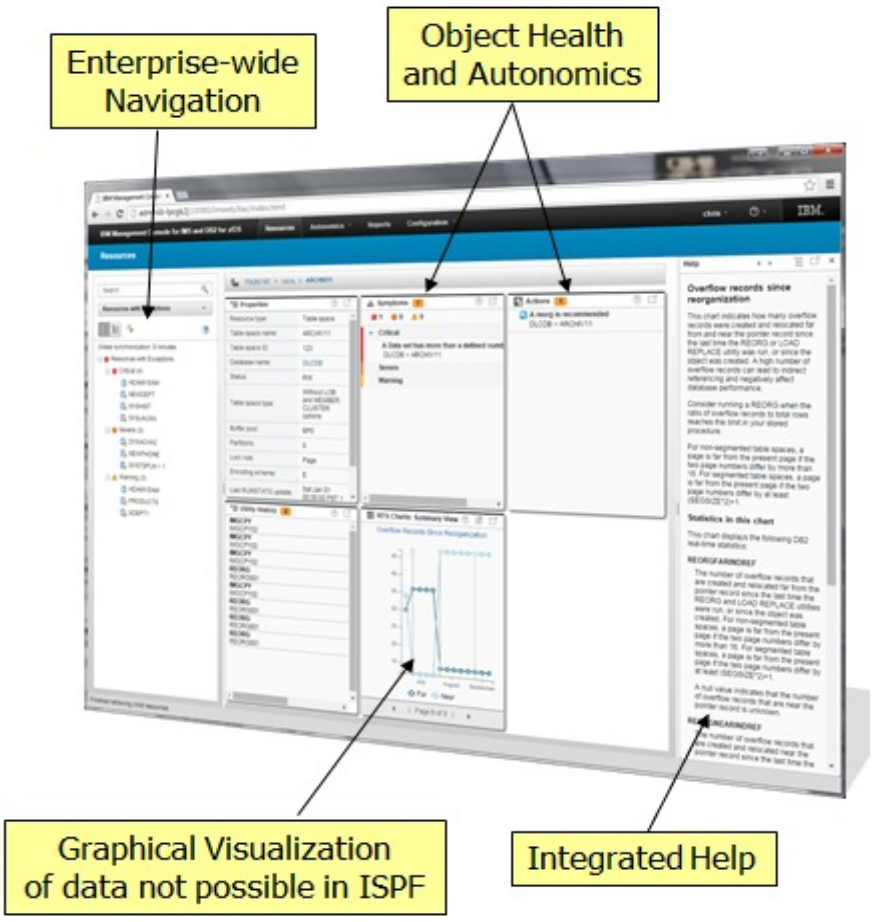

Figure 3.
The IBM Management Console for IMS and DB2 for z/OS

The web-based console offers:

- Rich user interface features to visualize data, quickly show trends, graphing capabilities that are not really possible using ISPF, and so on…
- A framework to integrate information from multiple tools seamlessly and…
- Integrated help to guide less experienced users through data management tasks.

A Powerful Management Console

The high-level goal of the IBM Management Console for IMS and DB2 for z/OS is to provide a single holistic, easy-to-use interface to manage IMS and DB2 systems and databases. You can perform tasks ranging from configuration, automation, performance management and administration – all from a web-based console.

Of course, this is the goal and delivery for DB2 for z/OS tools will be in phases. The first phase is the integration of the Management Console with IBM's Utilities Solution Pack with the eventual long-term goal of integration to support IBM's other IMS and DB2 Tools. IMS users have been using the Management Console, formerly known as the Administration Console, with several IMS tools solution packs. The Management Console supports both DB2 and IMS environments.

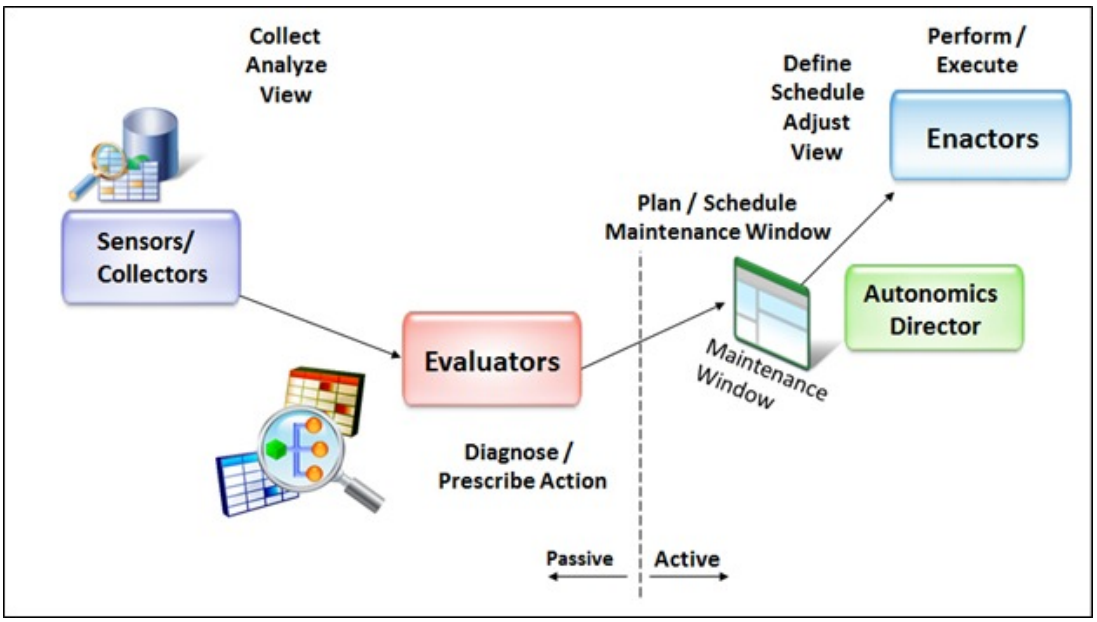

Figure 4. Autonomic Capabilities of the IBM Management Console

Underlying the Management Console, which is the user interface seen by the DBAs and other users, are powerful autonomic components (refer to Figure 4). Think back to the discussion of the components of autonomics outlined earlier in the white paper. Here we see these autonomics components – sensors, evaluators, and enactors – built into the DB2 management solution controlled using the management console. It is important to note that there are passive capabilities, as shown on the left hand side of Figure 4, and active capabilities as depicted on the right hand side. So it is possible to set things up to collect and evaluate, and then enact solutions actively as you become comfortable with the technology.

IBM Autonomics Director for DB2 for z/OS

The autonomics capabilities that are controlled using the web console are implemented via the Autonomics Director for DB2 for z/OS. The goal of the Autonomics Director is for it to enable IBM's DB2 administration tools and utilities to work together and leverage each other's functionality. The Autonomics Director can be set up to automate the routine collection of data, to analyze the collected data, and then to automate decisions and executions based off of the analysis. This opens up a wide range of administrative functionality that can potentially be controlled. But, of course, not all of the tools and utilities will be integrated right out of the gate.

The DB2 Autonomics Director lets you schedule utilities to run autonomically in a maintenance window. So what is coming first? The answer to that question is autonomic management of IBM DB2 Utilities like REORG and RUNSTATS. The DB2 Autonomics Director supports either a passive or active approach for your DB2 utility execution. For example, in a passive autonomics environment, you can automate the gathering of DB2 real-time statistics and information about the results of your DB2 utilities execution. That data is used as input to identify potential problem situations and to recommend actions that you can take to correct the any problem situations. Based on your own analysis, you decide whether to follow the recommendations provided. In an active autonomics environment, utilities are run for you automatically based on a number of factors, including the priority of the job, how many tasks are already running, and how much time remains in your maintenance window. You can also specify the application objects that are most critical to you that should run first when a maintenance window opens.

The Bottom Line

The combination of autonomics and a modern tools console offers intelligent, central management of your database systems. Using a single web interface you can administer both DB2 for z/OS and IMS. With advanced analytics, you can automate basic administration tasks to give DBAs freedom to work on higher business value tasks.

IBM's unified strategy for automating and optimizing DB2 for z/OS and IMS capabilities can simplify and optimize the management of your mainframe database environment. And that is something that overworked DBA groups are likely to embrace.

Mullins Consulting, Inc.

15 Coventry Ct, Sugar Land, TX 77479,

Phone: (281) 494-6153

www.MullinsConsulting.com

About Mullins Consulting, Inc.

Mullins Consulting, Inc. Is a consulting and research firm that offers data management services and solutions that meet the demanding requirements of all size organizations. Let us help you with database and data warehouse design, performance audits and reviews, application performance tuning, SQL optimization, security and governance audits, and DB2 version migrations. Mullins Consulting provides a wide variety of IT services and help clients reduce IT costs through performance tuning engagements.

Craig S. Mullins is president & principal consultant of Mullins Consulting, Inc. He is also the publisher and editor of The Database Site (thedatabasesite.com)

Craig is the author of two best-selling books on database management, including DB2 Developer's Guide, the best-selling book on DB2 for z/OS (IBM Press). He has written numerous technical articles for many IT publications, including popular journals such as Database Trends & Applications, The Data Administration Newsletter, Enterprise Tech Journal, DM Review, and many others.

Craig is a frequent speaker at IT conferences, having spoken about DB2 and database issues to thousands of folks at conferences such as IDUG, SHARE, IBM Insight, IBM Information on Demand, and DAMA Symposium. He has spoken at events in North America, Europe, Asia, and Australia.