Policy-based IT Automation: The Role of Human Judgment

Eser Kandogan John Bailey Paul P. Maglio Eben Haber

IBM Almaden Research Center

650 Harry Rd, San Jose CA

408-927-1949

{eser,baileyj,pmaglio,ehaber}@us.ibm.com

ABSTRACT

Policy-based automation is emerging as a viable approach to IT systems management, codifying high-level business goals into executable specifications for governing IT operations. Little is known, however, about how policies are actually made, used, and maintained in practice. Here, we report studies of policy use in IT service delivery. We found that although policies often make explicit statements, much is deliberately left implicit, with correct interpretation and execution depending critically on human judgment.

Categories and Subject Descriptors

H1.2 [**User/Machine Systems**]

General Terms

Human Factors.

Keywords

Policy, Systems Management, Ethnography.

1. INTRODUCTION

Information Technology (IT) automation has long been used to ensure continuous monitoring and consistent execution of system administration tasks. In policy-based automation, policies are used directly in the management of IT systems in the form of computer-interpretable and executable specifications of guidelines and rules. Policies are expected to incorporate business considerations as well as technical considerations, aiming to connect business and system management more directly [3].

Policies are a promising foundation for automation of IT systems for several reasons. A complete, detailed description of all parameters of an IT system would be large, complicated, and inflexible – and even harder to understand and troubleshoot than the system itself. Policies are defined at a higher level, and so can express business and technical goals for a system in a way that is clear to all parties. But policy-based automation faces many challenges. Typically, control knowledge is built into the system, whereas in policy-based approaches, control knowledge is provided by users and may change on-the-fly given changing circumstances. Though there is extensive literature on automation, previously studied domains (e.g. power plant, air traffic control, etc.) are significantly different from IT systems (e.g., [4]). And little is known about how people understand, develop, articulate, communicate, implement, and use policies in IT management, let alone how people would interact with computers through policies [1].

We conducted a series of field studies in IT service delivery organizations. Our method included techniques such as naturalistic observation, contextual interviews, and artifact collection. Over the course of 16 field visits, observing and interviewing more than 30 system administrators and others in large corporate, university, and government data centers, our studies examined collaboration, tools, and practices among security, database, web, storage, and operating system administrators and data center operators [2].

In what follows, we first describe and analyze an episode from our data, and then discuss what we learned about policy creation and use.

2. STORAGE STUDY

We observed Ryan, a storage design architect who worked in a group that provided managed storage services for a large number of customers. His responsibility was to design the allocation of storage space in a cluster of enterprise storage systems that were shared among different customers. His policy was to allocate space so that each customer experienced the kind of reliability, efficiency, and extensibility that they would if the whole environment belonged to them alone.

Ryan knew a lot about the architecture, current allocations and utilizations, and typical workloads to meet his customers' needs effectively. When architecting storage, he frequently interacted with the customer to plan the best allocation. He explained to us that this "[…] is a manual process. You sit down and talk to the customer, try to get some numbers from them." Ryan's main collaborators worked at the storage operations center, where the actual allocation was done. He was in close contact with them to ensure that his design stayed in sync with the actual allocation of storage. Because design and allocation were done by different teams, discussions went back and forth, as teams tried to reconcile design and the real world.

2.1 "We need to have all our ducks in a row"

We watched as Ryan got a request from a customer to allocate 4 TB of new storage. This request came in the form of a spreadsheet that specified the total amount of storage requested, intended use (e.g. database server), kinds of services requested (e.g. backup, flashcopy), along with technical information such as host name,

CHIMIT'08, November 14-15, 2008, San Diego, CA, U.S.A.

WWPN (world-wide port number), preferred logical unit sizes, and the like.

Given this customer spreadsheet, Ryan examined the whole architecture to find the best spot to allocate storage. Because of the scale of their operations, he relied on several specific artifacts to do this job. One was a specifications database that was automatically populated by the storage devices in the network. Another was a manually maintained storage spreadsheet that showed the whole storage architecture. Ryan explained to us that quite often, "[…] everything comes down to this [storage] spreadsheet." In the case we observed, this was true as well.

Ryan went over the request to reconcile it with the database. At first, he could not find the specified host in the database. Thinking that the problem might be that a new storage device that was not yet reporting to the database, he queried the database using the WWPN information in the request. He found a database entry with the host name QB02QAS061, which did not match the specified host name, QB02QDS061. He decided to use the customer-specified hostname, explaining that his database,

> "[…] might be picking up as a zone name as opposed to the actual host name, so... WWPN numbers match what they gave us here from the originator, the requestor. So, I am comfortable with what's on this sheet and what we are putting together. Besides we'll go out there and look on the SHARKs (storage device) if we need to."

Verification was a vital part of his job,

> "[…] need to make sure that we have all our ducks in a row before we send it down (to operations center) there... So, we don't do storage to the wrong server and have to undo it."

Once Ryan decided on the allocation design, he prepared another spreadsheet for the operations center containing more technical information, such as adapters, clusters, disk groups, fiber channel ports, etc.

2.2 Analysis

Ryan spent quite a bit of time coordinating information among people. First, he needed to interact with the customers to understand their needs. He explained that this was a manual and inefficient process, as he often must talk with the customer to get additional numbers to fill in details of their requests. After familiarizing himself with the request, he tried to match it against the shared environment. Once the request was put into some standard form (in this case, a spreadsheet), he then transformed this request into a format that was useful for the operations center, where allocation was actually performed.

To verify the request, he compared it to what he knew of the environment using the various tools and documents at his disposal In this case, there were ambiguities and errors in the information he had, despite the fact that some tools were automated. We saw Ryan make the judgment call on whether to use the customer supplied host name or a name in the automatically populated database. Interestingly, he chose to believe the information in the customer's request rather than the information in the automated tool. He knew that the automated tool was unreliable in certain ways, i.e. picking zone name instead of host name, and that if the customer knew the correct WWPN, which he had verified, then the host name was probably correct too.

3. THE ROLE OF HUMAN JUDGMENT

As part of the formal provisioning process, Ryan applied his own judgment to find the available storage that best matched the customer's needs. His high-level policy was to ensure that each customer achieved performance and reliability as if the whole environment belongs to that customer alone. Exactly how much performance was sufficient to satisfy that goal was unspecified and left to the administrator to decide. Administrators use knowledge of their environment to determine the appropriate levels of performance and reliability, and to determine the corresponding tradeoffs in their actions. If explicit details and formal processes were the only aspects of policies, then policy-based automation might be straightforward. But as we saw here, policies are often defined so as to rely explicitly upon human judgment.

Policies are often insufficiently specific by design, because they rely on implicit or contextual information, requiring human intervention and judgment to determine a precise outcome (see also [5]). Policies may also conflict or need to be overridden, and in these cases, human judgment is necessary for achieving the desired system behavior. Thus, appropriate interaction methods must be designed to incorporate human knowledge and judgment in the development and use of policies. Policy-based automation systems must be able to capture those areas where human input or judgment is required, and to smoothly transition between human and automation roles.

4. CONCLUSION

Creating effective policy-based IT service automation is challenging. It requires an understanding of not only technological systems, but of business and organizational systems as well. In particular, we have illustrated how the complex, dynamic nature of IT services and business often requires policies to be written and implemented to rely on human judgment. We think that policy-based systems management will achieve the greatest improvements in efficiency and quality when designed to leverage human judgment and input.

5. REFERENCES

[1] Barrett, R., 2004. People and Policies: Transforming the Human-Computer Partnership, in: *IEEE 5th International Workshop on Policies for Distributed Systems and Networks*, 111-116.

[2] Barrett, R., Kandogan, E., Maglio, P. P., Haber, E., Takayama, L., & Prabaker, M., 2004. Field Studies of Computer System Administrators: Analysis of System Management Tools and Practices, in: *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, (Chicago, Illinois, USA, November 06 - 10, 2004). CSCW '04. ACM, New York, NY 388-395.

[3] Kephart, J. O., & Chess, D. M., 2003. The Vision of Autonomic Computing, *Computer*, January 2003, 41-51.

[4] Sheridan, T. B. 2002. *Humans and Automation: System Design and Research Issues*. New York: Wiley-Interscience.

[5] Suchman, L., 1983. Office procedure as practical action: Models of work and system design. *ACM Transactions on Office Information Systems*, 1(4), 320-328.Series. ACM Press, New York, NY, 19-33.