

Policy-Parametric Software

Travis Breaux

Department of Computer Science

North Carolina State University

tdbreaux@ncsu.edu

Abstract

Policy-parametric software describes a vision for enterprise software that promotes organizational policy as a computational parameter in future software systems. In the following position paper, I present a general model of the contemporary organization as the foundation for this perspective and later I will propose several requirements for policy-parametric software to facilitate the transition from conventional human resources to partially automated software solutions. The requirements clearly define the role of human and software agents in matriculating and enforcing organizational policy. The centerpiece of this proposal is a rich knowledge representation formalism that will serve as an interface between humans and machines to guarantee that machines are kept in compliance with organizational policy in real-time.

1. Introduction

Over time, the organization has become the basic framework wherein we define, manage, and propagate procedural knowledge to approximate change, stabilize our environment and therefore advance our civilization. Procedural knowledge in the form of natural language policies are intended to govern the performance of members in an organization across a variety of diverse, reactive contexts. We rely on policies to be predictive of future events so that members act correctly in order to fulfill organizational and individual goals and further guarantee a mutual benefit between provider and consumer of products and services. As an organization matures, policies are invoked, revoked and extended to strengthen the effectiveness and sustainability of the maturing organization. The following discussion will first introduce policy scope and the processes of policy specification and enforcement.

In the most general terms, there are internal and external policies that govern the activities of maturing

organizations. External policies are exposed to the public and govern the transactions between providers and consumers. In fact, external policies can be used by consumers to evaluate the ability and effectiveness of the organization to satisfy their individual goals. Alternatively, behind external policies are internal policies that govern the internal operations of the organization and may include proprietary knowledge. Within internal policies, there may be several sensitive, separate domains of policy-driven operations relating to intellectual property, trade-secrets or legally-protected information such as personal, financial and medical records.

Policy specification is performed by a diverse group of specialists including accountants, managers, business process analysts and lawyers. Each specialist contributes a unique perspective that helps to define the context and scope of each policy including specific rights, obligations, and sanctions over policy-governed activities. Specialists also ensure that policies are compliant with relevant legislation and defensible in a court of law. In this regard, policies have traditionally been codified in natural language with attention to the legal precedence of specific phrases. Ultimately, inconsistent policy effects will drive policy restatement and refinement to encourage consistent and unambiguous policy language. However, these inconsistent effects can include direct and indirect costs that make these policies undesirable to parties in the near term.

Responsibility for enforcing policies begins with human agents. Human agents interpret policy and then attempt to correctly identify actionable contexts for policy. Although incredibly adaptive, in fact human agents are difficult to evaluate and keep concurrent with complex policy changes in a dynamic environment. Inadequate training and miscommunication both lead human agents to miscalculate their environment, costing organizations and their constituents both valuable time and resources. From the point of view of the public beneficiary,

conflicting policy decisions are frustrating and a deterrent to the perception of mutual benefit.

2. Requirements

Policy-parametric software must be realized in a framework that includes a standard parameterization of software for policy. Policy must be encoded in a consistent, unambiguous knowledge representation formalism (KRF) and distributed to software with the purpose of directing data operations. Coupling policy with data representations in the KRF is necessary to guarantee effective and efficient control over data throughout a composition of distributed systems. Important issues in the following discussion include the agent-oriented, conceptual model that drives policy-parametric software and how the KRF will coordinate activities among human and software agents.

In policy-parametric software, conceptual models of human processes distinguish human and software agents that collaborate to fulfill organizational policy in a dynamic, real-world environment. Using concepts that characterize human agents and their roles in policy enforcement, we must develop a formal definition of *agency* (i.e., the capacity, condition or state of acting) to transfer the responsibilities of organizations and their human agents to software agents. Human agents outperform software when evaluating novel situations while software exhibits a verifiable advantage during repetitive tasks. With the *human-in-the-loop*, we must let software agents characterize the rote, software-tractable tasks of organizational policy while promoting unforeseen situations to the human expert for clarification through formal policy refinement and redistribution. Together, human and software agents learn and adapt quickly to a dynamic social and economic market while matriculating the organizational policy infrastructure.

The organizational policy, encoded in a KRF, becomes fundamental in guiding the autonomous, reactive agents in their roles within the organizational framework. The formalized policy serves as a software agent's internal model of the organizational environment, providing policy-based direction specific to an agent's role within the organization. The formal policy also serves as the interface between human and software agents. In opposition to the subjective acquisition and application of policy by human agents, software agents rely on intelligent sensors to interpret their environment and map their situational awareness into the governing, rule-based policies in order to produce predictably correct actions. Changes within the formalized policy dynamically propagate to direct

changes in software agent actions saving valuable resources by asserting that the operational state of the organization is in compliance with its policies in real-time.

In addition to those responsible for specifying policy, consumers, government agencies and system developers will use the KRF to interact with the policy-parametric systems. Consumers, either individuals or other organizations in a supply-chain, will use the KRF to formalize their policy-goals and determine if their goals are satiable before, during and after a transaction. Consumers may use the KRF to compose their own policies, or use the policies of independent oversight groups and trusted individuals. By relying on trust from a social-context that is established and well-understood, policy-parametric software can avoid adopting a particular trust model based on complex, poorly understood cultural norms. Alternatively, government agencies will use the KRF to test the compliance of providers within relevant legal contexts. If systems fail compliance checks, organizations can preemptively take action before government agencies must enforce necessary sanctions. Finally, system developers will use the KRF to assert that system components satisfy operational requirements or to design and develop their systems using interfaces to external systems. In order to reach the broadest possible audience, the KRF must be accessible by using syntax and semantics that are familiar to an everyday context. Accessibility must either be addressed in the formalism itself or in a separate software interface to the formalism. Regardless, accessibility is a significant obstacle to widespread adoption that must be addressed within the KRF specification.

3. Future Work

Three key, enabling technologies will drive the development of policy-parametric software including highly expressive agent-oriented formalisms, distributed and pervasive computing, and intelligent tools. The first technology is an early challenge to policy-parametric software while the last two technologies will only expand the scope and effectiveness of this framework.

Relevant formalisms must be agent-oriented as opposed to solely event-oriented by using conceptual models that describe agency. Inheriting from event-orientation, the formalism must include temporal relationships to express conditions with exact time and relative time, the latter case through relationships between activities. Through agent-orientation, the formalism must include notation for agent capabilities

including permissions, obligations, delegation and sanctions. The reference system in the formalism must avoid complex conceptual predicates, however, by decomposing concepts into rich semantic relationships inspired by natural language policies. The formalism may be developed using formal logic (FL) or context-free grammars (CFG), however, the interface between human agents and the policy specification process must include the full scope of possible stakeholders. Due to the limited exposure most of these stakeholders currently have with FL and given that CFG can better characterize correspondences to natural language policies, a CFG is a better choice for formalizing policies and improving accessibility. More work is needed to first develop a corresponding theory of agency, derive the requirements of a broad policy formalism and finally design a sufficient KRF for policy specification.

Recent advances in distributed and pervasive computing will empower software agents to be mobile and/ or composed from distributed resources. In order to perform localized, context-specific tasks, software agents will need to gain access to resources on remote systems from a variety of different computing environments. Two solutions to this problem are already highly developed: distributed processing and remote procedure calls (RPC). Through distributed processing, software agents as computer processes can be transferred from one system to another in order to access needed resources in hardware (i.e. processors, disks, networks) and/ or software (databases, software libraries, other agents). Alternatively, software agents can perform RPC using a number of standard application programmer interfaces (API) available in most prominent programming languages.

Limiting the application of software agents in non-traditional venues, however, are the capabilities of intelligent tools to perform the sensor-functions required by software agents. Software agents must be capable of interpreting the environment presented by the real-world, whether this is analyzing structured or

unstructured data. Intelligent tools represent functions traditionally restricted to the unique ability of human agents, including speech recognition, optical character recognition, image analysis, natural language processing, and information fusion, to name just a few. These applications have largely matured within the past decade with improvements and refinements occurring every year. Present-day capabilities in intelligent tools are sufficient to begin early work on software agents in policy-parametric software, however, venues emerging with ubiquitous computing will require new advances in processing unstructured information.

4. Conclusion

Policy-parametric software will allow us to trust systems using unconventional means in conventional ways. While policy-parametric software will benefit from the convergence of many emerging technologies, there are still a number of theoretical and methodological issues that need to be addressed before this paradigm shift can be realized in widespread applications. Computer scientists and engineers should step outside of their traditional boundaries to develop an interdisciplinary approach in support of this vision. Specialists from business management, cognitive and computer science, linguistics, law, and sociology all hold valuable insight into the full scope of this problem. Through collaborative effort, knowledge from each of these domains must be integrated into a unified theory of policy-governed interactions.

5. Acknowledgements

I gratefully acknowledge the influence of Dr. Stephen Fickas in the early conception of this position during my tenure at the University of Oregon in spring of 2003.