A Cognitive Policy Framework for Next-Generation Distributed Federated Systems

Concepts and Research Directions

E. Bertino
Purdue University
West Lafayette, IN, USA

S. Calo, M. Touma, D. Verma
IBM Research
Yorktown Heights, NY, USA

C. Williams
UK DSTL, Porton Down
Wiltshire SP4 0JQ, UK

B. Rivera
Army Research Labs
Adelphi, MD, US

***Abstract*—Next-generation collaborative activities and missions will be carried out by autonomous groups of devices with a large variety of cognitive capabilities. These devices will have to operate in environments characterized by uncertainty, insecurity (both physical and cyber), and instability. In such environments, communications may be fragmented. Proper policy-based management of such autonomous device groups is thus critical. However current policy management systems have many limitations, including lack of flexibility. In this paper, we articulate novel architectural approaches addressing the requirements for the effective management of autonomous groups of devices and discuss the notion of generative policies – a novel paradigm that enhances the flexibility of policy-based approaches to management. In this paper, we also survey types of policy that are essential for managing device groups. Even though many such policy types exist in conventional settings, their use in our context poses novel challenges that we articulate in the paper. We also introduce a research roadmap discussing several research directions towards the development of a cognitive and flexible policy-based approach to the management of autonomous groups of devices for collaborative missions. Finally, as our proposed policy paradigm is data-intensive, we discuss the problem of supplying the data required for policy decisions in environments characterized by mobility, uncertainly, and fragmented communications.**

Keywords— Coalitions; Policy based management; Autonomous systems; Edge computing; Generative policies

I. INTRODUCTION

Most activities we may think of are carried out by coalitions, that is, groups of parties joining forces, pooling resources, and sharing information for a common goal. Technological developments in the areas of information and network science [1] and distributed computing systems have resulted in powerful infrastructures supporting coalition operations in many different settings and application domains.

However, advances in robotics as well as the explosion of low cost mobile phones, wearables, drones, embedded devices and the Internet of Things (IoT) are changing the way coalitions will be operating. One the one side, coalitions will include parties or group of parties that are intelligent autonomous devices (e.g. drones, vehicles, robots), able to gather fine-grained information from the operating environments and to physically act on these environments. Many such devices will have rich cognitive capabilities and be able to autonomously control other devices. Also, in many cases, such groups will be hybrid collectives of humans and intelligent autonomous devices.

On the other hand, coalition operations will increasingly take place in environments with diverse sets of small elements able to compute, communicate, and acquire, store and process information. In other words, they will operate in environments today referred to as edge-centric computing environments [2, 3]. In particular, whereas we can expect that computing resources will not be a major limitation even on small devices, communications will be fragmented and bandwidth limited and of varying capacities at different times and physical locations. Devices, especially mobile devices, will move into physically unprotected spaces and dynamically collect data from many different sources. Also, a device or group of devices may have to interact with "unknown" devices. A nice way to characterize next-generation operations environments for coalitions is by the VUCA (volatility, uncertainty, complexity, and ambiguity) acronym [4].

Addressing such challenges requires a novel organizational paradigm for complex environments. Such a paradigm, that we refer to as the *cognitive collaborative systems (CCS) paradigm,* is characterized by constituent elements that are self-managing, by a coupling of knowledge mechanisms with the computing elements. Note that whereas there is typically a tight coupling of knowledge within a device, knowledge diffusion between devices can be less tight. It is thus beneficial to exploit other entities knowledge but from an agility/resilience/autonomy perspective it is critical that devices not to be highly dependent on such diffusion. In a system organized according to the CCS paradigm, groups of entities can self-organize and self-configure.

In order for coalitions organized according to the CCS paradigm to effectively and efficiently carry out their tasks and missions in VUCA operational environments, proper policy-based management is critical. At a higher level policies can be seen as directives given by a managing entity to one or more managed entities in order to guide their behavior. Policies can be of different natures; for example:

- *Constraint policies* focus on actions executed by the managed entities, where different actions are deemed allowed, not allowed, or obligatory based on a set of

policy rules. Access control policies [5] represent a well-known example of constraint policies.

- *Goal-based policies* aim and achieving a definite goal, e.g. maintain a minimum threshold of utilization or try to finish a task before a specific deadline.
- *Utility-based policies* aim at producing the best consequence according to some value function, such as for example maximizing the usage of certain resources, or accomplishing a task within the shortest time.

Note that constraint policies are the least flexible. Goal-based policies are more flexible but fail when all goals cannot be met. Finally utility-based policies allow some goals not to be met if the overall utility is higher. Note also that all types of policies are often present in many application domains.

Policy models, languages, formalisms, and systems have been widely investigated and applied to many different domains [6, 7, 8, 9]. Policy standards have been developed, most notably in the areas of access control (see the XACML standard for access control policies [10]) and network management [11]. However, as discussed by Verma et al. [12], a major shortcoming of existing policy-based management technologies is the lack of autonomy and flexibility at the level of the managed entities. Most technologies are based on some form of rule-based systems that rely on a centralized infrastructure and on the automated enforcement of directives. Managed entities are given policies and have limited or no ability to revise, modify, or customize the policies. However in dynamic coalition environments blind enforcement of predefined policies may prevent the delivery of a critical piece of information from a coalition partner that may be important for mission effectiveness. Policy infrastructures for coalitions must thus provide for the ability to trade-off mission effectiveness against policy relaxation, and support policy adjustment and negotiation to maximize mission effectiveness while minimizing risk. Therefore an autonomic, adaptive, and decentralized approach to policy-based management is required.

In this paper, we discuss challenges and initial approaches towards the design and development of next-generation policy-based management systems. In the next section, we discuss the nature of distributed computing in future environments and the cognitive capabilities edge devices may have. After that, we introduce a high level view of the architecture of policy-based management systems and discuss how such architecture has to evolve for next generation coalitions. We then discuss the types of policies that we believe are crucial for coalition management. Finally we introduce our research roadmap, and discuss initial approaches and research directions. As our envisioned policy-based approach for coalitions relies on the use of information, such as properties of protected resources and devices as well as contextual information, a critical issue is to provision all information required for sound policy enforcement processes. We thus discuss such problem that we refer to as *information logistics*. We finally outline some conclusions and on-going work.

II. Evolution of Cognitive Systems

Cognitive systems, which can be loosely defined as systems that use techniques from Artificial Intelligence to continuously improve their operations, assist humans in their daily tasks, and improve the effectiveness of their users, are becoming ubiquitous around us. At the current time, such cognitive systems tend to be implemented in a cloud-centric approach, e.g., the voice recognition in current phones would typically collect the samples and send them to a cloud-based service to be analyzed and an appropriate response sent back to the phone. While the phones have significant amount of processing power, the complexity of training the speech recognition engines makes it easier to deploy and support a cloud-centric cognitive solution. The same complexity considerations are applicable to other cognitive applications, leading to a cognitive approach which is heavily cloud-centric.

The cloud-centric approach does have its drawbacks, most notably in its need for good network connectivity between a device and the cloud site. Such an approach would not work in a disconnected mode. Even when connected, some types of networks, such as cellular or satellite, may have a significant costs associated with them. In some cases, the cloud centric approach could add a significant latency to the response. Furthermore, it does not take advantage of the computational power and storage capacity available within the device itself. As the nature of the predominant device shifts from phones to devices such as drones and autonomous vehicles, the capabilities of such devices will continue to increase, with further improvements coming from the natural improvement in technology marked by paradigms such as Moore's law.

In such an environment, devices need no longer depend on the cloud infrastructure for their cognitive capabilities. They would be able to learn from their environments, be aware of other devices in the environment, self-organize with these devices, and manage themselves. Such devices would then form a group of cognitive collaborative systems, as opposed to a centralized cognitive system. In this section, we discuss the possible evolution of cognitive systems, and the split of functions between the devices and the cloud system.

We begin by recognizing that any such cognitive system needs to have three parallel but interconnected planes of operations, similar to the design of communications networks where these three planes take the form of a control plane, a data plane, and a management plane. Since the distributed cognitive system performs functions that are more complex than the task of information forwarding in a network, it may be better to call these three planes as the development plane, the cognitive operation plane, and the management plane. In the development plane, the devices are instructed about the functions and capabilities they ought to support, e.g. they are programmed or given the directions about the task they need to learn and perform. In the cognitive operation plane, the devices analyze the data they encounter to learn about their environment, and improve their operations. In the management plane, the devices determine how to deal with exception situations, and deal with their security, performance, and safety issues. We believe that the capabilities in each plane, currently

heavily cloud-based and centralized, will shift gradually towards a distributed device-centric paradigm.

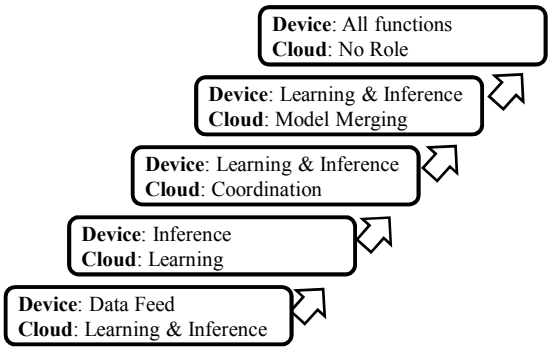

Fig. 1. Evolution of Capabilities in Cognitive Operations Plane

As shown in Figure 1, current cognitive systems are built using the devices as just sources of data. Given the complexity of learning algorithms, the next step in the evolution would be where learning happens in the cloud, but decisions based on the models that are learnt are taken by the device. This is shown as inference on the device. In the next stage, the devices can start learning on their own, extracting patterns from the data that they are seeing. The cloud would have the task of coordinating their learning, e.g. directing different devices to learn about different non-overlapping items. As an example, one device may learn about new flying objects, while another device may learn about objects moving on the ground, and the models are shared using the cloud as an intermediary. In the final stage, the devices can learn models about the same items, and the cloud help in merging the models they learn in this manner. Eventually, the devices can become fully autonomous, learning the models and merging them among themselves without relying on a cloud based system.

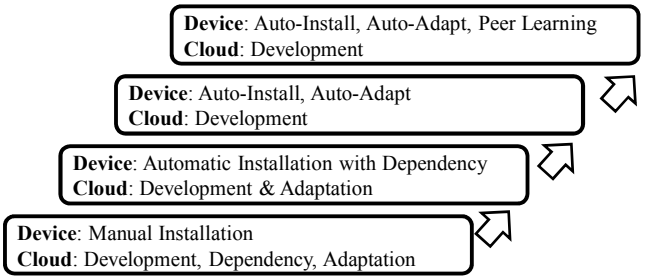

Fig. 2. Evolution of Capabilities in Development Plane

For the development plane, at present time, new capabilities for devices are created at a cloud based system, e.g. in an app-store. The installation of these capabilities on the device is manual. As devices become more intelligent, many of them would be able to dynamically get required components from a cloud based system as needed on their own. They would be further able to adapt and modify a function retrieved from the cloud to work in their environment, e.g. if an application in the cloud supports a different version of the operating system, devices would be automatically adapt it to work on the version available in the devices. In a subsequent phase, devices could learn insights about the components they need by coordinating with other devices, gaining further autonomy. However, due to the ease of development in a central location, the cloud would invariably be the site for developing and distributing capabilities and programs for the devices.

The management plane functions would similarly move gradually from the cloud based management capabilities to be more device-centric. The evolution of management capabilities, especially for policy based management, is described in detail in the subsequent sections of this paper. Note that for development, some of the functions would always have a better affinity to the cloud based system, e.g. maintaining a historic archive of data collected by various devices.

III. GENERATIVE POLICIES

As discussed by Verma et al. [12], in a policy based approach, a key component is the Policy Based Management System (PBMS). The PBMS is controlled by human operators and is connected via a communication network to the managed entities. The managed entities can have different computing and cognitive capabilities; they can range from full-fledged intelligent devices to small devices with no cognitive capabilities. In what follows, we use the term device to generically refer to the managed entities.

The PBMS has sets of policies to manage the behavior of the controlled devices. If events arise or actions have to be executed that are not anticipated by the set of available policies, a human operator has to intervene to define the appropriate policies and/or modify the existing policies.

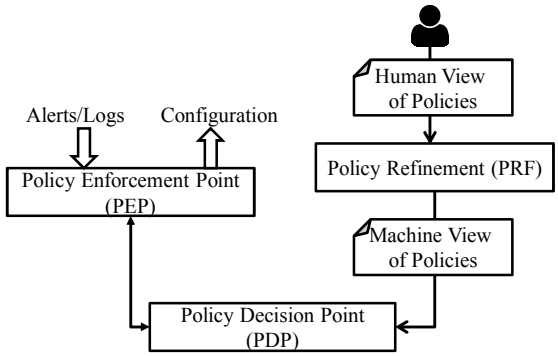

Fig. 3. High Level Reference Architecture for PBMS [12].

A commonly adopted reference architecture for PBMS (see Figure 3 [12]) is based on a policy life cycle under which policies are specified by a human administror, using some high level interface, and are then translated into a machine policy language through a process of refinement and transformation [7]. The machine level view of policies is in turn used by an automated component, referred to as policy decision point (PDP), to take decisions governing the actions of the managed devices. The PDP is typically invoked by the managed devices in order to obtain relevant directives before executing actions. Such invocation is performed via the policy enforcement point (PEP) which basically represents the interface between the managed devices and the PBMS.

In current PBMSs, the managed devices have only the ability to require a decision from the PBMS; they are unable to take their own policy decisions and perform their own policy refinement. However, it is expected that a characteristic of next generation coalitions will be the dynamic nature of coalition formation and composition. Such dynamics will not just impact the geographic extent of the coalition but also the trust between partners, the performance of the infrastructure and the access to capabilities sources from across the coalition. In such a context static policy frameworks would impede the fluid nature of operations, and manual policy management would not have sufficient timeliness to respond to the changing mission, threats or environment.

Therefore an evolution of PBMS is required by which policy management functions are delegated to the managed devices. Figure 4 [12] summarizes such an evolution.

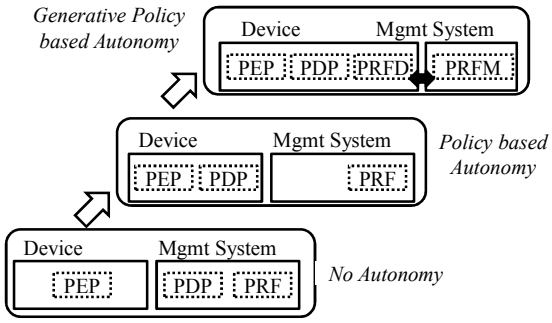

Fig. 4. Evolution of PBMS architectures [11].

Notice that in the architecture referred to as *policy based autonomy* the policy based decisions are taken by the devices, whereas the policy refinement is still executed by the PBMS. Therefore the managed devices have still limited autonomy. By contrast in the *generative policy architecture*, devices are provided an initial policy specification, referred to as *generative policy*. In such an architecture, a refinement component is added to each device, referred to as policy refinement at device (PRFD). Each device can then (dynamically) refine and adapt the generative policy, and based on its own "customized" policy take decisions about its own actions. The generative policy architecture addresses the requirements of autonomous management and flexibility for next generation coalitions.

It is however important to notice that devices may not necessarily interact with a single management party. Rather device management may have to be handed-off from one management party to another, such as in the case of physically moving devices like drones and autonomous vehicles. Or devices may have to be jointly managed by several management parties. As a result, a device may receive multiple generative policies which introduces the possibility of policy conflicts. Devices thus need to be agumented with conflict resolution strategies.

Also device handoff between different management parties needs to be efficient. We notice that in this respect the generative policy architecture may be quite effective, since devices would store policy relevant information and therefore the amount of information to be transferred among managing parties would decrease. Such a *decentralized generative policy architecure* is critical to address the requirement of decentralized management for next generation coalitions. It also addresses the collaborative decision requirement at the level of the management parties as these parties may collaborate on integrating different generative policies and/or decide priorities among generative policies and/or how to de-conflict policies.

To fully meet the collaborative decision requirement, it is also important to allow groups of devices to take autonomous collaborative decisions, as typical of P2P architectures. Such capability is critical in cases in which the devices get disconnected from the management parties. Thus devices are able to share policies, and each device then decides which policies are applicable. It also enhances the efficiency of policy based management as in many cases the devices may have enough information to carry out decisions, thus saving bandwith and other resources. We refer to such an architecture as *decentralized P2P generative policy architecure*. Figure 5 summarizes our discussion by indicating the main components implementing such an architecture.

It is however important to notice that, as a device may in turn manage other devices, a device may in turn delegate some policy management functions to the devices it manages. Therefore our proposed architectural organizations can repeat at different levels in the overall coalition. In addition notice that the various architectures can co-exist in that some devices may still be managed using conventional approaches whereas others may adopt a generative policies approach or a policy based autonomy approach.

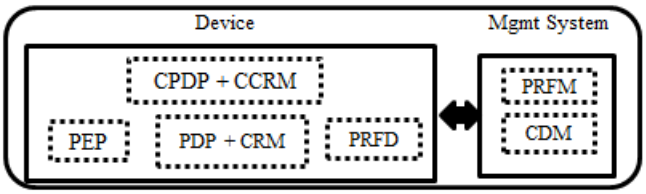

Fig. 5. Decentralized P2P generative policy architecture. Legend: PDP+CRM: PDP enhanced with conflict resolution management (CRM); CPDP+CCRM: Collaborative PDP enhanced with collaborative CRM; CDM: collaborative device management

IV. Policy Domains

The already complex management of coalitions in next generation operating environments must also take into account that many different types of policies, referred to as *policy domains*, are required to cover all critical aspects in coalition management. In what follows we discuss the policy domains that we believe are the most relevant and present in most application areas. In addition to these domains, different application areas may have their own application specific policy domains.

A. Coalition team management policies

This domain deals with the dynamic formation and management of groups, referred to as teams, of devices. An example would be a group of devices that have to form a mobile ad-hoc network to provide an emergency

communication infrastructure in a given geographical area. Policies would be needed to indicate upon which event or situation such a network has to be formed and which devices should be included.

Examples of team management policies have been proposed for group-based negotiations in P2P systems [13]. Two main policies have been identified in such a context:

1. *Membership policies*: They specify the conditions that a peer has to satisfy in order to become a member of the P2P group and assets/resources that the peer would receive upon joining the group. A membership policy is typically expressed as a combination of conditions concerning resources and capabilities that the peer provides and of contextual conditions. In the example of the mobile ad-hoc network, a condition for a device inclusion would be that the device be located in a certain geographic area and have a certain communication range. As such it requires that the peer makes such information available to the group it intends to join. In addition a membership policy may also include resources that the P2P group would make available to a new peer.
2. *Disclosure policies*: They regulate the mutual disclosure of information between a potential new peer and the P2P group. Such a regulated disclosure is critical when, for example, a potential new peer has sensitive capabilities and may be able to disclose such capabilities only if all the peers in the P2P group are from its same organization. Disclosure can be executed according to several steps based on trust negotiation protocols [14].

Even though membership and disclosure policies are key policies for team management, additional policies need to be defined, specifically policies for the team establishment which needs to be also based on occurring or anticipated events, and policies for team disbanding.

B. Security policies

This domain includes all policies that concern the security, both cyber and physical, of the managed devices and of the assets, including information, used by these devices. An observation to make is that, unlike more conventional systems, in our setting a device is both a resource to be protected and an "active agent" against which other devices must be protected. Relevant security policies include:

1. *Access control policies*: They specify the resources, including information resources, to which a device may have access. Notice that in addition to conventional information sources, devices themselves collect and store information that may have to be selectively shared with other devices. For example, a drone may have collected sensitive aerial images that may only be shared with other drones that are cleared to access these images. Therefore each device will likely have its own local access control policy. Generative policy grammars would be critical in simplifying the specification and management of these local policies, given also the very large numbers of devices one may have to manage and changes in trust.
2. *Mobility control policies*: They are relevant for mobile devices as they specify physical constraints about device movements. Such constraints are critical to ensure the physical security of the device and safety. An example related to drones is a policy specifying that certain types of drones can only fly in non-urban areas. Other interesting examples can be found in the context of navigation systems for drones [15]. It is important to notice that physical control policies are very much related to the physical environments which may continuously change. As such the ability to adapt policies locally at the devices is critical.
3. *Function control policies*: As many devices have capabilities to act on the physical environment, e.g. they are actuators, these policies specify which party (e.g. human user, other device, cloud based application, controller) can invoke which function among the functions that a device can execute. An interesting simple example is a policy controlling which mobile phones can open the door of a house by activating a smart lock [16] and for how long such policy is valid. For example, the house guests can only invoke the smart lock during the period of their stay in the house.
4. *Information acquisition control policies*: As many devices have powerful sensing capabilities, these policies regulate which information a device can acquire from the physical environment. For example, a drone may be forbidden from taking aerial images when flying over certain areas. As for the mobility control, policy flexibility is required to adapt to different situations.
5. *Firewall policies*: Members of a coalition would typically need to access common services in support of specific missions. These services can be protected by security mechanisms like firewalls. The members of the coalition that join a particular mission would agree on a set of mission policies. This would involve the specification of high-level network-wide security policies, and the generation of new firewall configurations to bring the system to a policy-compliant state. The policies of the different coalition members would be individually generated to achieve the overall mission security policies. The mission policy would specify whether or not a specific set of users (roles) should have access to a certain set of services. The specific elements associated with the firewalls protecting the different services would deduce low-level configurations which satisfy the high-level policy, or raise an alarm when the network security policy cannot be satisfied. The latter case would lead to the invocation of analytics to determine how the inconsistency can be resolved, and may lead to a modification of the higher level policy.

C. Risk and utility assessment policies

The actions taken by any device puts the device at a certain level of risk. The risk could be the risk of a security breach, the risk of battery running out, the risk of coming under attack by a hostile force, and any other risk determined in the context under which the device is operating. At the

same time, each action also delivers some utility or value, which could be coming closer to the probability of mission success, meeting utilization targets for resources, or having the ability to get some required resources. The balance between the risk and utility that can be provided needs to be taken into account. In the risk and utility assessment policies, the policies specify trade-off between different levels of risks and the utilities that can be obtained.

Risk and utility assessment policies assume that a model exists for evaluating risks and for defining the utility to be derived from an action. The models can be either quantitative or qualitative. In a quantitative model, a mathematical formula or computer simulation can be used to come up with a numeric representation of the risk or utility value. In a qualitative model, a set of questions or a decision tree may be used to come up with an assessment of the risk of utility, mapping it into loose categories. As an example, consider the qualitative risk assessment model used by the United States Homeland Security Advisory System from 2002-2011. Such model places the risk into five levels of risk (blue, green, white, yellow, red). Then a quantitative model for utility assessment computes a numeric value for the utility to be derived from an action. A risk and utility assessment policy may specify the thresholds of utility that must be satisfied in order for the action to be taken at any given level of risk. Risk and utility assessment policies may also determine which models to be used for assessing risk and utility, since different models may need to be used depending on the context.

D. Resource management policies

Within a coalition, there are many competing missions, and the resources available to perform a mission are almost always in limited supply. For any device operating at the tactical edge, resources like bandwidth and battery power are usually constrained, and managing them can benefit significantly from policies that determine how the scarce resources ought to be shared. Within the broad domain of resource management, the following types of policies can be identified:

1. *Asset assignment policies*: During any coalition mission, limited sets of assets such as surveillance equipment, drones, automated mules etc. need to be allocated to the mission. In these cases, policies need to determine which assets ought to be provided preferentially to which missions.
2. *Logistics management policies*: Logistics are a big part of any coordinate large scale operation, such as military ones, and require maintaining the supplies of items needed for coalitions. These supplies include food and rations for personnel on the field, equipment required at forward operating bases, as well as any items needed for social outreach programs, e.g. medicines or blankets to handle for a natural disaster. The management policies for such supplies require maintaining sufficient quantities in store for anticipated future needs, as well as determining how much of a supply to allocate for various requests.
3. *Local device resource management policies*: At each device, there are some resources that can be in limited supply. In current operations, battery power tends to be a resource which is usually critical. In some cases, when hand-held devices are used for high-volume data, such as recording videos, storage may become a scarce resource. Whenever resources are scarce at a device, appropriate policies to manage them need to be put into place.
4. *Topology dependent resource management policies*: Some resources need to be managed in a way that cannot be determined by a device in isolation, but is dependent on the attributes of other devices in the system. An example of such a resource is bandwidth, which is usually scarce in most coalition tactical edge scenarios. Allocating more bandwidth to an application may not impact its performance if the bandwidth at the device is not the bottleneck. Other resources that depend on the topology include computation capacity at a device. Policy based management of such resources requires each device to be aware of the topology, resources available at other devices, and then incorporate them into its own policy determination.

Other types of policies related to resource management can also arise depending on the details of a specific environment.

V. Research Roadmap

The development of a flexible and decentralized PBMS must address a number of requirements, in addition to support a generative policy approach. We discuss such requirements in what follows.

A. Attribute-based policies

A critical function to be executed as party of policy decision processes is to determine which policies are relevant, that is, *applicable*, to certain entities of interest, such as actions to be executed and events to be handled. A common approach is to include in the policies conditions expressed as Boolean combinations of predicates against properties, e.g., *attributes*, of the entities of interest. Attributes are also used to represent properties about contexts, including location, time, mission, task, organization.

An attribute-based approach to policy management has several advantages. It provides a high level and semantically meaningful view of policies [5]. It reduces the policy management complexity in that whenever the properties of an entity or of the context change the PBMS can automatically determine which policies do not any longer apply to the entity or the context. The attribute-based access control (ABAC) paradigm [17] and the XACML standard [10] are based on this notion. We will thus envision our devices (especially high-end cognitive devices) to carry comprehensive description about themselves (e.g., *be self-describing devices*) and also able to acquire information about their context. In addition devices managing other devices may have to provide or supplement the descriptions of the devices they manage, if these devices have limited capabilities (or even no capability) for self-description.

However, the adoption of attribute-based policies in our envisioned operational environments raises several challenges:

- *Attribute naming heterogeneity* – it occurs when different parties involved in a coalition use different vocabularies and concepts. Variations can be classified as syntactic, terminological, and semantic. In particular, semantic variations refer to the use of the same term to refer to different concepts from different domains characterized by different ontologies or to the use of different terms to refer to the same concept or semantically close concepts in different ontologies. Such variations can be identified by using ontology matching techniques [32]. One problem in adopting such an approach in our operational environments is that a device trying to enforce an attribute-based policy against another device or in a given context may not have available a mapping between its attribute ontology and the attribute ontology of the other device. One possible approach is to adopt an indirect ontology mapping strategy. Under such a strategy a device, having to determine whether concept *c* from its ontology maps onto concept *c'* of the ontology of another device (for policy enforcement purposes), may ask other devices whether they have mappings between their own concepts and the concepts of the other device. In this way, the device that has to enforce the policy can build a "path" connecting the two concepts through several matching steps. Of course, once these connections are discovered the device can augment its own knowledge by recording such connections. Such mappings can also be extended with similarity factors as often concepts may not precisely match.
- *Missing attributes* – it occurs when a device does not include in its own description an attribute required by some policy that needs to be enforced, or is unable for security reasons to disclose such an attribute (see Subsection IV.A). To ensure that the policy enforcement process can progress despite missing attributes, several strategies can be adopted. One strategy is to determine whether alternative policies are available that can be applied and do not require the missing attributes. The device having to enforce the policy may then decide whether it is preferable not to execute the action controlled by the initial policy, or to execute the action according to an alternative policy, which may lead however to changes in the actually executed action. One example is the case of a drone that has to transfer data to another drone able to connect to the Internet. The first drone may have the policy that data can only be offloaded to drones from its own organization and that the data have to be encrypted with a lightweight encryption mechanism to save energy. Also once the data have been offloaded, the drone may remove them from its storage. However, if the organization of the latter is not known, a different policy should be applied by which: (i) only data acquired via aerial image acquisition equipment can be transferred (so to hide the use of other sensitive equipment used by the drone); (ii) data should be encrypted with a longer key; and (iii) the offloaded data should be retained by the first drone in order to make sure that data are not lost, in case the second drone may maliciously delete the data. We emphasize that a device may not store all policies covering all anticipated circumstances. In such a case a drone may have to try to discover the policies from other devices. Once such policies are discovered they can be recorded in the device knowledge base.
- *Attributes trustworthiness* – as applicable policies are determined based on values of device attributes, it is obvious that erroneous values for these attributes may lead to the enforcement of incorrect policies. Also as devices may have to operate in hostile environments, attackers may aim at providing deceiving attribute values. Therefore, being able to assess the trustworthiness of such attributes is critical. The problem of data trustworthiness has been widely investigated in many areas, such as for example in sensor networks [18] and web data [19], and different approaches, such as approaches based on data fusion [20, 21], have been proposed. However, deploying such approaches in our setting requires being able to rapidly transmit all required metadata in very short timeframes as devices will often operate at the edge and thus have limited communication available. Specialized approaches are also required for specific types of attributes. A relevant attribute for mobile autonomous devices is the physical location. Attacks have been shown by which attackers can make a device reporting a false location and defense techniques have been proposed [22]. An interesting direction is to take advantage of current micro-location techniques and use location data fusion techniques [20, 21] to be able to validate location information obtained by multiple location technologies. Another complementary direction is to devise P2P location verification techniques.

B. *Dynamic policies*

It is expected that a key characteristic of next generation coalitions will be the dynamic nature of coalition formation and composition. Such dynamics will not just impact the geographic extent of the coalition but also trust between partners, performance of the infrastructure and access to capabilities sources from across the coalition. In such a context static policy frameworks would impede the fluid nature of operations, and manual policy management would not have sufficient timeliness to respond to the changing mission, threats or environment.

Changing mission requirements will lead to reassessment of the utility function (if using utility based policies), and at the device level this is likely to require policy changes to properly implement. While policies for changing threat and environment can be predetermined, the use of such policies assumes all possible threats and environmental conditions can be predicted a priori. Further in edge devices with limited storage and computation capabilities, holding and enforcing large policy sets may not be feasible. Thus agility to changing context is best supported through policy generation, as the point of demand (right time, right place) based on situation awareness of current context and future predictions. However, such a generative policy approach will place significant demands on the ability to identify and resolve policy conflict, as discussed next.

C. Policy conflict resolutions

In a distributed and possibly fragmented coalition environment, where coalition elements are highly autonomous and yet share resources, conflicts among different coalition elements are likely to happen. Conflicts can be of different types and arise for multiple reasons. Some arise because multiple policies recommend inconsistent decisions in regards to the same request, others may arise because of conflicts on the concurrent use of resources, and others can still arise because some circumstances arising at run-time have not been foreseen in the policy specification. For example, if multiple devices share a data repository, and each device has its own access control policies concerning the access to the repository, a conflict may arise if multiple policies are applicable to the same request and some policies allow access, whereas others deny access. In this case, we talk of *conflicts arising because of multiple applicable policies concerning a single request*. A different situation is represented by conflicting requests for the same resource by different parties. An example is of two devices concurrently requiring the use of a drone with specialized infrared equipment for high priority missions with a real-time deadline. In such cases, whereas each single request would be allowed if taken in isolation, it is clear that both requests cannot be satisfied at the same time and need to be serialized, if at all possible. In this case, we talk of *conflicts arising because of multiple conflicting requests and the lack of policies concerning concurrent resource usage.* More complex situations can occur where the different types of conflicts can arise for a given set of shared resources and concurrent requests for resource use. Policy de-conflicting is critical for dynamic policies as before evolving a policy (or set of policies) one must assess whether new conflicts would be introduced. Further, changes in context (mission, threat, and environment) will lead to changing priorities, impacting on policy conflict resolution, and former resolutions may need to be revisited as context changes.

A comprehensive and articulated conflict management process is thus required. It is important that such a process be organized around the following conflict management lifecycle: (1) Conflict prevention – to anticipate conflicts as much as possible, to investigate restrictions on generative policies to prevent conflicts, to reconcile conflicting policies, and to introduce policies for conflict resolution; (2) Dynamic conflict resolution – since not all possible conflicts may be identified in advance and/or it may not always be possible to reconcile possible policies in advance, one must be prepared to dynamically resolve conflicts at "run-time", in some cases resorting to the intervention of humans; (3) Post-conflict assessment and learning – once a conflict is dynamically resolved, it is important to gather data concerning the way the conflict was resolved and data related to the conflicting requests, including request context information. Such data can then be used for future conflict resolution and lead to enhanced conflict prevention. Such activity is particularly crucial when humans are involved in the conflict resolution (step 2) since this would allow the conflict management system "to learn" from humans.

Supporting such a conflict management process requires addressing several research challenges, including:

- *Static policy analysis techniques for conflict identification* – an example is represented by techniques for static analyses of XACML policies based on the use of multi-terminal binary decision trees, in turn based on model checking (see the EXAM system [23]). Such techniques allow policy analysts to statically determine for which access requests two XACML policies conflict and for which they do not. The use of such techniques requires however major extensions to deal with policy languages different from XACML and to support the analysis of conflicts arising because of multiple conflicting requests for concurrent resource usage.
- *Techniques for policy reconciliation* – based on results from the static policy analysis, the policies may need to be revised in order to eliminate conflicts. For example, in the case of access control policies, the analysts may decide that when two policies – say P_1 and P_2 – conflict, P_1 has precedence, or they may decide to restrict or expand the applicability of the policies. In order to support policy reconciliation a possible approach is to define an algebra of policy operations. An example of such an algebra is the one developed for XACML [24]. In addition to the development of an algebra suitable for a wider range of policy languages than XACML, important theoretical questions concerning such an algebra have to be addressed, such as completeness, e.g., whether the algebra allows one to express all possible policy reconciliation strategies, and whether the algebra is minimal. In addition, tools need to be developed supporting the high-level specification of algebraic expressions and the automatic generation of restructured policies, expressed in the policy language of interest.
- *Techniques and strategies supporting dynamic conflict resolution* – there are various strategies for dealing with conflicts at run-time. One simple approach is to put in place a default conflict resolution strategy; for example, in the case of access control policies, a simple default strategy is one by which if one policy grants access and the other denies access, the policy denying access prevails. Another possible strategy is to look for past occurrences of the same conflict and apply the same conflict resolution decision taken in the past. Finally, another strategy is to ask a (group of) human user(s) to solve the conflict. Notice that such strategies are often complementary. For example, the system may first look for past occurrences of the same conflict and automatically apply the same decision. If there are no such occurrences, the system may apply the default conflict resolution policy. If no such default exists, the system should prompt a human for conflict solving. In other cases, even though the system may be able to automatically solve the conflicts, the involvement of humans may still be required to validate the decisions. Strategies are required for dynamic conflict resolution, including possible conflict resolution steps and the optimal sequence of resolution step applications.
- *Human involvement* – a critical issue when involving humans in conflict resolution is which information is to be presented to humans in order to facilitate their decision taking. Various strategies are possible, including providing

humans with: examples of similar decision situations, recommendations with explanations, and examples of decision consequences. In this respect, it is important to notice that groundbreaking work on behavioral economics and prospect theory has shown that when making decisions humans have certain limitations and are subject to certain biases [25]. Therefore it is important that strategies for conflict resolution processes involving humans be informed by such previous work as well as work on bounded rationality [26] in order to determine the best information that should be presented to users.

- *Techniques for assessing decision consequences and learning from past decisions* – we need tools to acquire and log relevant information concerning the consequences of past decisions and to "quantify" such consequences. Analytic techniques need to be developed that allow one to mine data concerning past decisions. Data mining techniques may need to incorporate information about biases and circumstances under which policy decisions were taken, and continuously learn, which requires the use of active learning techniques. Results from such assessment and learning must be used to automatically revise policies and conflict resolution. Learning strategies are required suitable for dynamic, uncertain and adversarial environments [27] and tools need to be developed for automatic policy evolution and versioning based on machine learning results.

D. Policy formal models

Several formalisms have been proposed that are relevant for reasoning about different aspects of policies and as a basis for actual policy languages and policy management tools. Notable examples include event calculus [28], in its many different variations [29], model checking [30] used for determining whether a system complies with specific policies, and multi-terminal binary decision diagrams used for determining the impact of policy changes [31]. Such approaches have been very useful in the specification and analysis of policies in the context of conventional PBMS.

However, formalisms suitable for our intended application environments need to address additional requirements arising from the heterogeneous, distributed, autonomous, dynamic elements that must be accommodated. These will be increasingly prevalent because of the impact of edge computing and IoT. The ability to incorporate intelligence in even small devices and to make use of contextual information from widely deployed sensors has already begun to change management paradigms. The class of policy systems that we envision will need to work across different platforms and at different levels of abstraction. A key concept in the generative approach is that local elements will generate their own operational policies within the bounds of higher level policy structures supporting collaboration and meant to assure compliance with high level constraints and the pursuit of common goals. The formal model of a generative PBMS must thus provide constructs for specifying both the higher level policy structures, the operational policies that are locally generated, and the relationships between the two. In addition, techniques would be required for verification that the formal properties, specified at the level of the generative policies, are indeed being complied with by the operational policies.

One approach being considered is that of utilizing a grammar (a Context Free Grammar, GFC, or an Attribute Grammar) for capturing the set of allowable policies, and having the distributed elements generate and employ operational policies only if they are derivable from that grammar. Another approach would be to specify policies at the higher level in terms of more abstract concepts, and utilize a refinement hierarchy at the local elements to produce operational policies pertinent to the local context.

Another important requirement is to deal with uncertainty. In widely distributed, cooperating systems, no element may have an accurate view of global system properties. Uncertainty may affect, for example, the knowledge of events (in the case of an event calculus) as a device may not even be certain that an event has occurred. Also information used for policy decisions may be uncertain. Therefore, whatever reasoning and analysis techniques one uses, they have to take uncertainty into account.

Suitable formalisms may also have to support similarity-based reasoning by which one can reason on policies by taking into account similarities existing between different events, conditions, or actions. For example, suppose that a policy is specified that requires an action to be executed upon the occurrence of an event *e* and that a different event *e'* occurs for which no policy exists. If *e* and *e'* can be assessed to be "similar", then the policy specified for *e* can be applied to *e*'. Note that due to current advances in the areas of machine learning and data analytics, it is possible to develop effective similarity techniques. However, how to integrate such types of reasoning with different formal approaches requires research.

E. Decentralized and dynamic policy management architectures

In highly dynamic, tactical environments, the supporting infrastructure for the policy system will have intermittent, disrupted and high latency connectivity. Single points of failure, such as for centralized policy servers/repositories, are undesirable. Instead architectures without single points of failure are required, that can manage policy distribution and replication without undue overhead.

Conventional policy systems employ a 'policy push' approach, whereby the creator of policy is responsible for pushing that policy out to devices that require it. Disconnection from the network breaks this flow of policy. Instead alternative policy distribution mechanisms will be required, with a greater 'policy pull' from devices that will provide greater resilience. Resilience is further enhanced when the location of policies is not important, such as by following Content Centric Networking concepts that natively support policy replication and caching. Such an approach would be underpinned by the attribute-based policy approach previously described. In extreme cases of disconnection, where policy starvation would prevent effective device operation, local generative policy mechanisms ensuring minimal policy sets can be maintained, consistent with known goals/utility

functions, until connection back to the wider policy system can be established.

VI. Information Logistics

The discussion in the previous section has clearly shown that suitable PBMS require information about policies, contexts, attributes of devices and resources, past conflicts, past decisions and outcomes of these decisions. Such policy information in many cases may be pre-loaded into devices, or in components implementing policy management services. In many other cases, however, the policy information cannot be pre-loaded; for example the information rapidly changes, it is not known a-priory, or the device has not enough storage capacity to store the information. Therefore, the device must have the ability to search for the required policy information. In other cases, the policy information is very sensitive (since it reveals mission intent) and thus may not be loaded on the device for security. In such cases, the device may have to ask another device or some other party to process the policies. VUCA operational environments add complexity to the problem of information logistics. In our context, we have to also take into account that the devices will be operating on the edge and therefore may have very limited bandwidth available to search and gather required information. Conventional approaches, like information push by mean of publish-subscribe systems, may not work as devices may not know in advance the information they will need or even if they determine which information is needed, by the time this determination is completed, the devices may become isolated and thus unable to subscribe for new information.

A new approach has to be devised which is *anticipatory*, that is, an approach by which information is gathered in advance and policies are generated so to be ready for delivery to the device the moment the device connects again. Such anticipation has to be based on a number of factors, including information about the device and its missions, the information the device already has, the expected movements of the device (for mobile devices), and the potential contexts of operations for the device. For example, for a mobile device one may anticipate potential movements based on characteristics of terrains, roads, and navigation points. The task of carrying out an estimation of the information and policy needs of a device may seem very challenging. However, we note that edge-computing environments are very often supported by back-off infrastructures characterized by powerful data analytics and predictive tools. Therefore one may use such infrastructure for the purpose of anticipating the information needs for policy management and anticipating where and when this information needs to be delivered.

As well as logistical issues relating to policy information, a number of issues relate to mission information to be processed by edge devices in order to fulfil their designated tasks (e.g. sensor data processing). As noted in Section II, the shift towards increasingly cognitive devices will also see a shift away from information storage in the cloud. However, for some tasks historical information becomes relevant to the current task, but it would not be efficient or even possible (given storage limitations) to store all information that *may* be relevant in the future. Thus a degree of cloud storage is still necessary. In order to avoid issues of network loading and disconnection, the ability to pre-emptively move such information between the cloud and devices (or to local caches) ahead of when it is needed will circumvent such limitations. This mirrors the anticipatory positioning of policy as just described, but applied to mission data. In order to control such anticipatory data positioning, we can we use the generative policy paradigm to have each device (or cache) determine on its own what data it should request, store, and store for how long. Such decisions need to reflect the expected utility of the information, storage limitations and 'cost' to transfer the information. Kott et al. [21] well articulate the challenges related to information logistics for operations on battlefields. There is no doubt that similar challenges also occur in domains other than military ones.

VII. Concluding remarks

In this paper we have introduced a research roadmap towards the development of a data-intensive cognitive approach for policy based management of next-generation federated systems of devices, including robots, drones, self-driving vehicles, and IoT devices. We have discussed how existing approaches need to be extended for use in such systems. A key notion is represented by the generative policies by which devices are given specifications of the elements of the policies of interest and they can refine/change these elements according to their own local knowledge and situation. We are currently developing case studies in the area of firewall policies and drone navigation systems to assess the use of generative policies and identify further challenges.

As final remark we would like to emphasize that the development of a comprehensive management system based on generative policies requires techniques from different areas of computer science, including agent technologies, risk-based assessment techniques, reasoning and predictive techniques, service-oriented architectures, edge computing.

Acknowledgment

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] D. Verma, W.Leland, T.Pham, A. Swami, and G.Cirincione, "Advances in Network Sciences via Collaborative Multi-Disciplinary Research", Proceedings of 18th International Conference on Information Fusion, FUSION 2015, Washington, DC, USA, July 6-9, 2015.

[2] P. Garcia Lopez et al., "Edge-Centric Computing: Vision and Challenges", ACM SIGCOMM Computer Communication Review, Vol. 45, No.5, pp. 37-42, October 2015.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges", Internat of Things Journal, Vol. 3, No. 5, pp.637-646, October 2016.

[4] B. Johansen, "Get There Early: Sensing the Future to Compete in the Present", San Francisco, CA: Berrett-Koehler Publishers, Inc. , 2007, ISBN 978-1-57675-440-5.

[5] E. Bertino, G. Ghinita, A. Kanra, "Access Control for Databases: Concepts and Systems", Foundations and Trends in Database, Vol.3, No.1-2, pp.1-148, February 2011.

[6] S. Illner, A. Pohl, H. Krumm, I. Luck, D. Manka, and F. Stewing. "Policy-based Self-management of Industrial Service Systems", Proceedings of 4th IEEE International Conference on Industrial Informatics, Singapore, August 16-18, 2006.

[7] D. Verma, "Simplifying Network Administration Using Policy-Based Management", IEEE Network, Vol.12, No.2, pp.20-26, August 2002.

[8] M. Maullo and S. Calo, "Policy Management: An Architecture and Approach", Proceedings of the 1st IEEE First International Workshop on Systems Management, Los Angeles, CA, USA, April 14-16, 1993.

[9] D. Agrawal, K. Lee and J. Lobo, Policy-based Management of Networked Computing Systems, IBM Research Report TC23685, August 2005.

[10] OASIS eXtensible Access Control Markup Language (XACML). OASIS (oasis-open.org). Retrieved February 2, 2017.

[11] IETF, "Policy Core Information Model (PCIM) Extensions", January 2003, available at https://tools.ietf.org/html/rfc3460.

[12] D. Verma et al., "Generative Policy Model for Autonomic Management", January 2017, submitted for publication.

[13] A. C. Squicciarini, F. Paci, E. Bertino, A. Trombetta, S. Braghin "Group-based Negotiations in P2P Systems", IEEE Transactions on Parallel and Distributed Systems, Vol.21, No. 10, pp.1473-1486, October 2010.

[14] A. J. Lee, M.Winslett, K.J. Perano, "TrusBuilder2: a Reconfigurable Framework for Trust Negotiation", Proceedings of Third IFIP WG 11.11 International Conference on Trust Management (IFIPTM), West Lafayette, IN, USA, June 15-19, 2009.

[15] M. Gharibi, R. Boutaba, and S. L. Waslander, "Internet of Drones", IEEE Access, Vol. 4, pp:1148-1162, March 2016.

[16] G. Ho, D. Leung, P. Mishra, A. Hosseini, D. Song, D. Wagner, "Smart Locks: Lessons for Securing Commodity Internet of Things Devices", Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, ASIACCS'16, Xi'an, China, May 30 - June 3, 2016.

[17] R. Sandhu, "Attribute-Based Access Control Models and Beyod", Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ASIACCS '15, Singapore, April 14-17, 2015.

[18] M. Rezvani, A. Ignjatovic, E. Bertino, S. Jha, "Secure Data Aggreation for Wireless Sensor Networks in Presence of Collusion Attacks, " IEEE Transactions on Dependable and Secure Computing, Vol.12, No.1, pp.98-110, January-February 2014.

[19] X. Li, X.L. Dong, K. Lyons, W. Meng, D. Srivastava, "Truth Finding on the Deept Web: Is the Problem Solved?", CoRR abs/1503.00303, 2015.

[20] C. Bisdikian, L. M. Kaplan, M. B. Srivastava, D. J. Thornley, Dinesh C. Verma, R. I. Young, "Building Principles for a Quality of Information Specification for Sensor Information", Proceedings of 12th International Conference on Information Fusion, FUSION 2009, Seattle, WA, USA, July 6-9, 2009.

[21] A. Kott, A. Swami, B. J. West, "The Internet of Battle Things", IEEE Computer, Vol.12, No.12, pp.70-75, December 2016.

[22] J. Won, E. Bertino, "Inside Attack Filtering for Robust Sensor Localization", Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2016, Xi'an, China, May 30 - June 3, 2016.

[23] D. Lin, P. Rao, E. Bertino, N. Li, J. Lobo, "EXAM: a Comprehensive Environment for the Analysis of Access Control Policies", International Journal of Information Security, 9(4), pp.253-273, April 2010.

[24] P. Rao, D. Lin, E. Bertino, N. Li, J. Lobo, "Fine-grained Integration of Access Control Policies" Computers & Security, 30(2), pp.91-107, February 2011.

[25] A. Tversky, D. Kahneman, "Judgment under Uncertainty: Heuristics and Biases", Science, 185(4157), 1124-1131.

[26] G. Gigerenzer, D. G. Goldstein, D. G., "Reasoning the Fast and Frugal Way: Models of Bounded Rationality", Psychological review, 103(4), 650.

[27] M. Kantarcioglu, B. Xi, C. Clifton "Classifier Evaluation and Attribute Selection Agianst Active Adversaries", Data Mining and Knowledge Discovery, Vol.22, No.1-2, pp.291-335, January 2011.

[28] R.A. Kowalski, M. J. Sergot, "A Logic-based Calculus of Events", New Generation Computing 4(1): 67-95, January1986.

[29] F. Sadri, R. A. Kowalski, "Variants of the Event Calculus", Proceedings of Logic Programming, Proceedings of the Twelfth International Conference on Logic Programming, Tokyo, Japan, June 13-16, 1995.

[30] M. Bartoletti, P. Degano, G. L. Ferrari, R. Zunino, "Model Checking Usage Policies", Mathematical Structures in Computer Science 25(3): 710-763, January 2015.

[31] K. Fisler, S. Krishnamurthi, L. A. Meyerovich, M. C. Tschantz, "Verification and Change-impact Analysis of Access-control Policies", Proceedings of the 27th International Conference on Software Engineering (ICSE 2005), 15-21 May 2005, St. Louis, Missouri, USA.

[32] J.Euzenat and P.Shvaiko, "Ontology matching", Springer-Verlag, Second Edition, 2013.

[32] S. B. Kodeswaran, O. Ratsimor, A. Joshi, F. Perich, "Utilizing Semantic Tags for Policy Based Networking", Proceedings of the Global Communications Conference, 2007. GLOBECOM '07, Washington, DC, USA, 26-30 November 2007.