Community-based Self Generation of Policies and Processes for Assets: Concepts and Research Directions

Elisa Bertino*, Geeth de Mel†, Alessandra Russo‡, Seraphin Calo§, Dinesh Verma§
*Purdue University, USA
bertino@purdue.edu
†IBM Reseacrch, UK.
geeth.demel@uk.ibm.com
‡Imperial College London, UK
a.russo@imperial.ac.uk
§IBM T J Watson Research Center, USA.
{*scalo,dverma*}@*us.ibm.com*

***Abstract*—With the advancement in the technology, deploying connected assets—especially intelligent autonomous assets—to obtain the evolving picture of dynamic environments are fast becoming a reality—and a need—for effective and efficient decision making. In such environments, these assets need to function in unison with each other to achieve the goals, and especially in a collaborative environments (e.g., coalition environments) they need to respect the constraints placed on them by the collective as well as by the owner parties. Typically, policies are used to govern such constraints and interactions, but the existing state-of-the-art relies on predefined user policies to achieve the effect, which is not scalable nor practical in collaborative and dynamic environments. Motivated by this observation and the recent uptake in learning technologies, in this paper, we present our vision on a framework that can *(a)* employ multiple techniques to create domain knowledge that can help assets to determine which policies are critical for which context, how to solve conflicts among policies, and how to autonomously generate and refine existing policies; *(b)* represent knowledge in a localized wiki-like approach so that fault tolerant knowledge discovery is supported; *(c)* provide efficient query interface for assets to discover needed knowledge in a secure manner; and *(d)* contextualize knowledge so as to enable other similar assets to quickly bootstrap or initialize themselves in unknown contexts when new events occur.**

***Index Terms*—Generative policy, Domain model learning, Community-based knowledge, Connected devices, Internet of Things**

I. INTRODUCTION

Distributed operations—be they military or humanitarian—are increasingly expected to take place in volatile, uncertain, complex and ambiguous environments. In such environments, to make informed decisions and to support emerging and evolving tasks, a multitude of assets are deployed. It is envisaged that in future, small teams of operators (i.e., both humans and machines) will be operating at the edge in support of such tasks with a certain degree of autonomy to achieve individual—as well as collective—goals supported by intelligent assets which are inter- and intra- connected; such assets are referred to as *organic assets*.

When such organic assets become active and autonomous in the operating environment, they need to be able to manage and adhere to constraints—be they operational or functional, or individual or coalition—whilst interpreting the context they find themselves in. For example, in a military situation, an organic asset might decide to sacrifice itself when its platoon comes under fire to report the shooter's location rather than returning to its launch platform as it has been instructed to do; or in a smart home situation, a toaster may learn from a smart oven about smoke thresholds acceptable to the household.

Proper management of these organic assets requires an evolving and agile policy-based approach by which they can infer the right policies for managing their systems and the interactions with other assets and humans; the right policies depend on many factors, including context, asset status, current mission and so forth, and may evolve over time. In many cases, organic assets will have to deal with unforeseen circumstances. Therefore, it is critical that organic assets can dynamically select and refine policies, and even create new policies, based on all these factors.

In order to address such requirements, the state-of-the-art proposes the notion of generative policies for autonomous management of organic assets—be it home automation [1] or next-generation coalition operations [2]. In these works, the managed assets are provided with a set of policy structures and a specification of collective goals and constraints. Each asset can then dynamically generate its own policies (hence generative policy [3]) within the bounds of the higher-level constraints meant to support collaboration and the pursuit of individual goals. The generated policies take into account the context of the local system, and can be customized to the local environment. Such an approach has several advantages: *(a)* reduced cognitive burden on humans by automatically detecting conflicts among policies, idle and obsolete policies, and then solving such issues in the appropriate environment [4]; *(b)* timeliness of the responses as assets are much more turned into the operating environment; and *(c)* adaptation capabilities w.r.t. the ever evolving environment.

In support of the generative policy notion, in our recent research, we have implemented an approach based on higher-order ontology constructs for device and service-level policy generation [1]; additionally, a conceptual reference architecture was developed to support in-device policy generation for autonomous management of assets [3]. In such an architecture assets have their own refinement component allowing them to interpret the information they receive in terms of their own requirements and context. However, a limitation in the current state-of-the-art in supporting such an architecture is that assets still need to be given the generative policy information—especially the generative semantics—by some management party.

Motivated by the above challenge and the gaps in the state-of-the-art for generative policies, in this paper, we propose a framework to address these shortcomings inspired by a community-based approach that allows organic assets to (selectively) record context—especially the action and events that resulted in the said policy creation, and share policies and feedback from the use of policies in different situations in dynamic and evolving environments.

In a way, such an approach can be seen as similar to community-based knowledge sites such as wikiHow[1] and Stack Overflow[2]. The main difference, however, is that the users contributing knowledge will primarily be organic assets, such as computer systems, software agents, cognitive devices, and so forth. Knowledge—expressed in the form of policies, policy refinement, feedback from the application of policies, and actions and events which resulted in creating and consuming those polices—will be mainly consumed by organic assets. We refer to this community as *CASWiki (Contextualised ASset Wiki)* and we envision that it will be a collection of localized CASWikis representing each asset in the environment. To the best of our knowledge, CASWiki will be the first framework able to support knowledge production and sharing across communities of automated systems. In what follows we discuss our research roadmap towards the development of CASWiki.

The rest of the paper is structured as follows: in Section II we present scenarios to motivate the need for CASWiki-like approach for in-asset policy generation. We discuss the desired features of the CASWiki in Section III and present a running commentary of a scenario with respect these desired features in Section IV. We then discuss the potential architectural choices in Section V. We provide a plausible validation strategy in Section VI and conclude the paper in Section VII by sketching our near to midterm goals for the CASWiki.

II. Motivation

Today, policies are specified by humans in a centralized and a priori manner; they are then distributed and explicitly enforced on all elements to which they apply. However, in dynamic, distributed systems such as a coalition operating environment, it is very difficult to determine which policy is applicable to which context. The local context may change in ways that would make those policies inappropriate, ambiguous or even obsolete. Additionally, current policy management systems incorporate no cognitive mechanisms for learning from past experiences, and allow little local autonomy.

[1]https://www.wikihow.com

[2]https://stackoverflow.com

Let us consider a smart home scenario in which a smart oven, smoke detector and sprinkler have been working in unison to manage the smoke thresholds in cooking activities—e.g., smoke detector needs to know the different smoke thresholds for oven baking and grilling, and in turn the sprinkler has to know when smoke is not considered a fire hazard. In such environments having a wiki-like system to generate in-asset policy is useful as the oven could educate itself about cooking styles (e.g., grilling vs baking) and deduce the acceptable smoke thresholds for the cooking activity; the smart smoke detector in turn could learn about the acceptable ranges for the oven given the style of cooking, and last but not least, the smoke detector could educate the sprinkler the thresholds in which it should treat smoke as a fire hazard. Additionally, the devices in the environment can record false alarms in the wiki as well as observe the variations in the smoke level overtime with respect to the dishes being cooked to obtain utility functions to make in-context policy decisions.

Now, let us assume that a smart kitchen hood is introduced to the kitchen; with the stored information in the wiki for the smart kitchen, once advertised, the smart kitchen hood can compute the policies for fan speeds for different styles of cooking based on the smoke detector and sprinkler information as the kitchen hood share similar functions to those two assets. Without the contextual wiki, such computations are difficult, if not impossible, for the smart kitchen hood.

Now, let us consider a coalition military scenario that could only be executed effectively and efficiently with a contextual wiki backed generative policy system. Let us assume that two small teams of coalition troops are deployed to perform reconnaissance of a high value target (HVT); let us also assume that these teams are equipped with organic assets that can perform reconnaissance—only policies associated with the assets are to *perform reconnaissance*, *capture images and send to ground platform*, and *return to ground base after the task*. Now, let us assume that one of the teams come under sniper fire while performing the reconnaissance and the organic asset operator associated with the team figures he/she can use the reconnaissance capability of the organic assets to perform surveillance on the potential sniper location; he/she re-tasked the organic asset to do such. The asset then learns that in a *critical situation*, it can function under a new set of policies—namely, *change from a reconnaissance to surveillance task*, *locate sniper and report location*, and additionally it infers that it should *scarifies itself to save human life, if needed*; it stores such information in the contextual wiki it has access to.

Let us now assume that the second team comes under a shell attack near the village of the HVT and even without any action from the organic asset operator in the team, the candidate assets would create policies for themselves to convert to surveillance drones and would report the shell-firing locations to a nearby aerial platform to engage with the enemy as those organic assets would classify the shell-firing event being similar to the sniper attack based on the wiki knowledge. Without the wiki-like knowledge base to support generative policies for devices, this would have resulted in time consuming policy creation and negotiation at command and control, and at worst, resulted in loss of life.

In the next section, we shall discuss the desired features for our CASWiki framework.

III. Desired features of CASWiki

Developing the technologies for the CASWiki is not trivial, requiring a substantial research effort; this is mainly because knowledge has to be precise and expressed in a way that organic assets can understand and use it for their generative actions. Also as knowledge is expressed in the form of policies, the framework has to cater for different types of policies (e.g., constraint policies, goal-oriented policies, and utility policies) in a variety of policy domains.

The research approach underlying the development of our framework combines the following different building blocks. For each building block, we emphasize research challenges and expected novel results.

A. Knowledge Generation for CASWiki

The initial bootstrap of the CASWiki knowledge is particularly challenging for various reasons. First, the specific knowledge depends on many factors, including the policy domains of interest (e.g. security policies, resource sharing policies [5]) and especially types of policies (e.g., constraint policies, goal-based policies, utility-based policies, and so forth)—e.g., in the case of a goal-based policy, the knolwedge must be represented as a workflow of actions that can be executed to meet the specific goal, whereas in the case of a constraint policy, the knowledge should indicate how constraints can be refined, relaxed, restricted depending on the context. For utility functions, the knowledge base needs to provide suitable optimization expressions. Also as in many cases, assets have to interact with other assets, thus knowledge has to include information about which assets have to collaborate with which assets to carry out specific missions and activities. Such a knowledge can be encoded as *interaction graphs* [3], [6]. Second, unlike humans, organic assets do not have inherent knowledge. On the other hand, we have many technologies and approaches today for supporting (collaborative) learning and knowledge bootstrapping that we can explore, exploit and combine together. Here we discuss a few possible approaches that in our view will often be used together to create the knowledge in the CASWiki.

- *Learning from use case scenarios with human assistance:* This approach takes the inspiration from use case scenarios used in software engineering, with recent approaches used to train robots [7]. Under such an approach, several scenarios are generated; then the human assistant will instruct the asset about which actions to take, which policies to use, how to refine policies. The asset will execute these instructions and record everything in the CASWiki, including the scenario, the input received, and the outcome. The main challenge here is to develop significant scenarios. We notice however that scenarios that are generated and tested will also be recorded in the CASWiki and thus can be re-used. Over time, the CASWiki will record a substantial amount of scenarios. Notice also that even for human assistants may not be always trivial to determine which is the best policy to adopt or how to best refine policies. Therefore, it is likely that for the same scenarios different human assistants may provide different guidelines to assets.
- *Machine learning techniques:* Once a number of initial scenarios have been generated, machine learning techniques can be used to extract knowledge—especially domain features—from these scenarios [8]. For example, by looking at the scenarios, one may classify various contexts into clusters and determine how policies are refined and/or which actions is best to take based on the contexts, policy domain, policy type, and asset characteristics. New scenarios can then be generated in which the assets can perform tasks/missions without human assistance based on such knowledge. Results of such executions will be recorded in CASWiki. Additionally, reinforcement learning [9] techniques can be used to enable CASWiki to automatically explore the space of possible actions, assess the outcome of choices made by assets, in terms of rewards and state evaluation, and exploit this knowledge when deciding future actions. Also, relational machine learning, which learns by looking at interactions among the objects of interest, can be used here for learning interaction patterns. It is important to notice here that the knowledge generation is a continuous process. The more activities, scenarios, executions are executed and recorded in the CASWiki, the more the knowledge will grow and be refined.
- *Inferred knowledge through alternative paths:* Planning has been used for decades for automatic goal achievement, especially to create conflict free system—be it to enable intelligent agents working together [10] or to coordinate autonomous airborne platforms [11]. In our recent work, we have shown that such techniques can be used in the Internet of Things domain to compute alternative means to achieve the same effect without violating the constraints in the environment [1], [4]. The idea being is that, preferential knowledge could be learnt overtime so that multiple paths to achieve the same effect can be computed to minimize the violations. We believe such a planner-based approach to compute multitude of possibilities could be used to enhance and augment the CASWiki knowledge. For example, through the use of a planner on the existing knowledge in the CASWiki, we can compute multiple ways in an event could be achieved given conditions. This knowledge—i.e., the inferred knowledge—in turn can be stored in the CASWiki with respect to the associated event, action, and conditions for further discoveries. Additionally, the utility functions created in the use case based knowledge creation could be used in tandem with a planner to rule out the alternative knowledge which may result in unachievable—or prohibitively expensive—actions.

- *Bootstrapping new devices from sharing past experiences:* Knowledge acquired by devices and shared in the CASWiki can provide a validation mechanism for new devices in similar contexts. Analysis of past performance in different context can enable new devices to estimate the level of uncertainty and variability that existing CASWiki policies would have in similar situations, and use this information to predict the CASWiki policies' performance and determine the best action and policy instantiation to perform. Sampling mechanisms can be applied, for instance, to simulate, from existing CASWiki records, scenarios with similar situations and use these scenarios to cross-validate the performance of (variations of) existing policies in order to measure the level of accuracy that they would have in similar situations.

In the next subsections, we discuss the desired properties of a language that can be used to generate, infer, and represent knowledge for CASWiki. We acknowledge that one language formalism may not suffice for this purpose, thus requiring a hybrid language—for example, we envision that it may require combining a highly semantic language such as Web Ontology Language (OWL) [12] with convolution kernels for natural language [13].

B. CASWiki contribution language

The choice of a CASWiki contribution language is critical for a number of reasons: (1) the language should be expressive enough with sufficient semantics to generate, augment, or infer facts in the wiki; otherwise, the wiki will end up containing incomplete and uncertain information which minimizes its utility; (2) the costs associated with the language—be they computational or translational—is critical in high tempo distributed environments where timeliness is a key factor; (3) the language should provide efficient means to align different models as enforcing a single model may not be feasible in coalition settings; and (4) it should enable or have constructs to support humans to digest the information in the wiki through summarizations or rationale to enable critical analysis.

Various languages and technologies can support such representations: from knowledge frames, and the Web Ontology Language (OWL) [12] to controlled natural languages (CNLs) [14], [15]. However, in isolation, none of these languages can fully address the requirements identified above and none of them natively support distributed knowledge representation—e.g., the existing state-of-the-art in policy representation and refinement based on ontologies for ad hoc environments expects a full domain ontology within which policies are interpreted [16], [17]. The focus of this wiki is to allow organic assets to generate policies that are applicable to them by simply utilizing a fraction of the ontology that is relevant to them. Therefore, one important research direction is to investigate suitable language constructs to model the wiki contribution language. We envisage a high-level language, close to natural language, for users to receive summarization of policies and provide input relevant constraints (if necessary), which can be mapped into a low-level ontology-based language for automated generation of policies by the devices. This hybrid language will allow the ease of use of CNLs to be combined with the efficiency of a language like OWL with sufficient mechanisms to generate rationale.

C. Structures for efficient and effective organization of context

The organization of the wiki content plays a key role in its consumability; it does not only provide means for efficient and effective query answering services but also provides seamless integration of disparate pieces of information. We envisage CASWiki as a collection of pieces of information (i.e., each organic asset maintains its own smaller wiki) presented as a single wiki. Thus, modeling provenance aspects of the information held by organic assets is also be critical as well as coming up with metrics to evaluate such information.

In order to build the CASWiki, one interesting research direction is the use and extension of blackboard architectures for multi-agent systems. In such systems, each agent advertises its requirements so that other agents can provide information to satisfy those needs tempered by voting preferences. However, unlike in existing agent-based approaches, we envisage CASWiki to provide semantics to select and generate policies for organic assets—i.e., posts depict what assets did in a particular context with respect to their perceived utility. Therefore, an important research direction for developing the CASWiki approach is to build a new class of blackboard systems where semantics of policy generation are tempered with asset utility and the perceived voter confidence of the posts.

D. CASWiki policy generation and refinement through query

A key aspect of the CASWiki is to enable organic assets to retrieve, generate and refine policies with respect to existing assets; we envisage policy generation and refinement to be byproducts of querying—i.e., lack of results prompting assets to generate policies themselves. This requires a suitable query language that can express: *(a)* what do I need to know—i.e., to retrieve policies indicating constraints related to situations of interest; *(b)* how do I do—i.e., to retrieve policies indicating actions to be executed for meeting certain goals; or *(c)* what do I do—i.e., to retrieve policies to handle unexpected situations and so forth. Additionally, if an asset is new to the environment, it may have to retrieve generative policies from other similar assets to establish policies for itself. In such situations, queries may include input information, such as assets context, which will have to be decided upon by the asset itself by trading-off confidentiality requirements versus query precision.

Once an asset obtains generative policies for itself, those policies may have to be refined to represent the asset in context—e.g., an asset is similar to the class of assets in the environment but has a set of added capabilities available under certain conditions. In other cases, an asset may have different and evolving set of preferences; moreover, when an asset starts instantiating policies based on its needs, it may also need to consult other assets for guidance when unknown situations arise, especially with respect to preferences; thus, each organic asset needs to be augmented with techniques such that preferences of organic assets are learnt—e.g., action with respect to assets context so that the CASWiki is reinforced over time to provide improved in-context policy refinements. One important research direction is thus to build upon recently developed algorithms for preference learning that will allow devices to automatically learn their preference criteria for making choices among possible policies, based on their history and new evidence made available in the CASWiki.

Though there are techniques to efficiently query large knowledge bases using both approximate and precise queries [18], [19], to date, no query language supports the above types of policy retrieval queries for autonomous systems. Therefore, a relevant research direction is to investigate query language features and means to support complex pattern matching scenarios.

E. Methods for assessment of wiki contents and wiki contributors

Approaches have been developed for assessing the quality of content provided by users in community-based platforms and the trustworthiness of users. Many reputation mechanisms are variations of iterative filtering techniques. However, all such techniques have been designed for human users. Thus, the ratings provided are simplistic, typically a score from a set of categorical scores (e.g. excellent, good, and so forth). For the CASWiki whose users are primarily organic assets, more articulated scores are required consisting of several dimensions—e.g., overheads introduced by the use of a certain policy, failures in completing certain actions because of a certain policy, correctness of the policy application with respect to expected outcome, and so forth. Therefore, a major departure with respect to conventional rating systems would be the use of a multi-dimensional rating system whose dimensions are specific to the use of policies by organic assets. This system will be related to the preference learning, which will also be multi-relational and structured across different ranking levels. Once these multi-dimensional ratings are provided, different overall scores can be determined by giving different weights to different dimensions.

In addition, unlike conventional rating systems, the scores will include provenance information concerning the context of policies, as assets are equipped with tooling which can capture a large number of fine-grained context and mission parameters. Such information can be mined for knowledge which in turn will facilitate the re-use of the information by other organic assets. It is important to note that even though CASWiki is mainly oriented towards organic assets, human users will also be able to provide their own rating; however, humans most likely will use conventional rating systems, thus an interesting research direction is how to combine rating systems tailored to human users with rating systems tailored to organic assets. From the theoretical point of view, proofs of convergence of the rating algorithms is critical. One possible direction is to start from approaches developed in [20] and extend them to the context of CASWiki. For protection of the rating mechanisms from attacks, such as colluding attacks, a possible approach is to extend existing approaches based on statistical analysis of deviations of scores reported by organic assets.

F. Securing CASWiki

As knowledge recorded in the wiki may be sensitive, an access control mechanism is needed. Key requirements for such a mechanism include: *(a)* content- and attribute- based access control by which the granted accesses may depend on the knowledge contents and the attributes of the organic assets or human users performing the access; *(b)* different access modes, e.g. read, insert, modify; *(c)* support for cryptographic-based access control for protection in highly insecure environments; and *(d)* delegation techniques by which an organic asset may delegate its access rights to other organic assets Existing access control models and mechanisms must thus be extended for use in the CASWiki setting by tempering them with the contextual information from the CASWiki.

Additionally, mechanisms to guarantee the information integrity is also needed, especially in the CASWiki where the captured and inferred information is used to generate knew knowledge and policies—e.g., if malicious or malfunctioning processes alter the information, trained models will also be incorrect, so are the inferences and the policies generated based on such information. There is a wealth of research out there to address these issues—for example, ipShield [21] is an in-device framework to perform privacy-aware risk assessment of information whereas SenseGen [22] proposes a discriminator model to infer the trueness in information, especially for the training data for machine learning approaches. We aim to utilize such approaches in the CASWiki to address information integrity concerns.

IV. An Illustrative Manifestation of the CASWiki

In order to demonstrate the need for the above desired features in the CASWiki, in this section, we provide a running commentary on their utility for instantiating the smart home scenario we introduced in Section II. To recall, the smart home scenario is as follows: a smart home partially consists of a smart oven, a smoke detector and a sprinkler to manage the smoke thresholds in cooking activities and the question is *if a smart kitchen hood is introduced, how this environment could infer, generate, and place constraints on the devices—and the environment—autonomously* so that the least amount of human input is required to maintain the autonomy of the environment.

In terms of knowledge generation, we could envision a scenario in which manufactures provide different operational conditions for devices—e.g., a smart oven may have different preheating conditions based on the type of the household (apartment vs detached house) and the country of use (Northern Hemisphere vs tropics); a system like ours could utilize this information to automatically create the bootstrapping policies using the latest machine learning techniques—e.g., policy preference inferences using word embeddings [23]. We could also use reinforcement learning techniques to further enhance the knowledge generated—e.g., a user may override the smoke threshold inferred in the bootstrapping period as the layout of the house has changed. This observation can then be used to adapt the behaviors of the smart oven so that its interaction with the sprinkler is improved. We could investigate mechanisms that learn from human users [24] and adaptive techniques based on pattern mining [25] to achieve this goal.

We can also apply planning techniques to further enhance the knowledge in the CASWiki. For example, the approaches in [1], [4] use planning as means to find alternative paths when policy conflicts occur; in our case, we could use similar techniques to compute multiple paths in achieving the same goal and to perform added inferences—e.g., a smart oven is obliged notify users when there are higher readings of smoke, but notifying this could be done in a multitude of manner: from noise to text/visual messages. Using transferable learning, we could project such knowledge to other devices such as smart hood and smoke detector. Finally, we could also investigate formal models for creating realistic scenarios to generate and bootstrap knowledge. The approach in [26] relies on formal models to create realistic data sets for modeling energy consumption at district level for local policy implementation.

The knowledge captured and inferred according to the previous approaches needs to be represented in a way that enables efficient reasoning procedures (i.e., to perform added inferences) and easy consumption—be it other devices or human consumers. We aim to utilize conversational techniques with respect to human and machine understood languages such as Controlled Natural Languages to achieve the ease of consumability [27]. For example, when the smart oven is tuned by a human user for smoke threshold, the oven could have a conversation with the user to critique the actions based on its already existing knowledge. Once learnt, the oven could update its knowledge base as well as broadcasting the learnings to the other devices for their education.

Once the knowledge is captured and represented, the CASWiki needs mechanisms to allow the users (both devices and humans) in the environment to query the knowledge, especially for policy generation in previously unencountered situations. For example, the smart oven may need to know the safe smoke thresholds for meat grilling and the smart cooker, hood and the smoke detector in unison may be able to provide the needed information—or the insights—to generate the required policies for the oven in the meat grilling context. Additionally, we aim to use the new notions proposed in pragmatically-aware query reformulation strategies [18] to better support query answering in the system—hence improved policy generation—so as to better interpret the query parameters (e.g., what it is meant to be a safe smoke threshold) in the context.

When consuming the knowledge represented in the CASWIki, the users need to be able to temper the knowledge with respect to the trust and reputation matrices they have—or the system has—derived for the knowledge sources. For example, the smart oven may learn overtime that the manual tunings done by a particular human user is always overridden by the other human users, thus requiring it to temper the actions of the said user before recording or broadcasting. However, this is not always as simple as giving prominence to mass knowledge, since the knowledge held by the mass could be incorrect. This may be due to an adversarial attack or malfunctioning of a class of devices. Thus, it is also important to have proper security mechanisms to guard against such phenomena. For example, imagine a situation in which the knowledge held by the smart cooker and the hood is corrupted due to an erroneous external procedure, thus resulting in wrong settings for grilling meat. If the smart oven truly relies on the recorded information and does not have access to procedures to calculate the fitness for purpose in information, the smart oven may also result in over- or under- cooking the meat.

V. Physical Manifestation of the CASWiki

While the CASWiki can be thought of as a publicly available Internet based service, e.g. as the current sites of wikipedia[3] or wikihow[4], it can physically manifest itself in many different ways. In this section, we look at the different physical architectures for the CASWiki, and discuss the benefits and limitations of each architectural approach.

As discussed previously, the logical CASWiki consists of a service to which autonomous organic assets *(a)* upload new knowledge about the means in which situation were handled; and *(b)* send a query to retrieve existing knowledge about how others assets may have handled a situation they currently encountering. The piece of knowledge will be represented using the CASWiki contribution language and it will be queried via the CASWiki query language as described earlier, with an independent set of users that will perform assessment of the knowledge that is being uploaded. The interaction among the different users of CASWiki can be characterized as a series of API calls, some being write requests and the bulk being read requests. The write requests are the results of an automated device storing a new piece of knowledge, or an assessment system deciding to merge, update or delete existing pieces of knowledge. The read requests will be queries from different automated assets that are looking for existing pieces of information.

In this regard, the logical CASWiki structure is similar to traditional databases or directories, which support a set of APIs supporting read requests and write requests. The structure of the requests in the CASWiki is much more complex than traditional databases or directory APIs, but the physical implementations can be done in a similar manner. Several possible different physical manifestations of CASWiki can be realized, and some of them are listed below. The right physical implementation could be any one of these, or a combination of all the approaches, depending on the conditions under which CASWiki needs to operate.

A. Centralized CASWiki

The simplest implementation of the CASWiki would be a central structure, in which both requests and responses

[3]https://www.wikipedia.org
[4]https://www.wikihow.com

are stored within a central repository. This repository can be supported by means of any database, although the best supporting system would likely be an object oriented database such as MongoDB [28]. With this architecture, the CASWiki implementation consists of a set of read and write APIs, each of which are processed by a processing function in the system which takes each piece of knowledge represented in the CASWiki language, and stores it in the database.

In addition to the processing and querying of the objects, the CASWiki also needs to enable an interface for assessment and consolidation of the knowledge. These could be done by means of ratings, or by an independent assessor. Access control to the knowledge objects is done by means of access control policies, which determine who can read and write the pieces of knowledge that are stored in the CASWiki.

B. Federated CASWiki

In many situations, a central wiki is not a viable solution. In the case of coalition operations, each partner may want to operate their own CASWiki. At the same time, the coalition partners may also want to share knowledge with each other, or at least share fragments of the knowledge with others.

A federated wiki can be implemented by extending approaches for dynamic database federation [29] to the specific case of CASWiki. One efficient solution would involve that each member of the federation maintains its own knowledge (i.e. it does not forward the write requests to other members of the federation), but is willing to share the knowledge—i.e., willing to address the read requests of other members in the federation. In such a system, queries made to any single instance of CASWiki are forwarded to all other instances in the federation, and each instance of the CASWiki replies with any relevant answers it has within its individual repository. The originator CASWiki combines all the responses and sends back a merged response to the automated device making the query.

Optimizations to improve the performance of such a federation can be implemented, and the nature of the optimization depends on the structure of the federation. In federations which consist of a small number of members where new knowledge is created relatively slowly, caching of the responses from other federated members can result in an efficient implementation which reduces the network traffic among the federated members. In federations which consists of a large number of instances of CASWiki, creating a network structure among the members and sending queries along that network structure could reduce network traffic and increase the amount of cache hit that may result in the federation. Other optimizations can use concepts from peer-to-peer overlay networks [30] to further improve the performance.

C. Distributed CASWiki

The federated approach described above did not optimize the structure of the CASWiki based on the contents of the knowledge repository. Instead, it relies on the flooding approach to consolidate information from all the constituent members. Further optimizations can be obtained by considering the contents of the CASWiki, and using the contents of the query to locate the information. Techniques such as Information Centric Networking [31] and Distributed Hash tables [32] provide approaches for distributed CASWiki that may be more efficient than straight-forward federation.

Following the Information Centric Networking or named based networking paradigm, each piece of knowledge that will be created by a write operation can be given a unique name. The name may be created in a structured manner to define the scope to which the created knowledge belongs, e.g. by having a well-defined hierarchy into which problems, and the policies or processes related to them are mapped to. The name is used to determine the location where the knowledge will be stored. An automated device can search for a named entity, and use approaches from Named Based Networking to find the answer. The

Distributed Hash tables provide an alternative approach for physical instantiation of a distributed wiki. In these systems, a hash is computed for the contents of the knowledge, and the hash determines the physical instance of the CASWiki where the knowledge will be stored. The system can insert new objects very efficiently or locate existing objects very efficiently using the hash.

When querying the contents of a named based implementation, the determination of the right hash for the object would be the first step in making the query. A hybrid implementation, which floods a query using the federation approach, allows each member of a distributed CASWiki to respond just with a hash along with some properties, which allows the query maker to use the name or hash to efficiently retrieve the best answer, provides a system which is both efficient as well as user-friendly.

VI. Validation Plan

Our ultimate goal is to investigate the impact and the utility of a community-driven approach for deriving semantics for in-context policy generation for and by the organic assets supporting collaborative environments. Validating the CASWiki framework requires both theoretical and empirical verifications. We need to verify that the: *(a)* the proposed approach is valid; *(b)* the knowledge representation and query answering techniques are sound, complete, and consumable; and *(c)* the techniques scale and adopt with respect to evolving conditions in the environment. Theoretical evaluations can be carried out through simulations, but since there are a number of technologies involved in the creation of the CASWiki (e.g., knowledge creation and inference, knowledge formalisms for the wiki contribution language and the wiki query language, maintenance of wiki content, and so forth), experimental studies are also critical in order to evaluate the efficacy of individual components as well as collectives with subject matter experts. Our envisioned techniques will also be applied to different policy domains and their effectiveness would be compared to that of existing alternatives.

It is also critical to develop methods for assessing content and contributors for the wiki, including provenance aspects of the information and wiki access control. Studies need to be conducted to determine their accuracy and operational characteristics. Developing a reference implementation of the CASWiki concept is also crucial in order to assess its feasibility. Finally an important evaluation activity is represented by simulation experiments to explore the efficacy of the CASWiki under various conditions for large-scale networks of organic assets.

Operationally, the appropriateness and effectiveness of the policies generated from the information exchanged in the CASWiki is the most important consideration. Experiments will have to be conducted to compare the policies generated by the proposed approach to those that would have been generated by humans under the same circumstances; this could be achieved by means of approaches such as mechanical turk [33] to cloud-based facilities for repeatable experimentation [34]. This in turn would require a detailed modeling of realistic scenarios.

VII. Conclusion

Technological advances are making cognitive autonomous assets a reality. However for such assets to effectively function in a variety of (unforeseen) contexts and tasks, providing them with suitable knowledge is critical. In this paper we have introduced a novel community-based approach by which such knowledge is generated by the assets themselves with possibly human mediation and supervision. Some of our focus is on knowledge represented in form of policies driving the behavior of assets. In a way, our vision is that assets should learn from each other and thus the goal of community-based wiki of assets is to support such learning processes. There are formidable research challenges towards developing tools and techniques to make our vision a reality. The paper has outlined a few challenges and initial approaches. However much work remains to be done in order to further articulate the challenges and the solutions.

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] E. Göynügür, G. de Mel, M. Sensoy, K. Talamadupula, and S. B. Calo, "A knowledge driven policy framework for internet of things." in *ICAART (2)*, 2017, pp. 207–216.

[2] D. Verma, S. Calo, S. Chakraborty, E. Bertino, C. Williams, J. Tucker, B. Rivera, and G. R. de Mel, "Dynamic and adaptive policy models for coalition operations," in *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VIII*, vol. 10190. International Society for Optics and Photonics, 2017, p. 1019006.

[3] D. Verma, S. Calo, S. Chakraborty, E. Bertino, C. Williams, J. Tucker, and B. Rivera, "Generative policy model for autonomic management," in *Proceedings of 1st International Workshop on Distributed Analytics InfraStructure and Algorithms for Multi-Organization Federations*, 2017, pp. 4–8.

[4] E. Göynügür, S. Bernardini, G. de Mel, K. Talamadupula, and M. Şensoy, "Policy conflict resolution in iot via planning," in *Canadian Conference on Artificial Intelligence*. Springer, 2017, pp. 169–175.

[5] E. Bertino, S. Calo, M. Touma, D. Verma, C. Williams, and B. Rivera, "A cognitive policy framework for next-generation distributed federated systems - concepts and research directions," in *IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 1876–1886.

[6] Y. Wang, H. Sundaram, and L. Xie, "Social event detection with interaction graph modeling," in *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012, pp. 865–868.

[7] C. Claudia erez DArpino and J. A. Shah, "C-learn: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy," in *IEEE International Conference on Robotics and Automation (IEEE ICRA 2017)*, 2017, pp. 4058–4065.

[8] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial intelligence*, vol. 97, no. 1, pp. 245–271, 1997.

[9] L.-H. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Machine learning*, vol. 8, no. 3/4, pp. 69–97, 1992.

[10] K. P. Sycara, "Resolving goal conflicts via negotiation." in *AAAI*, vol. 88, 1988, pp. 245–250.

[11] V. N. Duong and K. Zeghal, "Conflict resolution advisory for autonomous airborne separation in low-density airspace," in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, vol. 3. IEEE, 1997, pp. 2429–2434.

[12] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, "Owl 2 web ontology language primer," *W3C recommendation*, vol. 27, no. 1, p. 123, 2009.

[13] M. Collins and N. Duffy, "Convolution kernels for natural language," in *Advances in neural information processing systems*, 2002, pp. 625–632.

[14] T. Kuhn, "A survey and classification of controlled natural languages," *Computational Linguistics*, vol. 40, no. 1, pp. 121–170, 2014.

[15] D. H. Mott, D. R. Shemanski, C. Giammanco, and D. Braines, "Collaborative human-machine analysis using a controlled natural language," in *SPIE Sensing Technology+ Applications*. International Society for Optics and Photonics, 2015, pp. 94 990J–94 990J.

[16] M. Sensoy, T. Norman, W. Vasconcelos, and K. Sycara, "Owl-polar: A framework for semantic policy representation and reasoning," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 12, no. 0, 2012. [Online]. Available: http://www.websemanticsjournal.org/index.php/ps/article/view/232

[17] L. Kagal, T. Finin, and A. Joshi, "A Policy Language for A Pervasive Computing Environment," in *IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, June 2003.

[18] A. Viswanathan, G. de Mel, and J. A. Hendler, "Pragmatics and discourse in knowledge graphs," 2015.

[19] H. Huang, C. Liu, and X. Zhou, "Approximating query answering on rdf databases," *World Wide Web*, vol. 15, no. 1, pp. 89–114, 2012.

[20] M. Rezvani, A. Ignjatovic, E. Bertino, and S. Jha, "Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks," in *IEEE Transactions on Dependable and Secure Computing*, vol. 12, 2014, pp. 98–110.

[21] S. Chakraborty, C. Shen, K. R. Raghavan, Y. Shoukry, M. Millar, and M. B. Srivastava, "ipshield: A framework for enforcing context-aware privacy." in *NSDI*, 2014, pp. 143–156.

[22] M. Alzantot, S. Chakraborty, and M. Srivastava, "Sensegen: A deep learning architecture for synthetic sensor data generation," in *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*. IEEE, 2017, pp. 188–193.

[23] S. Gurciullo and S. Mikhaylov, "Detecting policy preferences and dynamics in the un general debate with neural word embeddings," *arXiv preprint arXiv:1707.03490*, 2017.

[24] A. L. Thomaz, C. Breazeal *et al.*, "Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance," in *Aaai*, vol. 6, 2006, pp. 1000–1005.

[25] P. Rashidi and D. J. Cook, "Keeping the resident in the loop: Adapting the smart home to the user," *IEEE Transactions on systems, man, and cybernetics-part A: systems and humans*, vol. 39, no. 5, pp. 949–959, 2009.

[26] V. Cheng and K. Steemers, "Modelling domestic energy consumption at district scale: A tool to support national and local energy policies," *Environmental Modelling & Software*, vol. 26, no. 10, pp. 1186–1198, 2011.

[27] N. OLeary, D. Braines, A. Preece, and W. Webberley, "Conversational homes," in *9th International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE17)*, 2017.

[28] K. Chodorow, *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. " O'Reilly Media, Inc.", 2013.

[29] G. Bent, P. Dantressangle, D. Vyvyan, A. Mowshowitz, and V. Mitsou, "A dynamic distributed federated database," in *Proc. 2nd Ann. Conf. International Technology Alliance*, 2008.

[30] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.

[31] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, 2012.

[32] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Looking up data in p2p systems," *Communications of the ACM*, vol. 46, no. 2, pp. 43–48, 2003.

[33] G. Paolacci, J. Chandler, and P. G. Ipeirotis, "Running experiments on amazon mechanical turk," 2010.

[34] K. Marcus, M. Touma, and F. Bergamaschi, "Network science collaborative experimentation methods and tools to accelerate network science innovation within the ita and ns-cta," in *Military Communications Conference, MILCOM 2015-2015 IEEE*. IEEE, 2015, pp. 1385–1390.