Accelerating Humanoid Robot Learning from Human Action Skills Using Context-Aware Middleware

Charles C. Phiri, Zhaojie Ju, and Honghai Liu

School of Computing, University of Portsmouth, UK

Abstract. In this paper we propose the creation of context-aware middleware to solve the challenge of integrating disparate incompatible systems involved in the teaching of human action skills to robots. Context-aware middleware provides the solution to retrofitting capabilities onto existing robots (agents) and bridges the technology differences between systems. The experimental results demonstrate a framework for handling situational and contextual data for robot Learning from Demonstration.

Keywords: Context-aware middleware. JSON-LD. SLAM.

1 Introduction

Smart environments apply sensor technology to constantly monitor the spaces for variable controlling interests and conditions. A robot operating in this smart environment could adjust its belief-set and modify its decisions in response to the additional stimulus. While accessing the computable data is challenging, making online decisions in a timely manner is even more difficult. On the other hand, if the access link between the robot and the smart data source is compromised, the robot could be left exposed. In this paper, we propose a framework for deriving posterior beliefs from smart environments to enable adaptable context-sensitive behavior in robots. We explore how to add online adaptive behavior to enable robot Learning from Demonstration, LfD, allowing end-users to intuitively teach robots new tasks without programming knowledge.

Increased availability of ultra-low-power sensor devices is continually driving down the cost and power budget required to embed connected sensing technology within human environments [17, 20] creating observable smart environments. On the other hand, increasingly robots are being introduced to share human spaces. The ability to take advantage of the shared capabilities with humans has made humanoid robots among the most common types of robots being introduced in human environments. Humanoid robots can access the same tools and facilities as humans and are considered expendable, which makes them useful in disaster recovery and situations or places which are not compatible with human comfort and life [12]. Programming robots to exhibit emergent intelligent behavior in a stochastic environment requires specialist skills.

Unlike record and replay methodology, LfD, focuses on learning and generalization of observed demonstrations in order to perform the task [4]. By learning what it means to perform the task, the robot can make the correct decision at the right time; this is generally agreed as emergent intelligent behavior. Learning the meaning of a task involves the interpretation of the context in which the task is performed. Rich contextual data enables the robot to make decisions about situations and events at the right time improving the learning rate, acquisition of new tasks and their variations. The perceptual and physical equivalence problems are abstracted by providing the context data in a universally accessible format allowing the robot to make decisions based on its own embodiment and sensory perception above. This allows a symbiotic collaborative learning environment in which the observable state-space is richly mapped and computational localization data for the robot, objects and obstacles is highly available[5].

Conditional discrimination and disambiguation of context allows the context to be refocused and directed to the observable problem space. On the other hand, by classifying the robot's perception data as immediately required or not, the robot could distribute the computation of fringe details necessary for deriving posteriors to edge systems which can then provide the feedback as actionable data with the appropriate weighting. This allows the robot to form part of the interactive ecosystem with the smart spaces [5]. The architecture presented in this research does not hinder the application of the necessary security concerns. Although the experiments in this paper use the Aldebaran NAO H25 robot platform[2] the concepts explored are largely language and platform agnostic.

The robot uses its stereo cameras to estimate its location on a map that may be provided as prior information or estimated online. The motion planning (including collision avoidance) algorithms are determined using localization and mapping data estimated and calculated using onboard belief-set and reference frame. Using the smart environment, the robot location, landmarks, and map of the environment can be estimated using a global reference frame. The main challenge of motion planning can then be restated as a spartiotemporal correspondence problem. The robot is this work uses Active Online SLAM for motion planning. Fig. 1 shows the Active Online SLAM algorithm summary based on [22, 23].

A fixed Microsoft Kinect multisensor device is used to provide context data. Active Online implies that when Kinect data is not available, the robot may apply the same strategy using the onboard sensors with an increased computation burden and reduced field of vision. The mapping and localization data from the robot is also published allowing other robots and agents in the smart space to implement cooperative strategies with a high degree of complexity in reasonable timescales [12] based on the context data. Presence information plays a key part in determining the utility of the context data. The method proposed lets heterogeneous robots and smart environments to cooperate on tasks while permitting a high degree of autonomy for the robot [1]. We propose a context-aware middleware that allows events and actionable inputs from external agents to be applied to the online control policy determination, including the reduction of the latency between context acquisition to the derivation of actionable control policy. We decompose LfD into a motion planning problem in this environment. Both the robot and human instructor are observable in a shared state space.

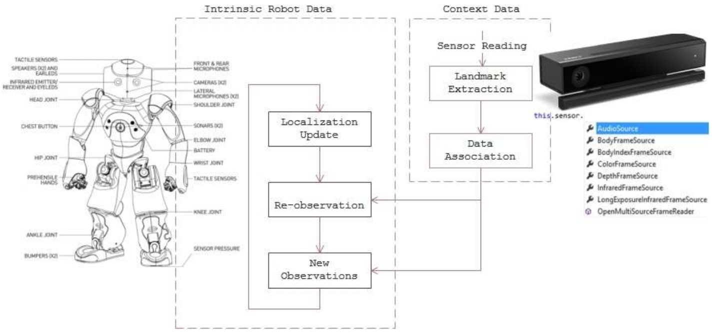


Fig. 1. Leveraging Distributed Common Context

1.1 Related Work

The Beecham sector map shows segmentation and availability of data sources constituting smart spaces [20]. Mesh networks have been proposed by researchers [16] as a way of alleviating the communication challenge of collaborative agents. 5G networks are set to be the core M2M communication technology for IoT enabling immersive software defined networks [15]. OpenFlow and SDN offer methods of separating the control plane from the service delivery [18]. The proposed methodology in this research does not compete with these transport infrastructure-heavy technologies but rather adds a complementary cross-domain service for sharing contextual knowledge between heterogeneous controllers in a common configuration space.

Kim, Cho [11] propose a context-aware middleware service for networked robots based on Common Object Request Broker Architecture, CORBA, technology. Context between agents is shared using the distributed object paradigm. A system that recognizes situations automatically and uses various sensors to observe and adjust its current knowledge is termed context-aware [3]. Babu and Sivakumar [3] propose an adaptive and autonomous context-aware middleware for multi-agents with Type 2 Fuzzy rough context ontology. In their work, they provide a model that allows context collection, context processing and application reaction to significant context changes.

Smart spaces are a product of the paradigm of Ambient Intelligence [13]. Context-awareness breaks the isolated clusters of computable knowledge and distributes it among agents and controllers thereby allowing collaborative learning [21]. Li, Eckert [14] present a survey of context-aware middleware between

2009 and 2015. In their work they show the graph in Fig. 2 which breaks down the process of acquisition of context through to distribution to the participating agents [2]. Li, Eckert [14] also present a very good summary of the prevalent context modeling techniques, their advantages, disadvantages and applicability.

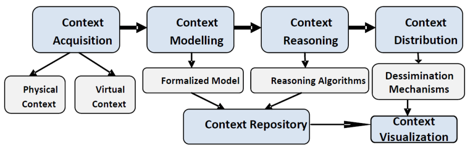


Fig. 2. Context-Modeling [8]

Most recent and noteworthy in this domain is Googles project Tango [8] which has an express desire to use computer vision to give devices the ability to understand their position relative to the world around them. Project Tango uses machine learning algorithms to understand the Motion Tracking, Area Learning, and Depth Perception. Area Learning uses an, Area Description File, ADF, to store the area map data and devices within it. The ADF can either be extracted during exploration or prepared beforehand and shared accordingly. At Google I/O 2015, Google in partnership with Infineon Technologies demonstrated a 60GHz radar-based sensor that allows highly sensitive decomposition of human action skills [7]. Technology explored in the presentation would greatly improve accessibility of human action skills in humanoid Learning from Demonstration.

2 System Architecture and Overview of Concept

Fig. 3 shows the deployment view of the multi-sensory system model implementing the concept with the Kinect as an example data source. Other suitable sensors can equally be used [20]. The middleware receives the data, shapes the data and distributes messages to all involved parties in the topic using the publisher-subscriber pattern. The messages are formulated as JSON-LD objects.

The topics provide a grouping concept for the context, a bounding condition. The bounding data define the boundaries of the search space according to the context defined. This may be used to limit the search space or even apply business rules, as may be applicable, enabling the robot to determine relevancy of both the source and the context data to the task in hand.

Looking at communication as a bounding condition, for example, the robot may, consider the quality of the signal based on the RSSI of the air interface to predict the expected QoS and from which it can then determine whether it should outsource computation of non-critical fringe data without breaking soft real time constraints of the task in hand. If the link is acceptable, a second level constraint may be based on availability of mapping and localization data from the Kinect Data source and the latency of sourcing this data, for instance.

Based on the context data the robot probabilistically determines to employ a computationally less expensive (O(n)) Spike landmark detection algorithm using the onboard stereo cameras to check for obstacles in its immediate environment thereby reducing both the search space for optimal solutions and in turn decreasing the contributing power budget. Spike landmarks are recognized by detecting pixels that protrude from their surrounding pixels. The context data includes labeled originators so that the agents can selectively ignore the irrelevant sources [22, 23]

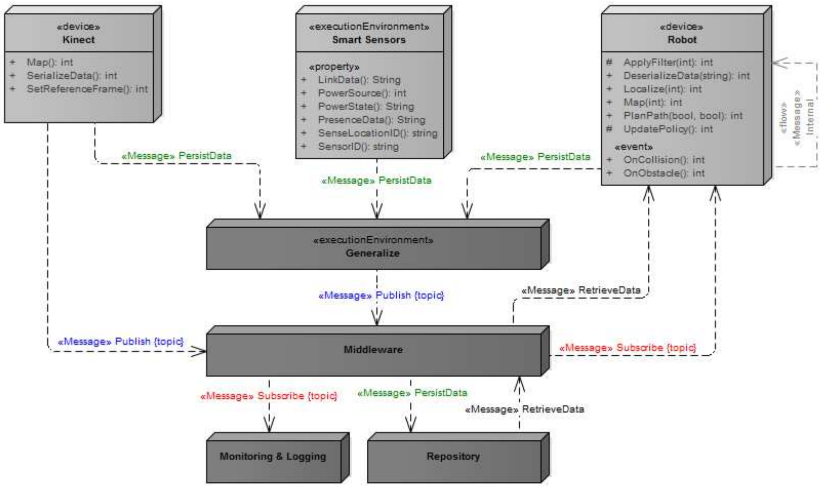


Fig. 3. Architecture Deployment View

The context data is the generalized data from both the smart environment and the participating robots. The artifacts propagated to the robot or persisted to the repository are formatted to be immediately available for computation of the control policy. The system may also add bounding context data before the computation of the policy [22]. The proposed middleware is completely auditable allowing all contributors to be appraised and monitored via a non-intrusive monitoring channel Fig. 3. The robot can decide the levels of exploration and exploitation to employ depending on availability of the relevant computational model and the distribution of resources [22, 23].

2.1 Context-Aware Middleware

Context-awareness enables service discovery and adaptation of computing devices based on the availability of ambient intelligence. It is generally agreed that context-aware system should be responsive to multi-agents, covering a large number of devices, assisting a large number of system actors and serving a large number of purposes [3]. Context-aware computing is a style of computing in which situational and environmental information is used to anticipate immediate needs and proactively offer enriched, situation-aware and usable content, functions and experiences [19]. Context-aware systems can sense their physical environment, and adapt their behavior accordingly. Knowledge representation of features extracted from the environment involves prioritization of what is needed. An autonomous robot has to make these determinations in real time. By providing computable context the robot, given prioritized data within a bounded configuration-space, can find a more directed solution in the in a shorter period of time. The wrong context or estimation, however, could end up with a highly biased greedy robot [11, 22, 23]. In this study, by providing the repository, contextual memory is enhanced allowing exploitation of previously mapped environment. The repository is implemented using Hopfield Network which forms an associative memory controlled by a Lypunov Function [9].

The challenge with context is that it is recursive. It represents the situation and communication of the ambient data to the agents about the environment in which the said agent operates. For a shared context between multiple systems, the context must be expressed in such a way that it is understood by all parties. Ontology modeling offers a context modeling tool resulting in rich descriptive models providing a shared understanding between the agents [10, 18–20]. The task starts by determining the ontology of the context. In this framework, the task only needs to be as complicated as it take to represent the context as W3C JSON Link Data, JSON-LD [10].

JSON-LD format is built on top of and is compatible with JSON (RFC7624). This allows serialization and deserialization of W3C RDF ontology data to JSON. JSON-LD simplifies RDF for both machines readability and human inspection [10]. The JSON-LD messages are published using ZeroMQs, ZMQ, extended publisher-subscriber pattern, XPUB-XSUB. XPUB-XSUB pattern allows multiple subscribers with very minimal overheads. The entire architecture takes advantage of the ZMQs lightweight scalability to share data across domains [6]. JSON-LD forms a well-defined technology neutral interface between participating applications removing the need of deep integration between components which may be heterogeneous. Messages can be decomposed and forwarded to the target without regard to implementation of either the source of the messages or the sink. As only messages are shared, the solution is platform, language and OS agnostic.

Unlike standard JSON messages, JSON-LD has defined primitives which allows a common understanding of the self-describing JSON document.

A critical fundamental system requirements guiding the framework sets that the computation data is delivered in soft real time and that the data is reliable, the system is highly observable and auditable. Fig. 4 shows the component view of the architecture focusing on the communication pattern. This illustrations the realization of scalable middleware using ZMQ which has very low latency and handles concurrency and self-discovery. In this configuration, the added advantage is that subscribers and publishers may join or leave without affecting the other. ZMQ also proposes strategies for dealing with slow subscribers such as encouraging them to drop the connection. Using ZMQ, the framework can handle multiple publishers and subscribers segregating them using topics. Although Fig. 4 shows a homogeneous ZMQ solution, ZMQ is designed to allow distributed multi-protocol integration and deployment.

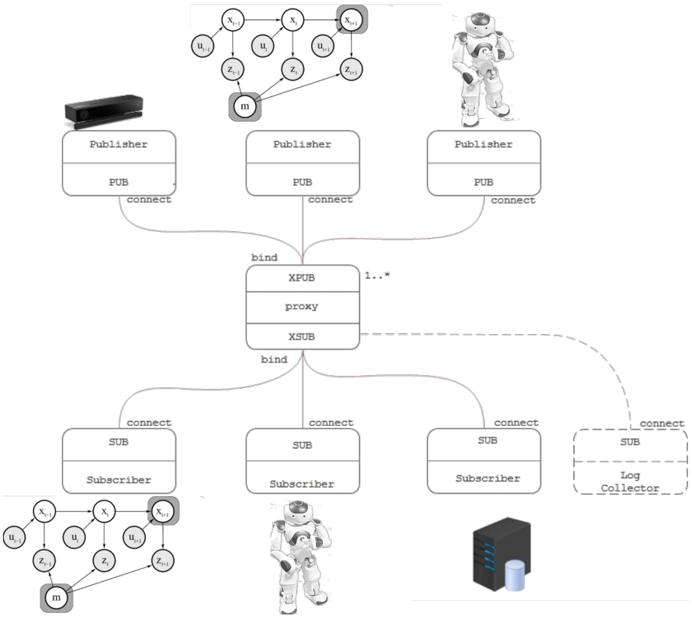


Fig. 4. Scaling with ZMQ

3 Preliminary Results Using SLAM

The use case employs the probabilistic EKF-SLAM algorithm to show applicability of the framework in LfD. Simultaneous Localization and Mapping, SLAM, is an example of a context problem that involves use of the best view of the world given the resources and information available to map stochastic environments. SLAM formally is concerned with the problem of building a map of an unknown environment by a mobile robot while at the same time navigating the environment using the map. SLAM consists of multiple parts; Landmark extraction, data association, state estimation, state update and landmark update. There are numerous solutions to the SLAM problem but in this paper we focus on the implementation of a software solution to aid sharing information to improve the observation of the state space to bias the (Active) Online SLAM problem towards a localization or mapping problem by taking advantage of collaborative learning [22, 23].

Fig. 5 shows feature extraction and classification from both the robot and the human instructor. Feature extraction in LfD is a high dimension problem. PCA is applied to reduce the problem space. The feature selection applies weighting to the feature to classify topical relevance. Relevance is contextual. The generalized output data is framed into a JSON-LD message and published via the middleware to listening subscribers. The robot uses the context information to adjust its trajectories obtained using Online Active SLAM. The human instructor, the robot and the state space are observable using the Kinect multi-sensor device which build. The Kinect currently provides the map and localization data based on the vision system only, however, this other sensors could equally be applied. The challenge is to solve the correspondence problem when switching reference frames.

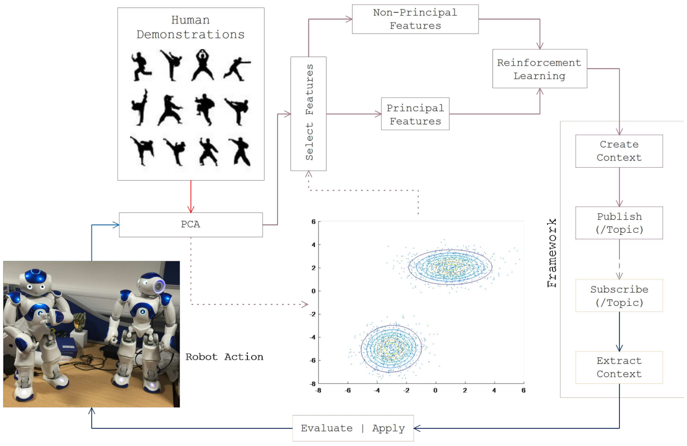


Fig. 5. Feature Extraction Using Principal Component Analysis, PCA and Reinforcement Learning to Build Context for Distribution.

The output of the work is the architecture of a message-based generic context-aware framework and how to implement it. For completeness, however, we briefly describe the notions behind SLAM to illustrate how it fits into the framework.

Given

The Robot's Controls: $u_{1:T} = \{u_1, u_2, u_3, , u_T\}$
Observations: $z_{1:T} = \{z_1, z_2, z_3, , z_T\}$

Wanted

Map of the environment: m
Path of the Robot: $x_{1:T} = \{x_1, x_2, x_3, , x_T\}$

Full SLAM represents an estimate of the entire path and map expressed as shown in equation (1):

$$p(x_{1t}, m|z_{1t}, u_{1t}) \tag{1}$$

Online SLAM, on the other hand, only considers the most recent pose and map. Online SLAM is expressed in terms of equation (2) and has the graph model shown in Fig. 6:

$$p(x_t, m|z_{1t}, u_{1t}) = \int\int ... \int p(x_{1t}, m|z_{1t}, u_{1t})dx_1 dx_2 ... dx_{t-1} \tag{2}$$

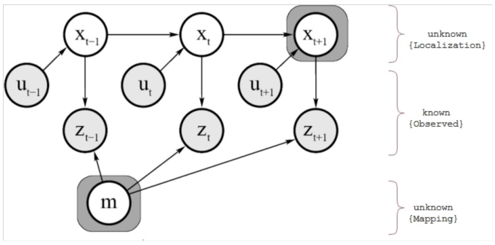
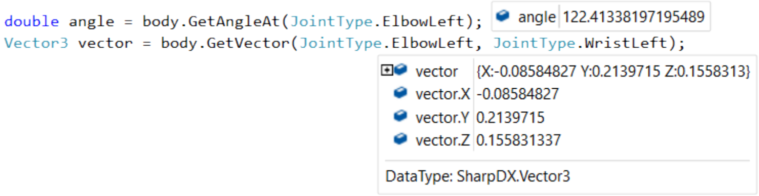


Fig. 6. Online SLAM

Recursive based filter is applied to the robots estimated perception data to adjust the map. Extended Kalman Filter, EKF, is then applied to improve the noisy sensory data to obtain a more accurate model. Thrun et.al [22, 23] have shown how SLAM techniques scale to real world challenges including the application of the Victoria Park Data Set in vehicular self-navigation tasks [22]. In this study, the focus is on LfD. The pose estimates of the human instructor are serialized and made available to the robot to reproduce using its own joint coordinate frame. The objects representing the human and humanoid robot joint coordinates are serialized to the same JSON-LD structure opening up a future possibility to automate the scoring of the learning effort as the data can be accordingly scaled and the directly compared. Fig. 7 shows Kinect joint and vector data objects. The data can be represented in the Kinect Depth or Camera Space.



```
double angle = body.GetAngleAt(JointType.ElbowLeft);     angle  122.41338197195489
Vector3 vector = body.GetVector(JointType.ElbowLeft, JointType.WristLeft);
```

```
⊞ vector    {X:-0.08584827 Y:0.2139715 Z:0.1558313}
  vector.X  -0.08584827
  vector.Y  0.2139715
  vector.Z  0.155831337
DataType: SharpDX.Vector3
```

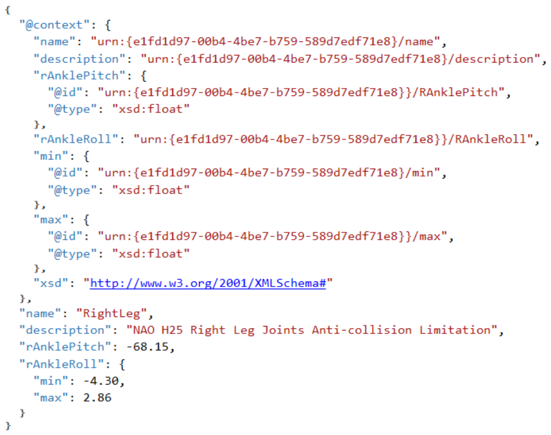
Fig. 7. Kinect Joint Angles and Bone Vector Data using KinectEx Extensions.

Looking at NAO H25 Kinematics frame, the data in table 1 shows the limits of the right ankle joint to avoid collisions with its own body frame. Taking the details of the first row (in degrees) in table 1, the data is formulated into RDF ontology using JSON-LD [10] as shown in fig. 8. The object data represented as JSON-LD shows a directed graph of the linked objects. JSON-LD, being JSON, is self-descriptive.

Table 1. NAO H25 Right Leg Joints Anti-collision Limitation [2]

RAnklePitch	RAnkleRoll Min	RAnkleRoll Max	RAnklePitch	RAnkleRoll Min	RAnkleRoll Max
(degrees)			(radians)		
-68.15	-4.3	2.86	-1.189442	-0.075049	0.049916
-48.13	-9.74	10.31	-0.840027	-0.169995	0.179943
-40.11	-12.61	22.8	-0.700051	-0.220086	0.397935
-25.78	-44.06	22.8	-0.449946	-0.768992	0.397935
5.73	-44.06	22.8	0.100007	-0.768992	0.397935
20.05	-31.54	22.8	0.349938	-0.550477	0.397935
52.87	-2.86	0	0.922755	-0.049916	0

Using the W3C JSON-LD 1.01 format,Fig. 8, the @context symbol encapsulates metadata carries forward knowledge that the consumer uses to understand the data types and purpose of the tuples. The resource name attached to the GUID (urn:e1fd1d97-00b4-4be7-b759-589d7edf71e8) allows bounding of the context to a particular NAO robot data set. The @type forward declares the data as W3C XML Schema float data type which, in turn, defines the boundaries of expected data. As the standard requires that the JSON-LD be valid JSON, it also follows that the data can be consumed without knowledge of either RDF or indeed JSON-LD [10], at the risk of losing most of the intrinsic benefits of the formats.



```
{
  "@context": {
    "name": "urn:{e1fd1d97-00b4-4be7-b759-589d7edf71e8}/name",
    "description": "urn:{e1fd1d97-00b4-4be7-b759-589d7edf71e8}/description",
    "rAnklePitch": {
      "@id": "urn:{e1fd1d97-00b4-4be7-b759-589d7edf71e8}}/RAnklePitch",
      "@type": "xsd:float"
    },
    "rAnkleRoll": "urn:{e1fd1d97-00b4-4be7-b759-589d7edf71e8}}/RAnkleRoll",
    "min": {
      "@id": "urn:{e1fd1d97-00b4-4be7-b759-589d7edf71e8}/min",
      "@type": "xsd:float"
    },
    "max": {
      "@id": "urn:{e1fd1d97-00b4-4be7-b759-589d7edf71e8}}/max",
      "@type": "xsd:float"
    },
    "xsd": "http://www.w3.org/2001/XMLSchema#"
  },
  "name": "RightLeg",
  "description": "NAO H25 Right Leg Joints Anti-collision Limitation",
  "rAnklePitch": -68.15,
  "rAnkleRoll": {
    "min": -4.30,
    "max": 2.86
  }
}
```

Fig. 8. JSON-LD representation of NAO H25 Right Leg Joint Limitation

4 Conclusion

In this work, we shown the concept and architecture of a context-aware middleware to improve on the availability and accessibility of the simultaneous localization and mapping data for robot learning from demonstration, LfD. We have demonstrated a cooperative strategy using context data to derive more accurate posterior beliefs based leveraging the global view provided by smart environment to map the state space and locate self within the state space as well as locating obstacles and landmarks. For the Active Online SLAM implementation the robot is able to switch between onboard sensors and the environment sensors and can offer redundant data to improve it confidence. Associative memory allows the robot to choose between exploitative or exploratory schedule [22, 23]. By using standard protocols, the knowledge-sharing is kept transparent to the transport mechanisms used such that only topical subscribers can take advantage of the published information regardless of the connectivity infrastructure [6, 10]. Using low-latency transfers and caching, the robot can off-board some other non-critical computation without compromising on intelligent behavior. Context-aware middleware provides the solution to not only retrofitting capabilities onto existing robots (agents) but also bridges the technology debt within future and existing system architectures. Preliminary results using LfD show that these features help accelerate the learning effort for humanoid robots.

In future work, we would consider using BSON format for the messages over the wire. For a truly immersive experience, smart environment must be able to take advantage of the robots' closed world knowledge as a starting point. Future work will explore using FGMM for pattern recognition leveraging on previous work by the authors [24, 25].

Acknowledgments. The authors would like to acknowledge support from DREAM project of EU FP7-ICT (grant no. 611391), Research Project of State Key Laboratory of Mechanical System and Vibration China (grant no. MSV201508), and National Natural Science Foundation of China (grant no. 51575412).

References

1. Ahmad, A., et al. (2013). Cooperative robot localization and target tracking based on least squares minimization. Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE.
2. Aldebaran (2016). "NAO Documentation." Retrieved 09/03, 2016, from http://doc.aldebaran.com/.
3. Babu, A. K. A. and R. Sivakumar (2015). Development of Type 2 Fuzzy Rough Ontology-based Middleware for Context Processing in Ambient Smart Environment. Intelligent Computing and Applications, Springer: 137-143.
4. Billard, A. and D. Grollman (2013). "Robot learning by demonstration." Scholarpedia 8(12): 3824.
5. Cheng, L., et al. (2012). Design and implementation of human-robot interactive demonstration system based on Kinect. Control and Decision Conference (CCDC), 2012 24th Chinese, IEEE.

6. Corporation, i. (2014). ”MQ - The Guide - MQ - The Guide.” Retrieved 03/04, 2016, from http://zguide.zeromq.org/.
7. Developer, G. (2015). Google I/O 2015 - A little badass. Beautiful. Tech and human. Work and love. ATAP.
8. Google (2016). ”Google’s Project Tango.” Retrieved 04/04, 2016, from https://www.google.com/intl/en_us/atap/project-tango/.
9. Hopfield, J. J. (2007). ”Hopfield network.” Scholarpedia 2(5): 1977.
10. JSON-LD (2016). ”JSON for Linking Data.” Retrieved 03/04, 2016, from http://json-ld.org/.
11. Kim, H., et al. (2005). CAMUS: A middleware supporting context-aware services for network-based robots. Advanced Robotics and its Social Impacts, 2005. IEEE Workshop on, IEEE.
12. Kohlbrecher, S., et al. (2015). ”Humanrobot Teaming for Rescue Missions: Team ViGIR’s Approach to the 2013 DARPA Robotics Challenge Trials.” Journal of Field Robotics 32(3): 352-377.
13. Li, R., et al. (2013). Towards ros based multi-robot architecture for ambient assisted living. Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on, IEEE.
14. Li, X., et al. (2015). ”Context Aware Middleware Architectures: Survey and Challenges.” Sensors (Basel) 15(8): 20570-20607.
15. Lin, B.-S. P., et al. (2016). ”The Roles of 5G Mobile Broadband in the Development of IoT, Big Data, Cloud and SDN.” Communications and Network 8(01): 9.
16. Murray, D., et al. (2010). An experimental comparison of routing protocols in multi hop ad hoc networks. Telecommunication Networks and Applications Conference (ATNAC), 2010 Australasian, IEEE.
17. Newmeyer, N. (2015). ”Changing the Future of Cyber-Situational Awareness.” Warfare 14: 32-41.
18. Opennetworking.org (2016). ”Software-Defined Networking (SDN) Definition - Open Networking Foundation.” Retrieved 01/04, 2016, from https://www.opennetworking.org.
19. Perera, C., et al. (2014). ”Context Aware Computing for The Internet of Things: A Survey.” Ieee Communications Surveys and Tutorials 16(1): 414-454.
20. Research, B. (2011). M2M world of connected services: Internet of Things.
21. Rutishauser, S., et al. (2009). ”Collaborative coverage using a swarm of networked miniature robots.” Robotics and Autonomous Systems 57(5): 517-525.
22. Stachniss, C. (2014). Introduction to Robot Mapping. SLAM-Course.
23. Thrun, S., et al. (2005). Probabilistic robotics, MIT press.
24. Ju, Z. and Liu, H.Fuzzy Gaussian Mixture Models, Pattern Recognition 2012, 45(3):1146-115.
25. Ju, Z., Li, J. and Liu, H. (In Press) An Integrative Framework for Human Hand Gesture Segmentation in RGB-D Data IEEE Systems Journal, DOI: 10.1109/JSYST.2015.2468231