Constructing Context-centric Data Objects to Enhance Logical Associations for IoT Entities

Bin Xiao*, Theo Kanter and Rahim Rahmani

Department of Computer and System Sciences, Stockholm University, Kista, SE-202 70, Sweden;
{xbin,kanter,rahim}@dsv.su.se.

Abstract

Entities in Internet of Things (IoT) need intelligent associations to allow a flexible and dynamic system reaction towards varying user situations. Purely data-oriented association methods (e.g., data-mining, machine-learning, etc.) are limited in deduction from existing associations. Also such data-oriented methods are limited in accuracy for working with small-scale datasets (e.g., working with patterns retrieved from historical data), outputting associations based on statistics rather than logic. Moreover, existing semantic technologies (ontological-oriented or rule-oriented) are facing with either flexibility or dynamicity challenges to discover and maintain the associations. This paper proposes an alternative technique of semantically constructing context-centric data objects based on service logics for logical associations, which enables an event net based on association nets adapting for the changing situations (called context-centric). A proof-of-concept implementation is carried out based on a vehicle planning scenario to validate the data construction technique. Comparing to previous work, this technique possesses advantages of flexibility and dynamicity for entity associations based on service logics.

1. Introduction

Entity associations in Internet of Things (IoT) don’t just mean to connect heterogeneous sensors for exchanging and sharing data. Instead, the entities associations should enable to perceive the user situations according to system service logics, called logical associations. User situation indicates the concurrent context a system participant locates in, reflecting the user wants. System service logics directly define the problems and services that the system should
care and deliver. The logical associations refer to both virtual and physical entities (indicating the entity has has no physical presence, such as human body vs. electronic bill) in IoT system. Data objects generated by those entities should be constructed with resolvable semantics to present user situations, thus to enable the talking between heterogeneous entities. We can generally categorize semantic context as text-based and situation-based. Text-based semantics enables text content become machine understandable, while situation-based semantics provides syntax for machine/system to understand user context and then to react pertinently. Nevertheless, the same situation is understood differently in different semantics domains, since semantic domain representing system logic defines the syntax and rules for comprehending user situation. The associations are dynamic and flexible within its semantic domain. “Dynamic” indicates that the associations can be updated by evolving through the changing situation. “Flexible” indicates decreasing coupling degree between data objects to make it easy for establishing a new association with the minimum cost. Therefore, we need a technique to store/retrieve the situation semantics into/from data objects, and make them expressive and comprehensible in support for creating and maintaining the dynamic and flexible associations between virtual and physical entities. This paper explores a context-centric data constructing technique to enable discover and establish the dynamic and flexible associations based on user situations and service logics, which can be utilized in many fields, such as elderly care and vehicle planning. A vehicle planning scenario is used for verification.

2. Background and Related Work

Available researches regarding logical associations for heterogeneous IoT entities have two general directions to create and maintain the associations: Data-oriented and Logic-oriented. Data-oriented approaches such as data mining and machine learning create associations based on similarity. The data-oriented techniques utilize historical data to explore patterns and train the system, but have following limits: 1) the discovered potential associations have more sense in mathematic rather than logic; 2) they challenge the small-scale system that cannot provide large enough dataset for processing. Since data-oriented approach establishes associations by analyzing data sample, accuracy are influenced if the data sample amount is small; 3) data-oriented approaches discover unrealized patterns and associations from dataset, but cannot do logic deducing for associations, with lack of data construction for deducing. Moreover, in many cases new associations are expected to be discovered based on existing known associations, which is not applicable by the data-oriented technique. Logic-oriented approach includes generally two directions: the ontology-oriented and the rule-oriented. The ontology-oriented techniques emphasize on association reasoning based on semantic annotated data, while the rule-oriented techniques care more on system behavior by rule matching. However, those methods have their own limits respectively. Rule-oriented technology relies on “if…then...” pattern to drive system logical behaviors when user situation fulfills with the trigger conditions by checking and matching the Right Hand Side (RHS) with Left Hand Side (LHS), with two vital defects hindering the flexibility and dynamicity. 1) the LHS part is not expressive enough to explore potential associations between entities. Its knowledge base focus more on the terminologies and assertions, but the relations between those terminologies and assertions are not described enough; 2) it is a heavy task to describe a complex context by creating many rules. Moreover, many existing associations have already presented by common knowledge that does not need to explore again. Therefore, we just need a deducing mechanism to find and discover those associations. For example, there are three concepts (virtual entities): Human, Male, Married Man. If Male is included in Human, then the Married Man should be included in Human since the subsumption relations between Male and Married Man. In such case, there is no need to redundantly define three different individual rules to describe the associations. Besides the rule-oriented association, another technique track is the ontological approach which associates groups of concepts containing instances and relations between instances. Ontology presents semantics through concepts and the properties between the concepts. Also ontology provides reasoning (such as the tableau algorithms) to explore logic associations among the successor nodes based on the associations among their ancestor nodes, according to the inheritance and subsumption relations between ancestor nodes and descendant nodes. However, this approach still lacks mechanism to dynamically evolve the associations when the threshold is triggered in Assertion Box. Ontology based algorithm for discovering associations lacks a trigger mechanism to establish dynamic associations. It sometimes makes user situations ignored by the system since missing explicit system responding reactions. Towards those defects, a technique to discover and establish the associations through perceiving user
situation based on service logics is expected. Therefore, we are motivated to explore the approach to construct the data objects and enable them with semantic schema for establishing the flexible and dynamic entity associations.

3. Constructing Context-centric Data Objects

Data objects of virtual and physical entities are constructed in the same schema using ontology & rule hybrid semantic structure with multiple adaptable and extendable parameters to ensure dynamicity and flexible. As shown in Fig.1 (left), the context-centric construction schema consists of four intuitive parameters: Terminology, Assertion, Role, and Rule. Each intuitive parameter is annotated with descriptive properties like Data Property, Rule Property, and Universal Recourse Identifier (URI). Intuitive parameters constructively connect with each other. In context-centric data construction, data objects from the entities are ontologically separated into Assertion box (A-box) and Terminology box (T-box). Terminologies are concepts defining the framework of entities. Assertions make the associations adhere to event net established upon the associated entities to ensure logic consistency, as shown in Fig.2 (left). It is resource-consuming to main the associations if associations are defined as either terminologies or assertions, as in such case any modification and deleting has to change ontological relations between all the terminology and assertions. To ensure flexibility and dynamicity, an association is defined as a reactive result of entities mutually attracting and integrating based on their characteristics in the surroundings. The parameter “role” is introduced to describe the characteristic. Hence we can make association process to emulate the chemical molecular combinations process, where an association is decided by roles and rules. An organized role parameter in each context-centric entity works like an individual chemical key. Each entity has its own role parameter, like each molecular has its own chemical key. Rules describe the laws that roles need to obey for association: 1) what roles of the entity can associate mutually attract entities; 2) limits for the association between those entities. Rules present and depict system logics using the schema shown in Fig.1 (right).

Each role has five inner properties: Uniform Identifier (UI), Verb Label (VL), Subject Label (SL), Object Label (OL) and Data Property Label (DL). URI uniquely identifies each role and entity. The Verb Label is a behavior marker for roles, where different roles might have the same VL. The Subject and Object property form a pair key to annotate the positive and negative characteristics performed by roles. Data properties annotate roles by exogenous data, like time stamp. With such construction, associations are constructed using a triple <verb, subject, object>. The constructed data objects accept three kinds of rule parameters, as Fig.2 (right) shows. 1) Rule type I: define primitive service logics among terminologies in T-box (blue lines). 2) Rule type II: define triggers between assertions in A-box (red lines) to set up event net. 3) Ontology type: support ontology reasoning to explore potential terminology relations from ancestors to successors, based on constructed data objects (from the blue through the yellow to the red). Each entity in cyber world is a mirror for the entity in physical world. Hence, to discover and create the associations between entities is to explore the associations between assertions, where associations are instantiated cases of terminology.

Table I. Rule Syntax for Service Logics

| Rule Syntax for Service Logics | Rule Syntax for Service Logics |
|-------------------------------|-------------------------------|
| If $\exists$ terminology $a$ with role $<1,0>$ & $\exists$ terminology $b$ with a role $<0,1>$, $\Rightarrow$ associate $a$ to $b$ | If $\exists$ terminology $a$ with role $<1,1>$ & $\exists$ terminology $b$ with a role $<0,1>$, $\Rightarrow$ associate $a$ to $b$ |
| “If $\exists$ terminology $a$ with role $<1,1>$ & $\exists$ terminology $b$ with a role $<1,0>$, $\Rightarrow$ associate $b$ to $a$.” | “If $\exists$ terminology $a$ with role $<0,0>$, $\Rightarrow$ dis-associate $a$ from any terminology $b$ with role.” |
This constructing approach maps virtual and physical entities into the cyber world with an event net above entities’ logical associations. And data objects from the physical world are timely updated into the cyber world in support of exploring new associations. Event net is an integrated entity cluster triggered by surroundings, so system can immediately detect user situations and respond, making associations dynamic. An event net is set upon associations based on service logic to avoid logical conflict (as the green layer shown in Fig.2 (left)) using rule based deducing. The green lines indicate service logics, and the red lines indicate the associations. The entity associations formed in cyber world are described by two attributes: Range and Depth, as shown in Fig.2 (right). Range refers to the associations’ scope, indicating the terminologies included in the cyber world. Range shows categorizes of the most preliminary associations starting from the root terminologies in ontology. In the depth and range ontology model, all the successors must obey the associations happened among their ancestors. The association depth shows how many hierarchies are there in the ontology model. At each hierarchy tree leaf, assertions are instantiated from the terminology. Arrows imply the association deducing direction.

4. Establish Associations Based on the Construction

Based on the constructed data object, entities are associated by three processes, as the three white boxes shown in Fig.3 (left). Firstly, service logics are presented using rule type I. The service logics are primitive system configurations to comprehend user situation for system reactions. Secondly, construct context-centric data objects and deduce associations in T-box, where all behaviour of entity obeys service logics. Finally, set up associations in A-box with event net using rule type II, where assertions are instantiated from terminologies. New associations are explored based on existing associations in T-box and evolving through user situation. Rule type II states about implementing RHS to overwrite assertions’ roles inherited from their ancestor terminologies, and then set up event net. The first process defines service logics using structural rules in type I. As shown in Fig.3 (right), each terminology is attached with roles and data properties, where a role is functional annotation for terminology. A role is formed by matching the \(<subject, object>\) pair key and an association is defined as \(<role(a,b), role(c,d)>\), where \(a,b,c,d\) is either 0 or 1. Also associations are described by data property in \(<role(a,b)\{x,y,z\}, role(c,d)\{u,v,w\}\>. Any array such as \(<2, 3>\) can be defined in data property with role \(Take\_pill\ (1,0)\), where the label means “Be taken as pills, 2 times per day and 3 pills each time”. Terminology with the same role in opposite pair key value will be associated automatically. Hence the service logics operate association by handling the roles. There are basically four optional types to define the service logics, as shown in Table I. As discussed, role attached to each terminology is like a chemistry key to a molecular, where entities are associated according to the role matching result. Role base and rules provide interface to update and maintain associations. So associations can be easily added, changed, and deleted by changing the pair keys of their role parameter to avoid large modifications, making associations dynamic and flexible. The process “Deduce Associations Based on Context-centric Construction” deduces and discovers the
associations between terminologies then establishes associations among assertions. Association deducing and discovering are based on Deductive Algorithm (DA)\(^\dagger\). For the terminologies in T-box, each successor node is inserted into the domain of its ancestor association, where an association hierarchy in each domain forms a tree of information subdivision with root node and leaf node. The DA deduces associations starting from the ancestor to the successor. Thus leaf nodes can be associated based on the deducing to their terminologies (their ancestors). Such association deducing goes through each leaf of the tree (A-Box), which may introduce redundant associations that should not exist. The redundant associations will be dealt with in the next process and to set upon the event net with rule type II. In the process “Adjust the Associations and Establish Associations in A-box”, assertions overwrite and inherit the associations from terminologies to ensure the dynamicity and flexibility. Assertions do overwriting by manipulating <verb, subject, object> and data property. All data generated by sensors are updated timely to instantiate assertions, where a rule property is instantiated to present service logics through the updated role value and data property. So the associations are updated automatically, as long as the data objects describing user situation are updated (role, data property, etc.). In A-box, an event net is established upon the associations to keep the logic correctness and consistency. If any entity tries to trigger the reaction by another disassociated entity, logic conflicts may happen. Fig.4 describes the correct event net and the incorrect event net. Event net should obey the associations to guarantee the logic consistency, as shown in Fig.4. The associations enable to avoid any illogical triggering event call by setting up trigger paths between entities. The yellow nodes in Fig.4 show the unexpected event call that should not exist in event nets. The established associations help to avoid such event mistakes. Two associations are associated when role of one assertion includes matching role value with another entity in the triple <verb, subject, object>. To update the association is to overwrite the <verb, subject, object>, and role triple is the only interface that can be accessed to update the association. Further, value of data properties in each entity can also trigger the overwriting where rules are defined for each entity.

5. Verify the Approach through Vehicle Planning

We use an algorithm called Deductive Algorithm\(^\dagger\) to deduce the assertions’ associations based on terminology associations, where rule type I associations are established between terminologies with the rules of type II, then the ontology reasoning are implemented between assertions. The Deductive Algorithm is an ontology based algorithm to deduce and discover the associations from the root nodes to the leaf nodes based on the service logics presented by constructed data schema. For proof-of-concept validation, we apply this technique to a simplified scenario of vehicle planning. To avoid the traffic congestion, vehicles need real-time planning suggestions to deal with the changing traffic situation by associating heterogeneous vehicle entities and set up an event net to synchronize driver situations, which enables us to integrate the dispersed data resources generated by each entity. For simplified proof-of-concept, we extract three entities from the scenario: local status, city map, and vehicle. Terminologies and assertions with their roles are defined in Fig.5 (left). Rules in type II are defined with service logics syntax to synchronize and update entity associations according to traffic situation, such as “If Street A.streetname != Situation
B.streetname, then remove <A.role (a,b), B.role(c,d)>", "If Street A.region != "Solna", then A.role (a,b)=(0,0)."", and "If vehicle a.speed = = 0, then a.role(a,b) = role (0,0)". The vehicle constructed data objects and service logics help to create evolving vehicle entity associations then evolve the associations through traffic situations. Protégé 3.4.1 with Jess Tab \textsuperscript{12} is used in our proof-of-concept implementation, which is a simulation environment for ontology engineering embedded with a rule based reasoning engine. With the hypothetical data object input, we get the brief planning message for the vehicles in Fig.5 (right), with the event net.

6. Discussion and Conclusion

The Internet-of-Things requires flexible and dynamic associations between entities. Existing approaches have mainly focused on data-centric methods, such as machine learning and data mining, rule-oriented or ontology-oriented. Both classes of methods are limited in flexibility and dynamicity, the former as these methods are statistical, whereas the latter are static. In response to these observations, this paper proposed a context-centric data construction technique to support associations based on service logic. Specifications of such associations entail both the associations and events between entities. Context-centric data constructing approach provide flexible and dynamic methods to deduce and maintain the associations by manipulating the roles and data property. Through the overlaying of associations with event nets, the system can smartly react to situations as long as a threshold is triggered. In proof of concept simulation, a smart vehicle scenario is used to demonstrate how the context-centric constructing approach can be used, showing the validity of the approach. Thus, our approach circumvents the limitations of pre-existing working and supports dynamic and flexible association. Our research is currently focusing on developing a platform and tools to support our approach. We are refining our methods to automatically achieve a self-configurable system, where associations and data can be adapted by self-learning.

Acknowledgements

We deeply acknowledge funding support from the EU AAL SALIG++ Project.

References

1. B. Xiao, T. Kanter, and R. Rahmani, “Logically Associate the Virtual and Physical Entities of IoT in Support of Ambient Intelligence Services,” unpublished.
2. P. Barnaghi, W. E. I. Wang, C. Henson, and K. Taylor, “Semantics for the Internet of Things: early progress and back to the future,” Int. J. Semant. Web Inf. Syst., vol. 8, no. 1, pp. 1–21, 2012.
3. B. Xiao, T. Kanter, and R. Rahmani, “Generic Distributed Sensing in Support of Context Awareness in Ambient Assisted Living,” in Lecture Notes in Electrical Engineering, 2014, vol. 308, pp. 99–107.
4. O. Maimon and L. Rokach, Eds., “relational data mining,” in Data Mining and Knowledge Discovery Handbook, Boston, MA: Springer US, 2010, pp. 887–911.
5. W.-Y. Chiang, “To mine association rules of customer values via a data mining procedure with improved model: An empirical case study,” Expert Syst. Appl., vol. 38, no. 3, pp. 1716–1722, Mar. 2011.
6. S. De, B. Christophe, and K. Moessner, “Semantic enablers for dynamic digital–physical object associations in a federated node architecture for the Internet of Things,” Ad Hoc Networks, vol. 18, pp. 102–120, Jul. 2014.
7. A. Sheth, C. Henson, and S. S. Sahoo, “Semantic Sensor Web,” IEEE Internet Comput., vol. 12, no. 4, pp. 78–83, Jul. 2008.
8. M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, “The SSN ontology of the W3C semantic sensor network incubator group,” Web Semant. Sci. Serv. Agents World Wide Web, vol. 17, no. June, pp. 25–32, Dec. 2012.
9. E. Klien, “A Rule-Based Strategy for the Semantic Annotation of Geodata,” Trans. GIS, vol. 11, no. 3, pp. 437–452, Jun. 2007.
10. A. Turhan, “Description Logic reasoning for Semantic web ontologies,” in Proceedings of the International Conference on Web Intelligence, Mining and Semantics, 2011, pp. 1–5.
11. J. Bao, D. Caragea, and V. Honavar, “A Tableau-based Federated Reasoning Algorithm for Modular Ontologies,” in IEEE/WIC/ACM International Conference on web intelligence, 2006, pp. 404 – 410.
12. H. Eriksson, “Using JessTab to Integrate Protégé and Jess,” Intell. Syst. IEEE, vol. 18, no. 2, pp. 43–50, 2003.