Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2016) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

The 3rd International Workshop on Internet of Things: Networking, Applications and Technologies (IoTNAT 2017)

# Recommendation Technologies for IoT Edge Devices

Alexander Felfernig*[a], Seda Polat Erdeniz*[a], Michael Jeran[a], Arda Akcay[a], Paolo Azzoni[b], Matteo Maiero[b], Charalampos Doukas[c]

[a]*TU Graz - Software Technology Institute, Graz, Austria*
[b]*Eurotech Group - EthLab, Trento, Italy*
[c]*FBK, Italy*

**Abstract**

The AGILE project (agile-iot.eu) aims to create Internet of Things (IoT) gateway technologies that support many devices, protocols, and corresponding management and development activities. In the context of this project there are scenarios that require the support of recommendation technologies. The major goal of this paper is to provide an overview of recommendation approaches and to discuss their relevance for AGILE.


*Keywords:* Recommender Systems; Internet of Things; Artificial Intelligence

## 1. Introduction

The AGILE project aims to develop the software and hardware required to build modular gateway solutions for managing devices and data in Internet of Things (IoT) scenarios [1]. It will support the local management of devices and data, app development, and include security features that allow users to share data in a trusted way. AGILE technologies are based on existing open source projects.

Recommendation technologies are progressively attracting the interest of new application domains as a valuable solution to increase system autonomy and efficiency. Modern embedded systems involved in IoT scenarios can exploit the advantages offered by context-aware and profile-driven recommendations. One important focus of AGILE is the development of recommendation technologies that help to personalize the interaction with IoT environments. End-users of AGILE infrastructures will be supported in retrieving apps and additional devices useful in their application context. App developers will be supported in the definition of new apps by receiving recommendations regarding the definition and extension of application workflows.

Recommender systems [2,3,4,5] suggest items (alternatives, solutions) that are of potential interest for a user. Examples of related questions are: *which book should I purchase?*, *which test method should I apply*?, *which method calls*

* Corresponding authors.
*E-mail address:* (afelfern, spolater)@ist.tugraz.at

1877-0509 

*are useful in a certain development context?* or *which apps are of potential interest for the current user?* In the IoT (Internet of Things) context, recommendation functionalities are required, for example, in the context of IoT workflow development, the recommendation of apps, and domain-specific scenarios such as food recommendation[6], personalized shopping[7], and technology fairs[8]. In this paper we provide an overview of recommendation approaches and discuss relevant IoT related scenarios.

There are different types of recommendation approaches – which recommendation approach to choose depends on the application context[5]. *Collaborative filtering*[9] is based on the idea of word-of-mouth promotion, i.e., the opinion of users with similar preferences plays a major role in a decision. These users are also denoted as *nearest neighbors*, i.e., users with similar preferences compared to the current user. The first step of a collaborative filtering recommender is to identify the *k-nearest neighbors*[1] and to extrapolate from the ratings of these users the preferences of the current user.

*Content-based filtering*[10] is based on the assumption of monotonic personal interests. For example, users interested in the topic *Operating Systems* are typically not changing their interest profile from one day to another but will also be interested in the topic in the (near) future. In online scenarios, content-based recommendation approaches are applied, for example, when it comes to the recommendation of websites (e.g., news items with a similar content compared to the set of already consumed news). The basic approach of content-based filtering is to compare the content of already consumed items (e.g., a list of news articles) with new items that can potentially be recommended to the user, i.e., to find items that are similar to those already consumed (and positively rated) by the user.

*Knowledge-based recommendation*[11,3] does not primarily rely on item ratings and textual item descriptions but on deep knowledge about the offered items. Such deep knowledge (semantic knowledge) describes an item in more detail and thus allows for a different recommendation approach. The current user articulates his/her requirements in terms of item property specifications which are internally as well represented in terms of rules (constraints). Constraints are interpreted and the resulting items are presented to the user. Note that items can also be interpreted as cases which are recommended to the current user as solutions for his/her current requirements (problem setting) – the underlying recommendation process is also denoted as *case-based recommendation* (which is interpreted as a kind of knowledge-based recommendation approach in the recommender systems research community)[3].

*Utility-based recommendation*[3] is based on the idea that – given a set of items – item ranking is determined on the basis multi-attribute utility theory (MAUT)[12]. In this case, each item is evaluated with regard to a set of interest dimensions (e.g., digital cameras could be evaluated with regard to the interest dimensions *economy*, output *quality*, and equipment *durability*). Utility-based recommendation is often combined with knowledge-based recommendation approaches since item ranking is needed after constraints (rules) have pre-selected the set of relevant items.

The contributions of this paper are the following. First, we provide an overview of applications of recommendation technologies in IoT scenarios. Second, we sketch AGILE-related application scenarios of recommendation technologies which go beyond existing developments. The remainder of this paper is organized as follows. In Section 2 we analyze existing applications of recommendation technologies in IoT contexts. Thereafter – in Section 3 we discuss and exemplify envisioned applications of recommendation technologies in AGILE. In Section 4 we discuss issues for future work and conclude the paper.

## 2. Related Work

There already exist a couple of contributions focusing on the application of recommendation technologies in the Internet Of Things (IoT) context. In the following we discuss existing related work and also point out similarities and differences compared to the research goals in the AGILE project.

Compared to other recommendation scenarios, IoT-based applications enable a deeper understanding of user preferences and behaviors which can primarily be explained by the availability of heterogeneous information sources[13,14]. *Personalized shopping*, for example, is a core element of IoT technology based retail environments[7]. Customers entering a store receive recommendations regarding items and corresponding price offers – these recommendations depend a.o. on the condition of the offered items. For example, if the durability of some of the offered items decreases, corresponding special offers could be announced to the customer. Important IoT related aspects are *automated quality control of items*, *context-dependent pricing*, and *targeted product information*. The recommendation approach pre-

[1] *k* represents the number of users with similar ratings compared to the current user.

Table 1. Collaborative filtering based recommendation of apps based on gateway profile information. Apps *a2* and *a3* can be recommended since they are installed on gateways with devices also connected to the local gateway.

| profile | d1 | d2 | d3 | d4 | a1 | a2 | a3 | a4 |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.0 | | 1.0 | | | | 1.0 | |
| 2 | | 1.0 | | | 1.0 | | 1.0 | |
| 3 | 1.0 | | 1.0 | | | 1.0 | 1.0 | |
| local | 1.0 | | 1.0 | | 1.0 | | | |

sented in[7] follows a knowledge-based (rule-based) approach – such rules can be generated on the basis of techniques such as sequence mining, i.e., what will be the next items a customer is interested in.

Valtolina et al.[6] introduce a *household scenario* where users ask for recommendations regarding recipes. In the context of an IoT infrastructure, a recommender system does not have to only rely on the preferences of the user but can take into account further information sources. The availability of orthogonal data sources provided by different IoT devices will help to increase the prediction quality of recommendation algorithms, however, visualization techniques and persuasive interfaces are important additional means to make recommendations acceptable for users[5].

Munoz-Organero et al.[8] introduce *technology fairs* as a scenario where context-aware recommender systems can be applied. In such a scenario, users can receive information about exhibits of relevance and also be informed about lectures to attend depending on their personal preferences.

Similar to knowledge-based recommender systems[15,3], knowledge-based configuration[16] is a process in which users specify their requirements and the configuration system (often denoted as *configurator*) provides feedback.

## 3. Recommendation Scenarios in AGILE

There are different scenarios where recommendation technologies play a role in AGILE. For example, a user installs an AGILE gateway to make his/her home smarter. The user is already using temperature and pressure sensors and temperature and pressure alarm applications. He/she buys a gas sensor, plugs it onto the gateway then opens the management user interface of the gateway (see Figure **??**) in the web browser. The user wants to receive some app recommendations according to the overall setting on the gateway. He or she presses the *get recommendation* button on the management user interface. On the basis of the gateway profile, the AGILE recommender applies collaborative filtering on the profiles knowledge base. The recommender returns the recommended applications for the gateway. The user selects the *fire alarm app* from the recommendation list and installs the app. He/she also wants to implement an own smart home application on the AGILE gateway. Therefore the user starts to build a workflow with the AGILE development user interface by adding a temperature node which collects data from the installed temperature sensor. The user then activates the recommender just to figure out how to best extend the current workflow. On the basis of the current workflow content (the temperature node), a collaborative filtering recommender can recommend possible extensions (e.g., a pressure node).

*Collaborative Filtering based Recommendation of Apps*. On the basis of a given local gateway configuration (e.g., devices and drivers), a recommender can provide a list of (additional) apps that can be of relevance. Collaborative filtering can determine such recommendations on the basis of gateway profile data collected in anonymous form (see, e.g., Table 1). For example, on the basis of the installed devices and drivers, a recommender could propose additional apps that can operate on the installed set of devices. Furthermore, a recommender could propose extensions (e.g., additional devices) that are needed to support apps often installed/used in similar gateway configurations.

A simplified example of a related recommendation approach is given in Table 1.[2] In this context, information about installation bases (profiles) from other gateways is used to infer relevant apps to be additionally proposed/installed on the *local* gateway. Devices *d1* and *d3* are connected to the local gateway. The installation bases 1 and 3 (profiles) include all the devices connected to the local gateway but include additional apps that are currently not installed on the local gateway (these are the apps *a2* and *a3*). Consequently, these apps represent recommendation candidates for the local gateway.

There are a couple of similarity metrics used in the context of collaborative filtering scenarios for determining nearest neighbors – for details we refer to[5]. For the purposes of our examples, we introduce a simplified formula that

[2] Note that for simplicity we base our examples on abstract representations for apps ($app$), devices ($d$), and workflows ($wf$).

Table 2. Collaborative filtering based recommendation of apps based on user ratings. User 1 is the nearest neighbor of the current user. *App7* has been purchased by user 1 but has not been purchased by the current user and is therefore recommended.

| user | app1 | app2 | app3 | app4 | app5 | app6 | app7 | app8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.0 | | 4.5 | | | | 3.0 | |
| 2 | | 2.0 | | | 4.5 | | 2.5 | 3.5 |
| 3 | 4.0 | | 3.0 | | | 2.5 | 3.0 | |
| current | 1.0 | | 4.0 | | | | | |

Table 3. Content-based recommendation of apps. App *2* it has the highest similarity to the local gateway profile information provided by the user.

| app | d1 | d2 | d3 | d4 | d5 | d6 |
|---|---|---|---|---|---|---|
| 1 | | x | x | | x | x |
| 2 | x | | x | | x | x |
| 3 | x | | | | | |
| local | x | | x | | | |

supports the identification of *k-nearest neighbors*[3] (see Formula 1). If applied to the example of Table 1, *profiles* have to be associated with the *items* contained in Formula 1.

$$similarity(item_a, item_b) = \frac{1}{1 + \Sigma_{i=1}^{n} |eval(item_a) - eval(item_b)|} \tag{1}$$

Alternatively, collaborative filtering can exploit the ratings of users when interacting with an app marketplace (in this context, *users* have to be associated with the *items* contained in Formula 1). The underlying idea is that AGILE gateways can be connected to app marketplaces where users can select and download apps that are of interest for their local gateway installation. In this scenario, the evaluation data (ratings) of users serve as a basis for determining recommendations (see Table 2): user 1 (the *nearest neighbor*) has provided app evaluations which are similar to those of the current user. Consequently, a collaborative recommender proposes apps to the current user which have been investigated by the nearest neighbor but not by the current user (e.g., *app7*).

*Content-based Recommendation of Apps*. Another simple alternative for recommending apps is to implement a content-based recommendation approach where apps can be recommended for installation if their required devices (it is assumed that this information is given for each app) are "compatible" with the local gateway configuration (profile). Please also note that it is possible to support scenarios where apps are recommended that do require additional hardware/device driver components in order to work properly.

When applying a content-based filtering based approach, recommended items are determined on the basis of the similarity of the local gateway profile information (e.g., in terms of installed devices) and the profile information of apps available, for example, on a marketplace in the cloud. Similar to collaborative filtering, there are different types of similarity metrics (see, e.g.,[5]). For the purposes of our examples, we introduce a simplified formula that supports the identification of, for example, relevant apps for the local gateway (see Formula 2).

$$similarity(profile, app) = \frac{devices(profile) \cap devices(app)}{devices(profile)} \cup devices(app) \tag{2}$$

Formula 2 determines the similarity on the basis of the information about installed devices. However, this approach can be extended to include further information, for example, regarding installed modules and network protocols available on the gateway. In our example of Table 3, *app 2* has the highest similarity with the local profile information, therefore this app is recommended to the user. Note that content-based recommendation is often applied when the similarity between textual information from different sources has to be determined. Therefore, the approach presented in this paper can be extended to the matching of text-based search criteria (a users searches for an app) and the textual description of apps.

*Utility-based Recommendation.* Modern embedded systems included in IoT scenarios support a rich set of connectivity solutions (e.g., 3G, LTE, TD-LTE, FDD-LTD, WIMAX, and Lora). In this context, recommendation technologies play an important role suggesting the best connectivity configurations for the selected communication channel.

[3] For simplicity we assume $k = 1$.

Table 4. Utility table as basis for utility-based recommendation.

| protocol | performance | reliability | costs |
|---|---|---|---|
| a | 9 | 5 | 2 |
| b | 5 | 8 | 3 |

Table 5. Example user preferences w.r.t. interest dimensions *performance*, *reliability*, and *costs*.

| user | performance | reliability | costs |
|---|---|---|---|
| $u_1$ | 10 | 3 | 1 |
| $u_2$ | 5 | 7 | 10 |

Table 6. Collaborative filtering based recommendation of nodes (partial workflows). Workflows *2* and *3* include the nodes also contained in the local (partial) workflow. Nodes not yet contained in the local workflow (*node5* and *node6*) can be recommended.

| workflow | node1 | node2 | node3 | node4 | node5 | node6 |
|---|---|---|---|---|---|---|
| 1 | | 1.0 | 1.0 | | 1.0 | 1.0 |
| 2 | 1.0 | | 1.0 | | | 1.0 |
| 3 | 1.0 | | 1.0 | | 1.0 | 1.0 |
| local | 1.0 | | 1.0 | | | |

The recommendation can be based, for example, on location information, available connectivity, performance and reliability requirements, and contractual aspects and costs. In AGILE, gateway configurations can be manually defined by users but also be determined on the basis of a configurator that is in charge of keeping the overall system installations consistent. A configurator (e.g., a constraint solver) can determine alternative configurations which have to be ranked. In order to determine a ranking for alternative configurations, a MAUT-based approach can be used. Examples of evaluation *dimensions* (dim) used in MAUT could be *performance*, *reliability*, and *costs*. Depending on the current gateway configuration and the usage context, a configurator can determine alternative re-configurations and rank them accordingly.

An example of the application of a utility-based approach is the following. Table 4 includes an evaluation of connectivity protocols (*a* and *b*) to be used on the gateway, for example, for different types of data exchange. Furthermore, Table 5 includes the personal preferences of two different gateway users ($u_1$ and $u_2$).

$$utility(protocol, user) = \Sigma_{d \in dim} interest(user, d) \times value(protocol, user) \tag{3}$$

In order to determine the protocol that should be chosen for a specific user, we can apply a utility function (see, e.g., Formula 3). In this example, protocol *a* has a higher utility for user $u_1$ (107.0) whereas protocol *b* has a higher utility for user $u_2$ (111.0).

A workflow consists of a set of interconnected nodes that reflect certain modular functionalities to be executed, for example, reading the data from a temperature measurement device or measuring the heart rate of a person.

Recommendation algorithms can also be applied to identify / predict functions or workflow nodes that will be useful in the current development context. On the basis of the information of already existing app workflows, related recommendations can be determined, for example, on the basis of collaborative filtering. Table 6 includes an example of applying collaborative filtering for the recommendation of workflow nodes to be additionally used in the current development context.

Alternatively, if we assume the existence of workflow ratings in a workflow repository, complete workflows can be recommended to the current user (see, e.g., Table 7). Users 1–3 have already rated the workflows $wf1 - wf6$ in the workflow repository. These ratings can be exploited by collaborative filtering to determine recommendations for the current user. For this user, a nearest neighbor has to be identified (e.g., on the basis of Formula 1) and workflows evaluated positively by the nearest neighbor but have not been evaluated by the current user are recommended.

Recommendation functionalities are typically triggered from two different sources. First, the AGILE development user interface supports the recommendation of workflows in situations where users are building their own applications (apps). Second, when using the gateway as an end-user, AGILE can provide recommendations in terms of new apps and also additional equipment needed to extend the current gateway infrastructure. Recommendation functionalitities are located on an AGILE server outside the gateway for two reasons: (a) to keep recommendation algorithms scaleable and efficient and (b) to be able to easily exploit profile information from other gateways (in an anonymous fashion) for recommendation purposes.

Table 7. Collaborative filtering based recommendation of workflows. User *3* is the nearest neighbor of the current user due to the most similar rating behavior. Workflow $wf5$ can be recommended to the current user since it has been evaluated positively (4.0) by the nearest neighbor.

| user | wf1 | wf2 | wf3 | wf4 | wf5 | wf6 |
|---|---|---|---|---|---|---|
| 1 | | 2.0 | 4.0 | | 4.0 | 4.5 |
| 2 | 1.0 | | 3.0 | | | 3.5 |
| 3 | 3.0 | | 4.0 | | 4.0 | 2.0 |
| current | 3.0 | | 4.5 | | | |

## 4. Future Work and Conclusions

In all of the mentioned scenarios, recommendation algorithms are in the need of additional information outside the local gateway for determining a recommendation. If a collaborative approach is used, installation data from other gateways is needed to infer possible additional apps for the current gateway. In the case of content-based recommendation, information about needed devices and related drivers has to be made available. A major requirement for AGILE technologies is *gateway autonomy*. Even in the case that network and Internet connections are unavailable, corresponding basic functionalities should be available. A major issue for future work is to figure out in detail in which contexts it makes sense to provide basic recommendation functionalities directly on the gateway. Since there are scalability issues with regard to the execution of standard recommendation approaches such as collaborative filtering and content-based filtering directly on the gateway, alternative representations have to be developed that can be exploited in the case that the gateway is not connected to the Internet.

In this paper we provide an overview of existing recommendation approaches in the Internet of Things (IoT). First, we give a short overview of existing work related to the application of recommendation technologies in IoT scenarios. Thereafter, we discussed application scenarios for collaborative filtering, content-based filtering, and utility-based recommendation. In contrast to related work, we showed how to apply recommendation technologies to increase the flexibility of IoT infrastructures in the context of application development and deployment.

## Acknowledgement

The work presented in this paper has been conducted within the scope of the EU Horizon 2020 project AGILE (http://agile-iot.eu/).

## References

1. L. Atzori, A. Iera, G. Morabito, The Internet of Things: A survey, Computer Networks 54 (15) (2010) 2787–2805.
2. A. Falkner, A. Felfernig, A. Haag, Recommendation Technologies for Configurable Products, AI Magazine 32 (3) (2011) 99–108.
3. A. Felfernig, R. Burke, Constraint-based recommender systems: Technologies and research issues, in: ACM International Conference on Electronic Commerce (ICEC08), Innsbruck, Austria, 2008, pp. 17–26.
4. A. Felfernig, M. Jeran, G. Ninaus, F. Reinfrank, S. Reiterer, Basic Approaches in Recommendation Systems, Recommendation Systems in Software Engineering (2014) 15–37.
5. D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, Recommender Systems – An Introduction, Cambridge University Press, 2010.
6. S. Valtolina, M. Mesiti, B. Barricelli, User-Centered Recommendation Services in Internet of Things Era, in: CoPDA2014 Workshop, Como, Italy, 2014.
7. C. Magerkurth, K. Sperner, S. Meyer, M. Strohbach, Towards Context-Aware Retail Environments: An Infrastructure Perspective, in: MobileHCI 2011, Stockholm, Sweden, 2011, pp. 1–4.
8. M. Munoz-Organero, G. Ramirez-Gonzalez, P. Munoz-Merino, C. Loos, A Collaborative Recommender System Based on Space-Time Similarities, IEEE Pervasice Computing 9 (3) (2010) 81–87.
9. J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, J. Riedl, Grouplens: applying collaborative filtering to usenet news full text, Comm. of the ACM 40 (3) (1997) 77–87.
10. M. Pazzani, D. Billsus, Learning and revising user profiles: The identification of interesting web sites, Machine Learning 27 (1997) 313–331.
11. R. Burke, Knowledge-based recommender systems, Encyclopedia of Library and Information Systems 69 (32) (2000) 180–200.
12. D. Winterfeldt, W. Edwards, Decision Analysis and Behavioral Research, Cambridge University Press, 1986.
13. R. Frey, R. Xu, A. Ilic, A Novel Recommender System in IoT, in: 5th International Conference on the Internet of Things (IoT 2015), Seoul, South Korea, 2015, pp. 1–2.
14. L. Yao, Q. Sheng, A. Ngu, X. Li, Things of Interest Recommendation by Leveraging Heterogeneous Relations in the Internet of Things, ACM Transactions on Internet Technology 16 (9) (2016) 1–25.
15. R. Burke, A. Felfernig, M. Goeker, Recommender Systems: An Overview, AI Magazine 32 (3) (2011) 13–18.
16. A. Felfernig, L. Hotz, C. Bagley, J. Tiihonen, Knowledge-based Configuration: From Research to Business Cases, 1st Edition, Elsevier/Morgan Kaufmann Publishers, 2014.