REWARD: Ontology for reward schemes

Ioannis Chrysakis[1,2], Giorgos Flouris[1], Theodore Patkos[1], Anastasia Dimou[2], and Ruben Verborgh[2]

[1] FORTH, Institute of Computer Science, Greece
{hrysakis,fgeo,patkos}@ics.forth.gr
[2] IDLab, Dept. of Electronics and Information Systems, UGent – imec, Belgium
{ioannis.chrysakis,anastasia.dimou,ruben.verborgh}@ugent.be

Abstract. Rewarding people is common in several contexts, such as human resource management and crowdsourcing applications. However, designing a reward strategy is not straightforward, as it requires considering different parameters. These parameters include, for example, management of rewarding tasks and identifying critical features, such as the type of rewards and possibilities such as gamification. Moreover, the lack of a common terminology introduces the problem of communication among experts and prevents integration among different reward strategies. An ontology can offer a common understanding among domain experts and flexible management of rewarding parameters. Apart from that, an ontology can also help in the interrelationship and integration between different reward schemes employed by different service providers. In this paper, we present REWARD, a general-purpose ontology for capturing various common features of diverse reward schemes. This ontology is a result of the CAP-A European project and its application to the crowdsourcing domain, but it is designed to cover different needs and domains.

1 Introduction

Rewarding is a common strategy for improving people's effectiveness in different domains, such as business[3], games [5], applications and services [2], or organizational workflows [7]. A successful reward strategy significantly improves the engagement of the applied target audience [1]. For example, by rewarding specific tasks' accomplishment, user loyalty increases[4], and continuous growth, recurring engagement and personal or team improvement are ensured [5].

Although studies analyze features of successful reward strategies [6, 9], most are tailor-made for specific target audience needs. Domain experts design a reward strategy from scratch, even though it is not straightforward as it requires redesigning and adapting to the specific target audience. Moreover, the lack of a common terminology introduces the problem of communication among experts and prevents integration among different reward strategies.

[3] https://www.shopify.com/enterprise/ecommerce-loyalty-programs
[4] https://www.justuno.com/blog/how-to-effectively-use-loyalty-programs-to-increase-customer-retention/

A *conceptual model* [10] can help domain experts in the design process, by setting a well-defined terminology, and in the combination of reward strategies. One way of implementing a conceptual model is to build an ontology. An ontology helps as well in the flexible adaptation of new features in any applied reward strategy and can enable integration among different service providers. Thus, a general-purpose ontology affects different beneficiaries, from service providers to end users. Service providers benefit from the flexibility and interoperability in the design process and actual application. End users may transfer features from one service provider to another (e.g., by exchanging points).

In this paper, we present REWARD, a general-purpose ontology that implements a conceptual model designed to represent a reward strategy. The ontology is published at `https://w3id.org/reward-ontology/`.

REWARD enhances and facilitates employing a reward strategy, by adopting common features of reward strategies, and casting them to a well-defined terminology. The ontology's concepts and relationships can give an extra level of common understanding, expressiveness and flexible knowledge manipulation through exploiting semantic web technologies. Thus, our proposed ontological engineering process requires appropriate (minor) changes at schema or instance level for applying any reward strategy. REWARD can help improve interoperability and integration among different reward schemes employed by different service providers and define a uniform process for creating reward strategies.

2 The REWARD Ontology

We build a general-purpose ontology that describes concepts employed by reward strategies, such as *Tasks*, *Points*, *Badges*, *Tiers* and *Rewards* [1, 5, 7, 9, 11]. For each concept we created a respective class (Fig.1). Tasks are related to actions to reach conditions to earn a reward. Thus, instances of the class *Task* capture information about the applied rewarding actions per reward strategy.

Tasks can be distinguished to *Platform-Based* and *Point-based* following a platform and user-centric approach respectively [11]. The former is related to the users' interaction with the applied system (platform) that is initiated by the service provider. In this case the rewarding is based on exclusive user's interaction with this platform, by following specific workflow scenarios. The latter is accomplished by the user to earn points due to execution of defined tasks. This distinction allows us to support tasks that are not necessarily bound to the applied platform or do not require a specific workflow of execution. Moreover, platform-based tasks allow the service provider to define specific tasks that are critical for the optimal performance of the system, and assign them to specific users; this differentiates platform-based tasks from point-based ones, which are freely selected by users without any prompt or encouragement by the system. These categories of tasks are represented as respective subclasses (*is-a* relationship) of the class Task. To support different levels of tasks according to various parameters such as difficulty of execution we introduce the class of *Task Level* which is connected with the Task class (via property: *belongsTo*).

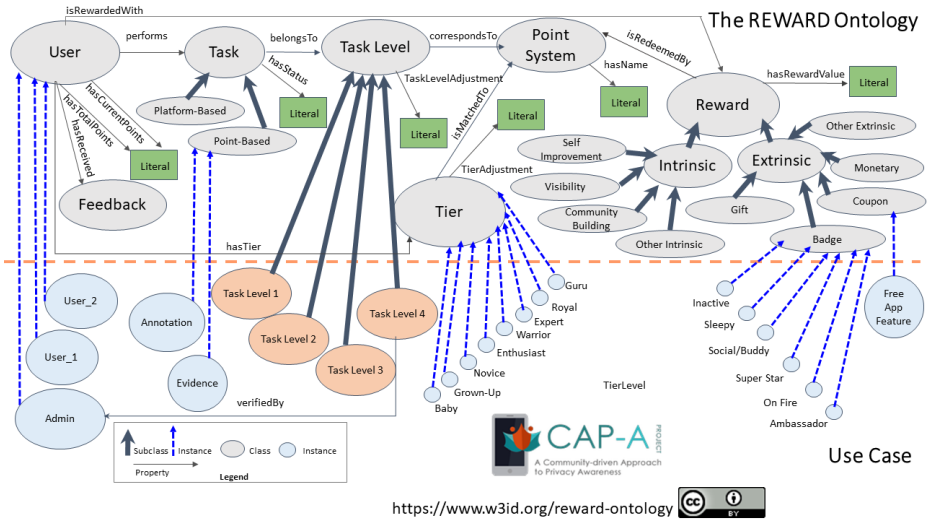

Fig. 1. The REWARD Ontology (top) and the CAP-A implementation (bottom)

Points quantify users' effort and as such they are related with the tasks [8]. In any reward strategy, a *Point System* classifies each user to a *Tier* according to currently earned points [8] and defines the redemption process, which determines the types of rewards users can claim with their points. Specifically, this information is ensured to the ontology through the properties *isMatchedTo* and *isRedeemedBy* respectively. Through multiple instantation, REWARD can support different reward schemes per user by applying a different point system. In this case we need to define different adjustment values for instances of Tier and Task Level class through the properties *TaskLevelAdjustment* and *TierAdjustment* respectively. For each point system the ontology keeps a name through the property *hasName*. Apart from the current points used for redemption, the ontology models the total acquired points for capturing users' activity through the respective properties: *hasCurrentPoints* and *hasTotalPoints*.

The users' activity is related to the defined *Badges* which represent the degree of user recognition [4] and is a type of Rewards. *Rewards* can be *intrinsic*, if users' effort is recognized based on internal motivation parameters, such as curiosity and self improvement, or *extrinsic*, if the motivation is based on external motivation parameters, such as earning a badge, a gift or a coupon [5].

Rewards are modeled under *Intrinsic* or *Extrinsic* class [5]. Intrinsic rewards are categorized in three general subclasses: *Self Improvement, Community Building and Visibility* [5,8] and *Other Intrinsic* class to capture rest cases. These can be ensured through the application of playful tasks that enable community commitment and increase peer recognition among all users.

Badges are implemented as subclass of the *Extrinsic* class, which also contains other subclasses (*Monetary, Gift, Coupon*) [8] and *Other Extrinsic class* to

capture rest cases. Badges and tiers are modeled as instances of Badge and Tier class respectively. For any critical notification the user receives feedback captured in the respective Feedback class, a fundamental feature for reward strategies [8].

The ontology supports various types of queries allowing the extraction of basic information regarding reward strategies. We present a set of representative competency queries that can be easily answered through REWARD: *(i) What is the tier of a user U? (ii) What rewards has user U gained? (iii) What is the level of task T? (iv) Which types of tasks have been completed and how many are they? (v) What type of rewards have already been redeemed by users? (vi) How many active points does user U have for the point system P? (vii) Given a minimum defined reward value, which point system(s) support any type of reward?*

3 Application

We present a use case implemented in the context of CAP-A project[5] (Fig.1, bottom part). The REWARD Ontology is used as a general-model for building the rewarding framework of CAP-A which aims to engage users in participating in crowdsourcing tasks for improving privacy awareness on mobile applications [3].

We define *instances* of the *Point Based* class related to the privacy context, that lead to rewarding, such as *Annotation* on Term of Service documents and adding of *Evidence* on a privacy topic. Four task levels are defined to facilitate task management. Each of them represents different level of difficulty and captured in the ontology as *subclasses* of the *Task Level* class (*Task Level 1, Task Level 2, Task Level 3, Task Level 4*). Furthermore, each task level subclass determines the task's visibility and applicability according to the user's tier and the *Point System*. User's tier is calculated according to the earned points.

In the CAP-A project, we use the following tiers: *Baby, Grown-Up, Novice, Enthusiast, Warrior, Expert, Guru, Royal* which are implemented as instances of *Tier class*. Rewards are stored as *instances* of the respective *Reward* class and include free features on mobile applications or acquirement of specific Badges. CAP-A Badges are modeled as *instances* of the *Badge* class (*Inactive, Sleepy, Social/Buddy, Super Star, On Fire, Ambassador*), while the free app features are modeled as instances of the *Coupon* class. The badges are given to users based on their activity in terms of specific task accomplishments, continuous work or gained expertise, but we skip further details due to lack of space.

The Rewarding Loyalty Programs by Air flight companies is another use case where the REWARD ontology can be used. An instance of a travel task belongs to *Point-Based* class. Tasks can be defend depending on the loyalty program, such as booking accommodation, renting a car, shopping etc., and with two levels of difficulty, depending om whether they require verification by service providers (e.g., Reclaim miles). The *Point System* denotes the appropriate lower point bounds for promoting users to different tiers (e.g., blue, silver, gold tier). The rewards are mostly *extrinsic*, in terms of free tickets, discounts and

[5] https://www.cap-a.eu

coupons. Frequent Travelers are instances of *User* class that uses the specific loyalty program.

Consequently, we conclude that REWARD can be adapted easily to these different domains as it is designed to be. Our future plans include evaluation of the ontology in more use-cases and domains. Finally, as REWARD provides a generic model for designing any reward scheme, it leaves room for collaboration among service providers by offering more features on their common end users. The first step for service providers would be to try matching their existing reward schemes to REWARD in order to incorporate features such as consolidation of points or offering combined rewards.

Acknowledgements

This work has been supported by the CAP-A project which has received funding from the European Union's Horizon 2020 research and innovation programme under the NGI_TRUST grant agreement no 825618. Ruben Verborgh is a post-doctoral fellow of the Research Foundation – Flanders (FWO). The authors wish to thank Yannis Roussakis, Elias Tzortzakakis and Elisjana Ymeralli for contributions in the validation process of CAP-A use case.

References

1. Antikainen, M.J., Vaataja, H.K.: Rewarding in open innovation communities–how to motivate members. International Journal of Entrepreneurship and Innovation Management (2010)
2. Detwiller, D., Sparks, R.: Rewarding users for sharing digital content (2012), uS Patent App. 13/306,917
3. Flouris, G., Patkos, T., Chrysakis, I., Konstantinou, I., Nikolov, N., Papadakos, P., Pitt, J., Roman, D., Stan, A., Zeginis, C.: Towards a collective awareness platform for privacy concerns and expectations. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer (2018)
4. Hamari, J., Eranti, V.: Framework for designing and evaluating game achievements. In: Digra conference. Citeseer (2011)
5. Kavaliova, M., Virjee, F., Maehle, N., Kleppe, I.A.: Crowdsourcing innovation and product development: Gamification as a motivational driver. Cogent Business & Management (2016)
6. Morschheuser, B., Hamari, J., Koivisto, J.: Gamification in crowdsourcing: a review. In: 49th Hawaii International Conference on System Sciences. IEEE (2016)
7. Prendergast, C.: The provision of incentives in firms. Journal of economic literature (1999)
8. Reiners, T., Wood, L.C.: Gamification in Education and Business. Springer (2015)
9. Scekic, O., Truong, H.L., Dustdar, S.: Incentives and rewarding in social computing. Communications of the ACM (2013)
10. Wand, Y., Storey, V.C., Weber, R.: Analyzing the meaning of a relationship. ACM Transactions on Database Systems (1999)
11. Yang, D., Xue, G., Fang, X., Tang, J.: Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In: Proceedings of the 18th annual international conference on Mobile computing and networking. ACM (2012)