Proceedings of the Federated Conference on Computer Science and Information Systems pp. 705–709 DOI: 10.15439/2019F262 ISSN 2300-5963 ACSIS, Vol. 18

Visual Rule Editor for E-Guide Gamification Web Platform

Artur Kulpa
University of Szczecin,
Faculty of Economics and Management,
ul. Mickiewicza 64, 71-101
Szczecin, Poland
Email: artur.kulpa@usz.edu.pl

Jakub Swacha
University of Szczecin,
Faculty of Economics and Management,
ul. Mickiewicza 64, 71-101
Szczecin, Poland
Email: jakub.swacha@usz.edu.pl

Karolina Muszyńska
University of Szczecin,
Faculty of Economics and Management,
ul. Mickiewicza 64, 71-101
Szczecin, Poland; Email:
karolina.muszynska@usz.edu.pl

Abstract—**Gamification is applied in different information systems to motivate the users and make their experience with the system richer and more engaging. Gamification employed in e-guides aims at enhancing the process of visiting a tourist attraction. Even though each tourist attraction is unique and requires an individual gamification scheme, similarities in the components and procedures used to develop such schemes led to the development of a generic e-guide gamification framework. One of its main principles is to store the gamification rules as a content separate from the engine to process them. This way, the rules can be easily edited by subject matter experts. This paper describes a visual rule editor developed to facilitate this process.**

I. Introduction

GAMIFICATION, understood as "the use of game-design and game psychology in non-game settings to engage the target audience and motivate specific behaviors" [1], has been considered as "one of the significant new trends in the development of services and applications in the software industry" [2]. It can be applied to information systems of various character [3]. One of these are multimedia visitor guidance systems, better known as e-guides, which may employ gamification to enhance the tourist attraction visiting process [4].

Tourist attractions may significantly differ in their character, and e-guides may as well differ in their form and functionality. For this reason, there cannot be a fixed scheme for e-guide gamification. Rather, individual gamification schemes have to be designed for respective tourist attractions with specific characteristics of their own. There is still significant similarity among components and procedures used to develop gamification schemes for different tourist attractions. Motivated by this observation, a generic e-guide gamification framework has been proposed, aiming to standardize architectural and design solutions so that it could be easier to implement gamification to e-guides, reuse gamification layer among different e-guides, and maintain and update gamified e-guides [5].

An indispensable ingredient of any gamification scheme are the rules linking users' actions to game-inspired consequences, such as specific rewards or feedback [6]. While they could be developed as a part of e-guide software, the framework mentioned above proposes not only the separation of the core e-guide functionality and content from the gamification engine and rules, but also the separation of gamification rules and the engine to process them, so that the rules could be treated as a kind of content, separately edited and transferred among e-guides [5].

For this purpose, a common notation of gamification rules for e-guides has to be used, and such notation has been proposed [7]. Although, the textual notation it uses was designed for readability, in the course of implementation of the BalticMuseums: Love IT! project [8], it was found that the tourist attraction personnel who was responsible for devising the e-guided tours featuring gamification, and who had no programming experience found it difficult to use.

In this paper, we describe a solution for this problem in a form of a dedicated visual rule editor, so that the prior knowledge of the notation syntax, keywords, and appropriate parameters is no longer required from the gamification designer. This approach follows the example of general-purpose task automation services where the provision of visual rule editor is considered as an architectural requirement [9, p. 13].

The structure of the paper is as follows: section II gives a short glimpse of prior work on this topic. In section III, we present the user interface of the visual rule editor, so that the reader could see how it can be used to define gamification rules. Section IV provides necessary technical information about the implementation of the visual rule editor. In section V, the results of its evaluation are reported. The final section concludes.

The presented rule editor was developed as a part of the BalticMuseums: Love IT! project and part-financed from the European Regional Development Fund within the Interreg South Baltic Programme.

This paper was developed with a financial support from a project financed within the framework of the program of the Minister of Science and Higher Education under the name "Regional Excellence Initiative" in the years 2019-2022; project number 001/RID/2018/19; the amount of financing PLN 10,684,000.00.

II. RELATED WORK

The idea of replacing text statements with a form to be filled in can be traced back to QBE database query language released commercially already in 1978 [10]. In its contemporary implementations (such as in Microsoft Access 2019), not only the query building form has a predefined structure (so that there is no need to know the syntax), but also the available options are simply checked or unchecked (rather than typed in) and wherever the set of possible parameter values is predefined, they can be chosen from a list rather than input as text (which has to be done with other parameter values).

Although some research results show that trained and experienced users prefer textual notation than forms (see e.g. [11]), for a first-time or occasional user, the advantage of using a form over writing text in a language unknown to him/her is obvious. This is a reason for which a number of similar query languages were developed, including [12]:

- Aggregates-by-Example, Summary-Table-by-Example, and Query-by-Statistical-Relational-Table designed for querying statistical and scientific databases;
- Time-by-Example designed for querying historical databases;
- Generalized-Query-by-Example designed for querying relational, network, and hierarchical databases;
- Picquery and Query-by-Pictorial-Example designed for querying image databases.

As gamification rules, like database queries, also have a well-defined context and structure, they are suitable for editing using forms. This has been proven by well-known enterprise gamification platforms, such as Bunchball Nitro [13] or Gametize [14], which use forms to specify the rules.

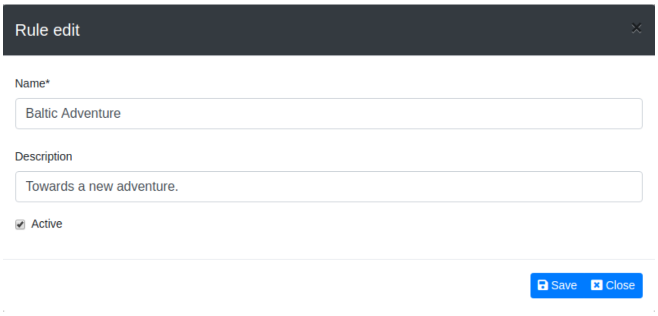
In the following two sections, we will show how this approach has been implemented to the e-guide gamification rule specification.

III. VISUAL RULE EDITOR

The described visual rule editor forms a part of the e-guide content management system which in turn is a part of the e-guide gamification web platform developed within the BalticMuseums: Love IT! international project [8]. The purpose of the editor is to facilitate gamification designers to add, view, modify and delete gamification rules which are to be applied by the e-guide gamification web service [15], itself called by e-guide client applications running on tourist attraction visitors' mobile devices.

The visual rule editor features a list of defined rules along with the search and pagination mechanisms, as shown in Fig. 1. The editor follows the principle of a clear division of rule specification into three parts: the name (identifying a rule), conditions (required to trigger a rule) and results (caused by a rule) parts as presented in [5]. This allows for easy specification of rules having multiple conditions and results. Both the conditions and results regarding a given rule are displayed next to its name in a readable form and the standard functionalities of adding, viewing, modifying and deleting them are provided.

Adding and modifying rules is done using the "Edit rule" form presented in Fig. 2. Similar forms are used to add or edit a condition or a result for a selected rule.



Rule edit

Name*

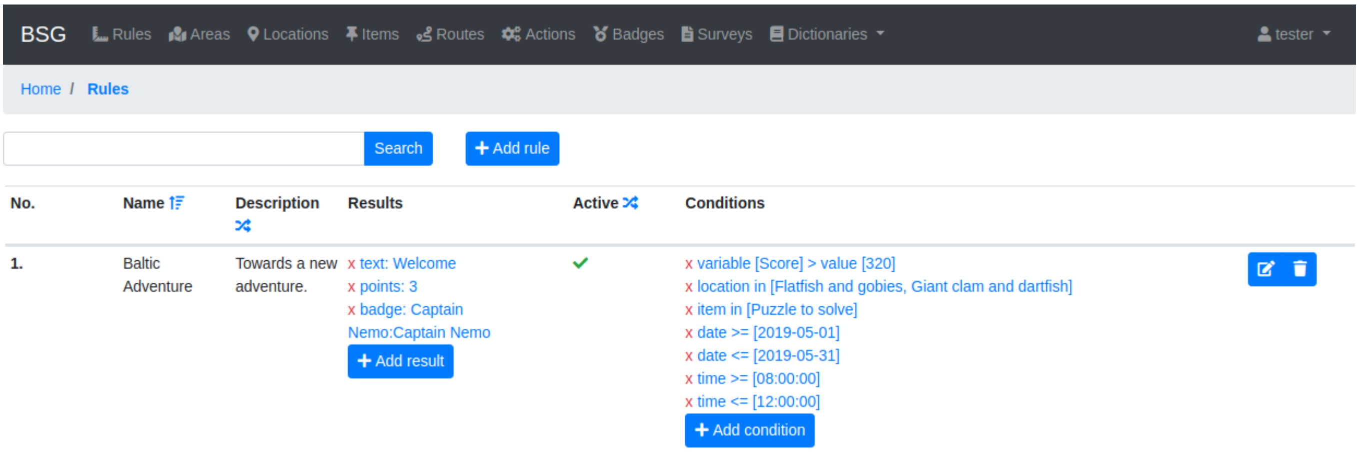
Baltic Adventure

Description

Towards a new adventure.

☑ Active

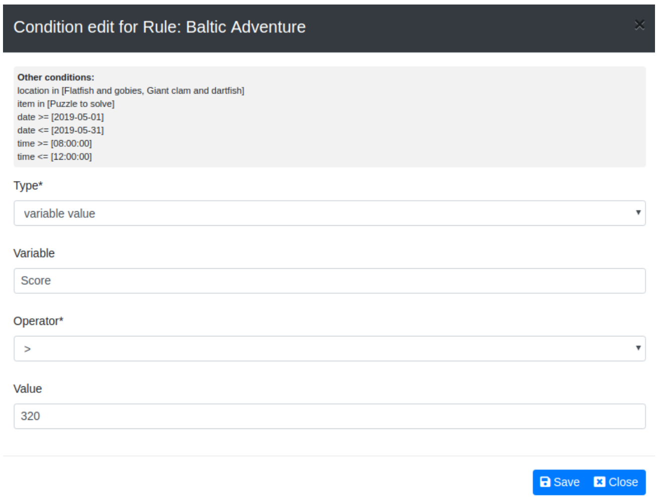
Fig. 2 Rule edit form



No.	Name	Description	Results	Active	Conditions
1.	Baltic Adventure	Towards a new adventure.	x text: Welcome x points: 3 x badge: Captain Nemo:Captain Nemo + Add result	✓	x variable [Score] > value [320] x location in [Flatfish and gobies, Giant clam and dartfish] x item in [Puzzle to solve] x date >= [2019-05-01] x date <= [2019-05-31] x time >= [08:00:00] x time <= [12:00:00] + Add condition

Fig. 1 Rules definition module in the e-guide gamification system

Figure 3 presents a condition edit form with a *Var_Value* type condition, which makes it possible to specify: the name of the variable that is to occur in the description of the event that triggers the rule, the comparison operator and the threshold value at which the condition is met. For the user's convenience, other conditions specified for the rule whose condition is being edited are shown above the edit text boxes. The available comparison operators include those for comparing two values (=, <>, >, >=, <, <=) and checking the existence of a value in a set (*in, not in*).



Condition edit for Rule: Baltic Adventure

Other conditions:
location in [Flatfish and gobies, Giant clam and dartfish]
item in [Puzzle to solve]
date >= [2019-05-01]
date <= [2019-05-31]
time >= [08:00:00]
time <= [12:00:00]

Type*
variable value

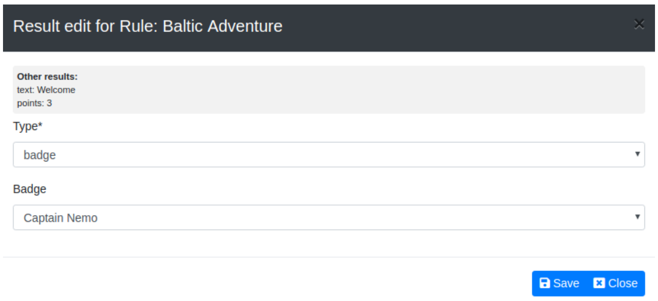
Variable
Score

Operator*
>

Value
320

Fig. 3 Condition edit form

Figure 4 presents a result edit form with an example of a *Badge* type result. The other two result types currently available are *Points* and *Text* (to be displayed to the visitor). The list of result types is to be extended in consideration of the detailed results of the evaluation survey (see section V). As with editing the conditions, for the user's convenience, other results specified for the rule whose condition is being edited are shown above the edit text boxes.



Result edit for Rule: Baltic Adventure

Other results:
text: Welcome
points: 3

Type*
badge

Badge
Captain Nemo

Fig. 4 Result edit form

IV. Data Model For Representation of Rules

The data model for the representation of rules of the gamification service is based on classes implemented in the Django framework (Fig. 5). During implementation, this model is translated into SQL, and then a SQL script is executed in the PostgreSQL database management system, which results in the creation of the respective tables. The

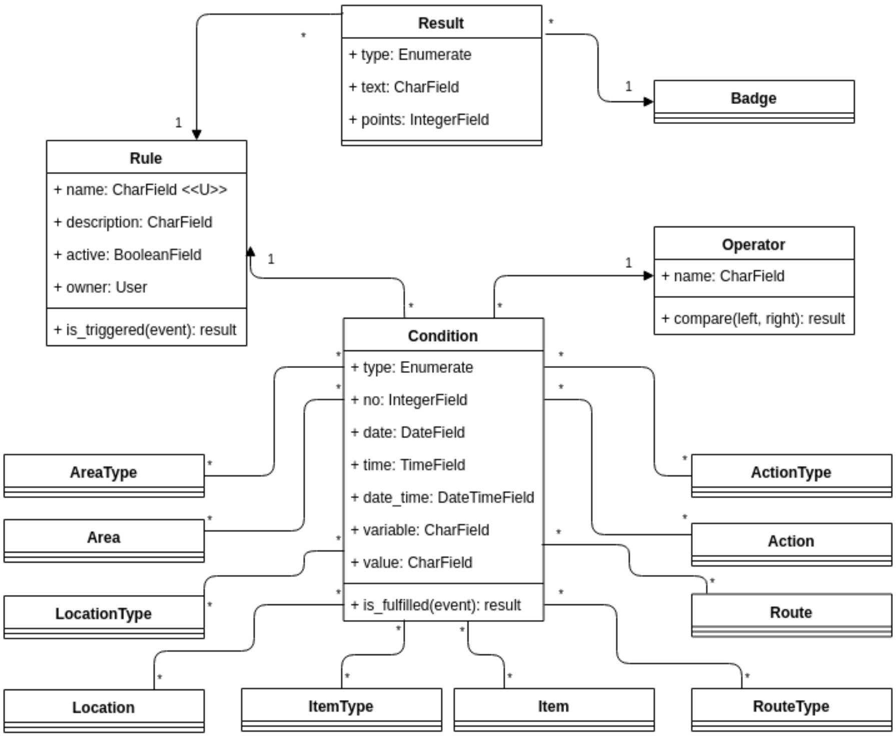


Fig. 5 Data model for representation of rules

main class of the model is the Rule class which defines objects storing basic information about each rule – its *name* and *description*. The content of the *name* field has to be unique (no two rules may have the same name). An additional *active* field has been introduced to indicate whether, for a given rule, the data stored in objects of associated classes (see below) can be modified or not, as well as whether the rule can be executed by the e-guide gamification web service. Each rule is assigned to the user who created it. This way, only an owner of the rule can see and modify it.

The *is_triggered* method defined within the Rule class checks if the rule has been triggered as a result of an event (whose description is the input parameter) and if positive, returns the results assigned to the rule.

The Result class contains a field specifying the result type. Currently, there are three available result types: Text, Points, and Badge. Depending on the result type, the appropriate field is filled in. In the case of Text type, the *text* field is set to the result message to the visitor. In the case of Points type, the *points* integer field is set to the number of points to be received by the visitor. In the case of Badge type, an object from the Badge class specifying the badge to be awarded to the visitor is assigned to the Result class object.

The Condition class is meant to map any conditions associated with a given rule. One rule can have many conditions and a certain condition can be associated with only one rule. The *type* field specifies the object of the left side of the condition. There are several entities in the e-guide gamification system that can be used to form the right side of the condition. They can be seen on the data model associated with the Condition class, i.e.: *AreaType, Area, LocationType, Location, ItemType, Item, RouteType, Route, ActionType* and *Action*. Other available condition types include *Date, Time, Date_Time* and *Var_Value*. For the *Var_Value* type, both the variable name and its threshold value must be specified.

The *is_fulfilled* method, defined within the Condition class, specifies if a certain condition has been fulfilled, based on an event description (the input parameter).

Each object of the Condition class is associated with an object of the Operator class which defines the way of comparing both sides of the condition. The *compare* method of the Operator class is used to perform the comparison.

V. Evaluation

The preliminary evaluation of the described visual rule editor was based on a qualitative survey among the representatives of tourist attractions participating in the BalticMuseums: Love IT! project [8] who are in progress of development of their respective gamified e-guides and to whom the tool was presented earlier. The three rule editor evaluation questions constituted a part of a larger survey aimed at defining future work directions and possible technical improvements, therefore the answers could be considered as objective. Out of the five tourist attractions participating in the BalticMuseums: Love IT! project, answers from four (Gdynia Aquarium, Experyment Science Center, Malmö Museums and NaturBornholm) were received (six answers in total as there were two organizations with two representatives from each of them participating in the survey). Note that though the number of the tourist attractions involved in the survey is low, they differ significantly in their character, which makes their answers far more representative than if they all belonged to a single category of attractions.

Answering to the first evaluation question, all the six surveyed representatives agreed that a form-based gamification rule editor is better than writing rules as text (using a dedicated rule definition language). This confirms the original observations that led to the development of the editor.

The second question asked whether there were rules that the tourist attraction representatives invented but did not manage to define using the editor. Half of the surveyed (three of six representatives from two of the four involved attractions) answered positively, with one respondent pointing to her (or her team's) possible lack of skills in using the tool, and the two other (from one attraction) suggesting that the rule definition form has to be extended.

In the third question, the respondents were asked if they encountered other problems with defining rules. Again, half of the surveyed (three of six representatives from two of the four involved attractions) answered positively. This time, all of them pointed to the lack of a good manual as a reason for those problems.

VI. Conclusion

A key point in the implementation of gamification is the specification of rules governing the system. While such rules may be embedded in the gamification software, in many circumstances it makes a lot of sense to have them defined separately so that a change in gamification rules would not require a change in the software handling them.

This is especially true in the case of e-guide gamification where the rules should make use of the specificity of a tourist attraction, which strengthens the role of subject matter experts (such as guides and educators) rather than gamification experts. Another reason for such an approach to e-guide gamification is due to quickly changing content featured in many tourist attractions (such as those having seasonal exhibits or short-living fauna species on display).

While e-guide gamification rules can be effectively represented in textual notation [7], their editing by subject matter experts (rather than IT professionals) could be much simpler using predefined forms.

The visual rule editor described in this paper was designed following this approach. All the surveyed representatives of the tourist attractions introduced to it unanimously agreed on their preference for the form-based rather than textual rule specification. On the other hand, the survey revealed that the decision to make the editor simple by limiting the possible scope of defined rules (as compared to the textual notation) resulted in some of the tourist attractions representatives being unable to define sophisticated rules they had invented. Moreover, even though the operation of the editor is intuitive, the users encountered some problems which could be addressed by providing a better user manual.

The identified drawbacks will be addressed in the next version of the rule editor, whose development constitutes the nearest future work, after which another evaluation survey is planned, having an extended scope and involving more tourist attractions.

References

[1] A. Marczewski, “Defining gamification – what do people really think?,” http://www.gamified.uk/2014/04/16/defining-gamification-people-really-think, last accessed 31.5.2018.

[2] J. Kasurinen and A. Knutas, “Publication trends in gamification: A systematic mapping study,” *Computer Science Review*, vol. 27, 2018, pp. 33–44. doi:10.1016/j.cosrev.2017.10.003.

[3] J. Swacha, „Gamification in Enterprise Information Systems: What, Why and How,“ in: *Proceedings of the Federated Conference on Computer Science and Information Systems*, *Annals of Computer Science and Information Systems*, vol. 8, 2016, pp. 1229–1233. doi: 10.15439/2016F460.

[4] J. Swacha and R. Ittermann, “Enhancing the tourist attraction visiting process with gamification: key concepts,” *Engineering Management in Production and Services*, vol. 9, no. 4, pp. 59–66, 2017. doi:10.1515/emj-2017-0031.

[5] J. Swacha and K. Muszynska, “Towards a Generic eGuide Gamification Framework for Tourist Attractions’, in: *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts, CHI PLAY 2018,* Melbourne, 2018, pp. 619–625. doi: 10.1145/3270316.3271535.

[6] J. Swacha, „Architecture of a dispersed gamification system for tourist attractions,“ *Information*, vol. 10, no. 1, 2019, art. 33. doi: 10.3390/info10010033.

[7] J. Swacha, “Representation of Events and Rules in Gamification Systems,” *Procedia Computer Science*, vol. 126, 2018, pp. 2040–2049. doi: 10.1016/j.procs.2018.07.248.

[8] BalticMuseums: Love IT! Project website. http://www.balticmuseums.info, last accessed 15.5.2019.

[9] M.C. Barrios, “A Personal Agent Architecture for Task Automation in the Web of Data. Bringing Intelligence to Everyday Tasks,” PhD Thesis, Universidad Politécnica de Madrid, Madrid, 2016.

[10] M. M. Zloof, “Query-by-Example: A data base language,” *IBM Systems Journal*, vol. 16, no. 4, pp. 324–343, 1977. doi: 10.1147/sj.164.0324.

[11] J. M. Boyle, K. F. Bury, and R. J. Evey, “Two Studies Evaluating Learning and Use of QBE and SQL,” *Proceedings of the Human Factors Society Annual Meeting*, vol. 27, no. 7, pp. 663–667, 1983.doi: 10.1177/154193128302700732.

[12] G. Özsoyoglu and H. Wang, “Example-based graphical database query languages,” *Computer*, vol. 26, pp. 25–38, 1993. doi: 10.1109/2.211893.

[13] Nitro: The Enterprise Engagement Platform Powered by Gamification, https://www.bunchball.com/products/nitro-platform, last accessed 15.5.2019.

[14] Community Engagement, Gametized, https://gametize.com/index, last accessed 15.5.2019.

[15] J. Swacha and A. Kulpa, “A Cloud-Based Service for Gamification of eGuides,” in 2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Barcelona, 2018, pp. 220–224. doi: 10.1109/W-FiCloud.2018.00042.