

STRUKTURA FUNKCJONALNA ZGAMIFIKOWANEGO SYSTEMU ZARZĄDZANIA NAUCZANIEM

JAKUB SWACHA¹

Uniwersytet Szczeciński
Wydział Nauk Ekonomicznych i Zarządzania
¹ e-mail: jakubs@wneiz.pl

SŁOWA KLUCZOWE

informatyka w edukacji, gamifikacja, systemy zarządzania nauczaniem, LMS

STRESZCZENIE

Gamifikacja jako forma wzbudzania motywacji wewnętrznej poprzez wykorzystanie elementów zaczerpniętych z gier cieszy się coraz większym zainteresowaniem instytucji edukacyjnych i nauczycieli. Jednak pomimo iż wzrasta liczba dostępnych podręczników gamifikacji, to jej praktyczna implementacja nadal wymaga znaczących nakładów sił i środków. Choć popularne systemy zarządzania nauczaniem udostępniają już moduły pozwalające na wprowadzanie do kursów elementów gamifikacji, ich wykorzystanie jest pracochłonne, a możliwości nader ograniczone. Rozwiązaniem jest wykorzystanie systemu zarządzania nauczaniem, w którym gamifikację uwzględniono już na etapie projektowania, co pozwoli na jej łatwe wykorzystanie przez projektujących i prowadzących kursy. W niniejszym opracowaniu zaproponowano i skonkretyzowano wizję struktury funkcjonalnej takiego systemu.

Wprowadzenie

Gamifikację można zdefiniować jako „tworzenie doświadczeń przypominających gry w kontekstach nie będących grami” (Marczewski, 2014). Przypisuje się jej charakterystyczną dla gier zdolność budzenia wśród uczestników motywacji wewnętrznej (Muntean, 2011), co w pewnym stopniu znalazło potwierdzenie w opublikowanych już wynikach badań (Hamari, Koivisto, Sarsa, 2014). Jest to z pewnością wielka zaleta w kontekście edukacji, stąd szerokie zaintereso-

wanie wykorzystaniem gamifikacji dla wspomagania nauczania (zob. np. Kapp, 2012 i prace tam cytowane).

Skuteczna gamifikacja nauczania wymaga budowy systemu reguł, który będzie możliwie kompletny i precyzyjny, co z kolei wiąże się z koniecznością monitorowania i rejestrowania dużej liczby różnego rodzaju danych, stanowiących wejście i/lub inicjujących poszczególne reguły (Swacha, 2015). Naturalną tego konsekwencją jest potrzeba informatycznego wspomaganie gamifikacji, tak by uwolnić osoby prowadzące kurs od ręcznej ewidencji wszystkich niezbędnych danych. Jakkolwiek do tego celu można wykorzystać choćby tak uniwersalne narzędzia, jak arkusze kalkulacyjne, to z pewnością najbardziej naturalne dla niego środowisko oprogramowania stanowią systemy zarządzania nauczaniem (ang. *Learning Management Systems*, LMS). Przy czym choć dla większości popularnych systemów LMS (jak choćby Moodle czy Claroline) dostępne są już rozszerzenia pozwalające na wprowadzenie do realizowanych z ich wykorzystaniem kursów wybranych elementów gamifikacji, ich możliwości są ograniczone, a wymagany nakład sił znaczący. Brak jak dotąd systemu LMS, który byłby zgamifikowany już na etapie projektowania, tak by pozwolić z jednej strony na bardzo łatwą i szybką gamifikację kursów przez użytkowników (nauczycieli) o niewielkich wymaganiach, z drugiej zaś dać nieskrępowane możliwości rozbudowy zasad gamifikacji bardziej wymagającym użytkownikom. W artykule starano się nakreślić założenia funkcjonalne dla takiego systemu w celu stworzenia wzorca dla jego implementacji. Pierwszym krokiem ku temu było zdefiniowanie specyficznej architektury zgamifikowanego LMS (Swacha, 2014), natomiast drugim – stanowiącym temat tego artykułu – był projekt jego struktury funkcjonalnej.

Zasadniczą część artykułu podzielono na sześć części. W pierwszej z nich przybliżono wybrane cechy architektury zgamifikowanego LMS, mające znaczenie dla projektu jego struktury funkcjonalnej, ale także wyjaśniające podział na kolejne części, które będą odnosić się do poszczególnych obszarów funkcjonalnych zgamifikowanego LMS, według wspomnianej architektury.

Podstawowe cechy architektury zgamifikowanego LMS

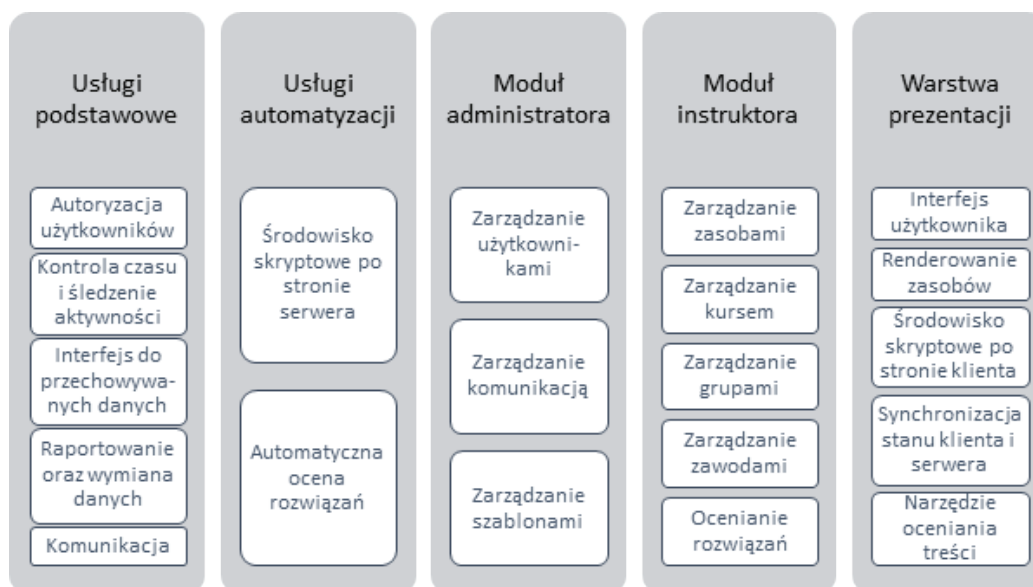
Architektura zgamifikowanego LMS stanowi fuzję klasycznych architektur LMS (zob. np. Avgeriou, Retalis, Skordalakis, 2003) i potrzeb gamifikacji, wynikających ze specyfiki jej elementów (zob. np. Werbach, Hunter, 2012). Definiując ją, starano się przede wszystkim spełnić następujące wymagania (Swacha, 2014):

1. Możliwość łatwego i szybkiego przygotowania kursu o niewyrafinowanych zasadach dzięki dostępności spredefiniowanych mechanizmów.
2. Możliwość implementacji dowolnie skomplikowanych reguł gamifikacji dzięki dostępności języka pozwalającego na definiowanie mechanizmów własnego pomysłu.
3. Otwartość na gamifikację wszystkich składowych kursu, niezależnie od ich typu, w szczególności materiałów wykładowych, ćwiczeń do samodzielnego wykonania i testów zaliczeniowych.

4. Dostępność systemu dla użytkowników końcowych (uczniów) niezależnie od miejsca, sprzętu i platformy systemowej.
5. Persystencja stanu systemu, umożliwiająca przerywanie i kontynuowanie nauki w dowolnym jej momencie, bez szwanku dla rejestrowanego postępu uczestnika.

Przesłanki 2–5 mają konsekwencje dotyczące technologii implementacji systemu zgodnego z omawianą architekturą. W szczególności przesłanka 4 wskazuje technologię webową jako szczególnie odpowiednią dla implementacji LMS o takiej architekturze, zaś przesłanka 2 (a pośrednio także 3) wskazuje na konieczność ujęcia w systemie środowiska interpretującego skrypty definiujące niestandardowe reguły gamifikacji.

W technologii webowej środowiska skryptowe mogą być realizowane zarówno po stronie serwera, jak i klienta (przeglądarki internetowej). Zaletą pierwszego rozwiązania jest persystencja danych i ich dostępność dla wszystkich użytkowników (w tym nauczycieli) podłączonych do serwera. Zaletą drugiego: dostęp do danych o szczegółowych działaniach użytkownika w środowisku przeglądarki internetowej (które mogą być istotne dla pewnych reguł gamifikacji), odciążenie serwera od zadań obliczeniowych oraz zmniejszenie obciążenia połączenia internetowego (Swacha, Baszuro, 2013). Ponieważ obie grupy zalet są pożądane dla zgamifikowanego LMS, w omawianej architekturze przewidziano dwa środowiska skryptowe: jedno działające po stronie serwera, a drugie – klienta. Jednocześnie z piątej przesłanki wynika wymóg ciągłej synchronizacji stanu obu tych środowisk.



Rysunek 1. Komponenty architektury zgamifikowanego LMS

Źródło: opracowanie własne.

Kluczowe komponenty architektury zgamifikowanego LMS przedstawiono na rysunku 1. Jak widać, wydzielono tam pięć obszarów funkcjonalnych, a mianowicie:

- usługi podstawowe – wspomagające działanie funkcji pozostałych obszarów,
- usługi automatyzacji – zapewniające możliwość automatycznego stosowania reguł gamifikacji oraz automatycznej oceny rozwiązań zgłaszanych przez uczestników kursów,
- moduł administratora – grupujący funkcje związane z administrowaniem systemem,
- moduł instruktora – grupujący funkcje wykorzystywane przez osoby prowadzące kursy,
- warstwę prezentacji – obejmującą funkcje służące prezentacji materiałów dydaktycznych, ale także rejestrowaniu zachowań uczestników kursu i kontrolujące elementy interaktywne kursu.

Usługi podstawowe

Usługi podstawowe obejmują pięć podobszarów funkcjonalnych (rysunek 1). Pierwszym z nich jest autoryzacja użytkowników. Grupuje on następujące funkcje:

- zalogowanie użytkownika (administratora, instruktora, uczestnika kursu) do systemu,
- wylogowanie użytkownika z systemu,
- weryfikacja praw dostępu do wskazanej funkcji systemu przez zalogowanego użytkownika,
- weryfikacja praw dostępu do wskazanego zasobu przez zalogowanego użytkownika.

Drugi wyróżniony podobzarsz to kontrola czasu i śledzenie aktywności. Zawarto w nim następujące funkcje:

- wywoływanie wskazanych funkcji systemu o zaplanowanych porach lub mających stanowić następstwo określonych działań użytkownika (np. automatyczne sprawdzenie rozwiązania po jego wprowadzeniu przez uczestnika kursu do systemu),
- uruchamianie reguł gamifikacji inicjowanych nastaniem określonej godziny lub upływem określonego czasu od momentu rozpoczęcia wskazanej aktywności przez użytkownika,
- rejestrowanie działań użytkowników w systemie,
- rejestrowanie funkcji automatycznie wywołanych przez system.

Interfejs do przechowywanych danych stanowi trzeci podobzarsz wyodrębniony wśród usług podstawowych. Zapewnia on dostęp do zasobów danych przechowywanych przez system (w szczególności metadanych i danych dotyczących kursów i wykorzystywanych w nich treści oraz użytkowników i ich postępów w nauce), ograniczony jedynie prawami dostępu, którymi włada zalogowany użytkownik (weryfikowanymi z wykorzystaniem funkcji podobzarsza kontrola czasu i śledzenie aktywności), a niezależny od faktycznej technologii bazodanowej (organizacji systemu plików) wykorzystywanej przez daną instancję systemu. Obejmuje on następujące funkcje:

- dodanie nowego rekordu/zasobu/opisu,
- sprawdzenie istnienia wskazanego rekordu/zasobu/opisu,

- odczytanie istniejącego rekordu/zasobu/opisu,
- aktualizacja istniejącego rekordu/zasobu/opisu,
- usunięcie istniejącego rekordu/zasobu/opisu.

Kolejny wyróżniony podobszar to raportowanie oraz wymiana danych. Zawarto w nim następujące funkcje:

- pobranie kolekcji danych o określonym zakresie i sformatowanej według wskazanego szablonu,
- wprowadzenie kolekcji danych sformatowanej według wskazanego szablonu.

Należy tu wyjaśnić, że powyższe funkcje służą zarówno do przygotowania danych do wizualizacji przed użytkownikiem, jak i ich wyeksportowania do pliku (tudzież zaimportowania z pliku) w określonym formacie wymiany danych celem przeniesienia do innej instancji systemu: o formie danych przesądza wybrany szablon formatu. Sam dostęp do danych realizowany jest z wykorzystaniem funkcji podobszaru interfejsu do przechowywanych danych, z kolei zarządzanie szablonami realizowane jest poprzez funkcje modułu administratora.

Ostatnim podobszarem wydzielonym wśród usług podstawowych jest komunikacja. Zawiera on funkcje służące do przekazywania wiadomości pomiędzy użytkownikami z wykorzystaniem wbudowanych lub zewnętrznych usług komunikacyjnych (wybór aktywnych usług realizowany jest poprzez funkcje modułu administratora). Obejmuje on następujące funkcje:

- przekazanie trwałej wiadomości osobistej (przykładowa usługa: poczta elektroniczna),
- przekazanie nietrwałej wiadomości osobistej (przykładowa usługa: komunikator internetowy),
- przekazanie wiadomości otwartej (przykładowa usługa: forum dyskusyjne, portal społecznościowy),
- przekazanie powiadomienia (przykładowa usługa: SMS, powiadomienia push).

Projekt systemu nie definiuje wbudowanych usług komunikacyjnych: są one dostępne jako rozszerzenia systemu, z technicznego punktu widzenia stanowiące rodzaj komponentu kursu (analogicznie do materiałów dydaktycznych).

Usługi automatyzacji

Usługi automatyzacji obejmują jedynie dwa podobszary funkcjonalne (rysunek 1). Pierwszym z nich jest środowisko skryptowe po stronie serwera. Udostępnia ono następujące funkcje:

- uruchomienie skryptu administratora, mogącego służyć celom konserwacji, aktualizacji lub rekonfiguracji systemu i posiadającego szeroki dostęp do funkcji i danych systemu,
- uruchomienie skryptu instruktora, mogącego implementować złożone reguły gamifikacyjne, procedury oceny rozwiązań nadsyłanych przez uczestników kursu lub służyć do konfiguracji kursu (np. przepisaniu wszystkich uczestników jednego kursu spełniających pewne warunki do innego kursu), i mającego ograniczony dostęp do funkcji i danych systemu,

- uruchomienie skryptu uczestnika kursu, mające zastosowanie jedynie w przypadku kursów programistycznych, a służące weryfikacji poprawności lub innych cech jakościowych rozwiązań stanowiących wykonywalne programy komputerowe, gdzie wymagane jest całkowite odizolowanie środowiska wykonywania skryptu od funkcji i danych systemu.

Drugim podobszarem jest automatyczna ocena rozwiązań, zawierająca jedynie dwie funkcje, takie jak:

- weryfikacja poprawności rozwiązania, porównująca przekazane przez uczestnika rozwiązanie z przypisanym do ćwiczenia prawidłowym wynikiem lub wywołująca skrypt weryfikujący poprawność (wynik tej funkcji może mieć jedynie wartość „tak” lub „nie”),
- ocena rozwiązania, wywołująca skrypt oceniający różne aspekty rozwiązania i zwracający zestaw danych określający poszczególne składowe wyniki oceny i, opcjonalnie, ocenę całościową.

Moduł administratora

Moduł administratora obejmuje trzy podobszary funkcjonalne (rysunek 1). Pierwszym z nich jest zarządzanie użytkownikami, służący zapewnieniu dostępu do systemu i jego zasobów przy jednoczesnym ograniczeniu go do autoryzowanych użytkowników. Grupuje on następujące funkcje:

- rejestracja nowego użytkownika,
- uzyskanie listy zarejestrowanych użytkowników według podanych kryteriów,
- uzyskanie zbioru uprawnień przypisanych wskazanemu użytkownikowi,
- zmiana uprawnień przypisanych wskazanemu użytkownikowi,
- wyrejestrowanie wskazanego użytkownika.

Drugim podobszarem jest zarządzanie komunikacją pozwalające na konfigurowanie i administrowanie komunikacyjnym aspektem systemu. Zawarto w nim następujące funkcje:

- określenie usługi mającej służyć do przekazywania wskazanego typu wiadomości (trwałej wiadomości osobistej, nietrwałej wiadomości osobistej, wiadomości otwartej, powiadomienia),
- weryfikacja skuteczności dostarczenia wiadomości z wykorzystaniem wskazanej usługi komunikacyjnej i wskazanego odbiorcy (o ile dana usługa to umożliwia),
- anulowanie wskazanej wiadomości wysłanej za pośrednictwem systemu (o ile wykorzystana do tego usługa to umożliwia).

Trzecim podobszarem funkcjonalnym modułu administratora jest zarządzanie szablonami zawierające następujące funkcje:

- wprowadzenie nowego szablonu wskazanego typu (ustawień graficznego interfejsu użytkownika, formatowania danych, renderowania zasobów),
- aktualizacja definicji istniejącego szablonu wskazanego typu (ustawień graficznego interfejsu użytkownika, formatowania danych, renderowania zasobów),

- pobranie zdefiniowanego szablonu wskazanego typu (ustawień graficznego interfejsu użytkownika, formatowania danych, renderowania zasobów).

Moduł instruktora

Moduł instruktora obejmuje pięć podobszarów funkcjonalnych (rysunek 1). Pierwszym z nich jest zarządzanie zasobami, przeznaczony do obsługi dowolnego typu zasobów (materiałów wykładowych, testów, ćwiczeń i opartych na skryptach lekcji interaktywnych). Do tego podobszaru przypisano następujące funkcje:

- wprowadzenie do systemu nowego zasobu,
- aktualizacja zasobu istniejącego w systemie,
- pobranie zasobu z systemu (w oryginalnej postaci),
- edycja metadanych opisujących zasób,
- zmiana zasad dostępu do wskazanego zasobu,
- wyszukiwanie zasobu w systemie (w oparciu o podane metadane),
- edycja zasobu (o ile dostępne jest stosowne narzędzie, wbudowane lub zainstalowane jako rozszerzenie).

Drugim podobszarem modułu instruktora jest zarządzanie kursem. Obejmuje on takie funkcje, jak:

- tworzenie nowego kursu,
- przypisywanie zasobów do wskazanego kursu,
- usuwanie zasobów z kursu,
- pobieranie informacji dotyczących wskazanego kursu,
- modyfikacja metadanych dotyczących kursu,
- modyfikacja reguł gamifikacji przypisanych do kursu,
- modyfikacja reguł oceny ćwiczeń przypisanych do kursu,
- modyfikacja zasad końcowego oceniania przypisanych do kursu,
- eksport pełnej definicji kursu wraz z zasobami do pliku,
- import pełnej definicji kursu wraz z zasobami z pliku.

Kolejnym podobszarem modułu instruktora jest zarządzanie grupami. Grupa ma kluczowe znaczenie w przyjętej koncepcji LMS jako obiekt nie tylko łączący kursy (treści nauczania) i uczestników (odbiorców treści), ale także określający przebieg nauki (terminy) i zasady oceniania. Podobszar ten obejmuje następujące funkcje:

- definiowanie nowej grupy kursowej,
- duplikowanie istniejącej grupy kursowej,
- przypisanie uczestników do istniejącej grupy kursowej,
- usuwanie uczestników z istniejącej grupy kursowej,
- przyporządkowanie kursu do grupy kursowej,
- definiowanie harmonogramu kursu,
- modyfikacja reguł gamifikacji przypisanych do grupy kursowej,

- modyfikacja reguł oceny ćwiczeń przypisanych do grupy kursowej,
- modyfikacja zasad końcowego oceniania przypisanych do grupy kursowej,
- modyfikacja zasad komunikacji dla grupy kursowej,
- pobieranie informacji dotyczących wskazanej grupy kursowej.

Czwartym podobszarem modułu instruktora jest zarządzanie zawodami. Przez zawody rozumie się tu indywidualną lub grupową rywalizację między uczestnikami kursu, o określonych zasadach, celach i terminach, gdzie absolutny poziom opanowania wiedzy i umiejętności oceniany jest jedynie pośrednio, a bezpośrednio – poziom relatywny względem innych uczestników kursu. Podobszar ten obejmuje poniższe funkcje:

- zdefiniowanie nowych zawodów,
- przypisanie zasobów do istniejących zawodów,
- określenie zasad oceny obowiązujących na zawodach,
- przypisanie grup kursowych do zdefiniowanych zawodów,
- pobranie informacji o zawodach,
- rozesłanie zaproszeń do uczestnictwa w zawodach,
- otwarcie zawodów,
- weryfikacja otwarcia zawodów,
- zbieranie zgłoszeń nadsyłanych w ramach zawodów,
- zamknięcie zawodów,
- ocenianie zgłoszeń nadesłanych w ramach zawodów,
- rozesłanie informacji o zwycięzcach zawodów.

Ostatnim podobszarem omawianego modułu jest ocenianie rozwiązań. Odnosi się on do wprowadzania i zatwierdzania wyników oceny rozwiązań zgłaszanych przez uczestników kursu, jak również wystawiania uczestnikom kursu ocen końcowych, podsumowujących cały kurs lub jego istotną część. Podobszar ten obejmuje następujące funkcje:

- wprowadzenie wyników ręcznej oceny,
- pobranie wyników oceny (ręcznej lub automatycznej),
- wprowadzenie ocen końcowych ustalanych ręcznie,
- zatwierdzenie ocen końcowych obliczanych automatycznie, według reguł zdefiniowanych dla kursu lub grupy kursowej.

Warstwa prezentacji

Warstwa prezentacji to ostatni z pięciu obszarów funkcjonalnych architektury zgamifikowanego LMS. Przypisano do niej komponenty systemu funkcjonujące głównie po stronie klienckiej systemu, a zatem te, z którymi bezpośrednią styczność mają jego użytkownicy końcowi. Na warstwę prezentacji składa się pięć podobszarów funkcjonalnych (patrz rysunek 1). Pierwszym z nich jest interfejs użytkownika, który obejmuje następujące funkcje:

- wyświetlanie ekranu użytkownika zgodnie z wybranym szablonem (określającym jego rozmiar, układ, kolorystykę, czcionki),

- wyświetlanie wskazanego komponentu interfejsu użytkownika (jak: menu, okno dialogowe, przycisk, pole wyboru) zgodnie z wybranym szablonem,
- wywołanie funkcji systemu przypisanej do danego komponentu interfejsu użytkownika (wraz z przekazaniem informacji o kontekście użycia tego komponentu).

Drugim podobszarem warstwy prezentacji jest renderowanie zasobów, do którego przypisane są takie funkcje, jak:

- renderowanie zasobów statycznych, które generuje kod HTML pozwalający na prezentację takich zasobów zgodnie z wybranym szablonem,
- renderowanie zasobów dynamicznych, generujące kod HTML przygotowujący tło, które zostanie wypełnione zawartością dopiero w wyniku działania skryptów uruchamianych po stronie klienta.

Trzecim podobszarem warstwy prezentacji jest środowisko skryptowe po stronie klienta. Obejmuje on następujące funkcje:

- wywołanie skryptów odpowiadających za prezentację zasobów dynamicznych (wprowadzających animację i/lub interaktywność),
- wywołanie skryptów odpowiadających za gamifikację na poziomie mikroaktywności uczestnika kursu (w szczególności rejestrujących wykonywane przez niego proste czynności i generujące natychmiastową informację zwrotną, a także dokonujących agregacji danych o takich czynnościach w celu przekazania skróconej informacji na serwer).

Kolejnym podobszarem warstwy prezentacji jest synchronizacja stanu klienta i serwera obejmująca następujące funkcje:

- zabezpieczenie na serwerze aktualnego stanu środowiska skryptowego po stronie klienta,
- odtworzenie stanu środowiska skryptowego po stronie klienta z kopii zabezpieczonej na serwerze,
- przekazanie na serwer istotnej informacji o postępie uczestnika kursu,
- przekazanie z serwera do środowiska skryptowego po stronie klienta istotnej informacji o zmianie stanu kursu lub zawodów.

Ostatnim podobszarem warstwy prezentacji jest narzędzie oceniania treści. Jego celem jest pozyskanie *explicite* informacji zwrotnej dotyczącej oceny jakości kursu i jego komponentów od uczestników. Podobszar ten obejmuje następujące funkcje:

- renderowanie komponentów interfejsu użytkownika umożliwiających uczestnikom kursu ocenę jakości lekcji, ćwiczeń i informacji,
- zbieranie od uczestników kursu informacji zwrotnej dotyczącej oceny jakości lekcji, ćwiczeń i informacji.

Podsumowanie

Pomimo rosnącej świadomości zalet wykorzystania gamifikacji w nauczaniu, jej praktyczne stosowanie napotyka na bariery, z których jedną z najważniejszych jest nakład sił i środków

konieczny do przygotowania, prowadzenia i końcowego podsumowania kursu zgamifikowanego. Z pewnością bardzo pomocna w tym kontekście byłaby dostępność systemu zarządzania nauczaniem oryginalnie przygotowanego do obsługi takich właśnie kursów. Oczywiście wyzwaniem stanowi określenie struktury funkcjonalnej takiego systemu, która nie tylko spełniałaby klasyczne wymagania dotyczące oprogramowania klasy LMS, ale także odpowiadałaby specyficznym potrzebom związanym z gamifikacją.

Wyzwanie to podjęto w niniejszym artykule, zamieszczając w nim zhierarchizowaną listę funkcji uznanych za niezbędne dla zgamifikowanego systemu zarządzania nauczaniem opartego na architekturze zaproponowanej we wcześniejszej pracy autora (Swacha, 2014). Zaproponowane rozwiązanie może posłużyć praktykom jako wzorzec dla implementacji systemu o postulowanych walorach. Dla teoretyków stanowić może punkt odniesienia, zarówno dla weryfikacji zgodności przyszłych zgamifikowanych LMS z przedstawionymi tu rozwiązaniami i ułatwienia ewentualnych porównań między nimi, jak i dla wyprowadzenia nowych, bardziej wyrafinowanych lub, przeciwnie, skromniejszych propozycji struktury funkcjonalnej takich systemów.

Literatura

- Avgeriou, P., Retalis, S., Skordalakis, M. (2003). An Architecture for Open Learning Management Systems. W: Y. Manolopoulos, S. Evripidou, A.C. Kakas (red.), *Advances in Informatics* (s. 183–200). Heidelberg: Springer. DOI: 10.1007/3-540-38076-0_13.
- Hamari, J., Koivisto, J., Sarsa, H. (2014). Does Gamification Work? – A Literature Review of Empirical Studies on Gamification. W: *Proceedings of the 47th Hawaii International Conference on System Sciences* (s. 3025–3034). Waikoloa: IEEE. DOI: 10.1109/HICSS.2014.377.
- Kapp, K.M. (2012). *The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education*. San Francisco: Pfeiffer.
- Marczewski, A. (2014). *Defining gamification – what do people really think?*. Pobrano z: <http://www.gamified.uk/2014/04/16/defining-gamification-people-really-think/> (9.12.2015).
- Muntean, C.I. (2011). Raising engagement in e-learning through gamification. W: *Proceedings of the 6th International Conference on Virtual Learning* (s. 323–329). Cluj-Napoca: ICVL.
- Swacha, J. (2014). An Architecture of a Gamified Learning Management System. W: Y. Cao, T. Väljataga, J.K.T. Tang, H. Leung, M. Laanpere (red.), *New Horizons in Web Based Learning* (s. 195–203). Cham: Springer. DOI: 10.1007/978-3-319-13296-9_22.
- Swacha J. (2015). Gamifikacja w nauczaniu programowania: przesłanki i dostępne rozwiązania. W: P. Kosiuczenko, M. Śmiałek, J. Swacha (red.), *Od procesów do oprogramowania – badania i praktyka* (s. 15–26). Warszawa: Polskie Towarzystwo Informatyczne.
- Swacha J., Baszuro P. (2013). Gamification-based e-learning platform for computer programming education. W: N. Reynolds, M. Webb (red.), *Learning whileweareconnected. Vol. 1: Research papers* (s. 122–130). Toruń: Wydawnictwo Naukowe UMK.
- Werbach K., Hunter D. (2012), *For the Win: How Game Thinking Can Revolutionize Your Business*. Philadelphia: Wharton Digital Press.

THE FUNCTIONAL STRUCTURE OF A GAMIFIED LEARNING MANAGEMENT SYSTEM

KEYWORDS | IT in education, gamification, Learning Management System, LMS

ABSTRACT | Gamification, as a form of inducing intrinsic motivation by using elements characteristic for games, is gaining more and more interest in educational institutions and among teachers. Notwithstanding the increasing number of available gamification textbooks, its practical implementation still requires substantial effort on instructor's behalf. Although the popular Learning Management Systems already provide modules allowing to include gamification elements within the courses, making use of them is labor-intensive, and the range of provided features is very limited. The solution is to use a Learning Management System not only having a wide range of gamification features embedded already at its design stage, but also suitable for easy use by course designers and teachers. This paper proposes a detailed vision of the functional structure of such a system.

