

Kierowanie procesami integracji środowiska symulacji procesów projektowania węzłów internetu rzeczy

Integration driven simulation environment for designing Internet of Things nodes

CEZARY ORŁOWSKI
MARIUSZ WĄSIK
PIOTR WELFLER
MACIEJ KACPERSKI*

DOI: <https://doi.org/10.17814/mechanik.2018.12.208>

Zaprezentowano możliwości wykorzystania środowisk symulacji do projektowania węzłów internetu rzeczy. Najpierw scharakteryzowano problematykę budowy i zastosowania urządzeń internetu rzeczy oraz konstrukcji środowisk symulacyjnych. Następnie przedstawiono eksperyment badawczy projektowania węzła internetu rzeczy do pomiaru pyłu. Oceniono przydatność środowisk symulacyjnych do wspierania procesów projektowych.

SŁOWA KLUCZOWE: internet rzeczy, węzły IoT, środowiska symulacyjne, technologie *open source*

The article presents the possibilities of using simulation environments for designing Internet of Things nodes. At the beginning, the issues of the construction and use of Internet of Things devices and the construction of simulation environments were characterized. Next, a research experiment on designing the IoT node for dust measurement was presented. The work is summarized by the evaluation of the usefulness of simulation environments to support design processes.

KEYWORDS: *Internet of Things, IoT nodes, simulation environments, open source*

Rośnie zainteresowanie tematem internetu rzeczy (Internet of Things – IoT). Internet rzeczy to połączenie ze sobą urządzeń wyposażonych w pamięć i niewielką moc obliczeniową oraz oprogramowanie pozwalające na gromadzenie danych, ich przetwarzanie i przesyłanie pomiędzy urządzeniami [1]. Coraz większa dostępność takich urządzeń wymusza stosowanie odpowiednich metod projektowania.

Autorzy artykułu przyjęli takie podejście przed przystąpieniem do budowy węzła IoT. Sformułowali następujące pytania:

- Czy możliwe jest projektowanie systemów bez stosowania metod symulacyjnych?
- Na ile znane metody symulacji są adekwatne do projektowania systemów IoT?
- Czy zastosowanie technologii *open source*, pozwalających na dowolną integrację narzędzi symulacyjnych, jest odpowiednim środowiskiem do symulacji procesu projektowania urządzeń IoT?

Internet rzeczy a procesy symulacji

Współcześnie internet rzeczy otwiera nowe możliwości projektowania i budowy systemów [2]. Systemów, w których wszystko lub prawie wszystko można połączyć. Dzięki zastosowaniu dowolnych protokołów komunikacyjnych łączy się czujniki, stacje pomiarowe, siłowniki lub inne urządzenia. Uzyskane dzięki mnogości tych urządzeń informacje mogą być wykorzystywane w chmurze (*Cloud Computing*) lub na krawędzi (*Edge Computing*) [4]. Mogą także zasilać dowolne systemy, np. usługowe (*software as a service – SaaS*) [5].

Na podstawie [6] przyjmuje się, że projektowanie złożonego środowiska IoT ze względu na jego heterogeniczność i stosowanie wielu zróżnicowanych scenariuszy może być trudne. Dlatego niezbędna staje się symulacja procesu projektowania – zarówno z punktu widzenia ilościowego (analizy scenariuszy), jak i jakościowego (oceny dostępności).

Do symulacji złożonych systemów stosuje się odpowiednie metody i narzędzia. Rozważa się np. przyjęcie podejścia adaptacyjnego z wykorzystaniem symulacji równoległej i rozproszonej, symulacji wielopoziomowej oraz – w przypadku złożonych systemów – symulacji opartej na agentach [7].

W symulowaniu procesu projektowania systemów IoT mogą znaleźć zastosowanie metody typu *discrete event simulation* (DES) [8]. Stwarzają one warunki do analizy ewolucji systemu, oceny zdarzeń i odniesienia tych zdarzeń do stanu systemu.

Rozważa się stosowanie symulacji bazujących na modelach równoległych *parallel discrete event simulation* (PDES) [1]. Wówczas złożone modele mogą być reprezentowane przez odpowiednie części modelu symulacyjnego.

Można także stosować podejście adaptacyjne PADS (*parallel and distributed simulation*), które zostało użyte w symulatorze GAIA/ARTIS [9] z wykorzystaniem synchronizacji czasowej. Zastosowano tam metodę wielopoziomowego modelowania do symulowania modeli Yuki [3].

Wykorzystanie procesów ciągłej integracji jako metoda symulacji wbudowanych systemów

Symulacja procesów wytwarzania oprogramowania dla systemów IoT jest możliwa dzięki ciągłej integracji procesów, usług i urządzeń. Integracja ta może być problemem

* Prof. dr hab. inż. Cezary Orłowski (corlowski@wsb.gda.pl) – Katedra Zastosowań Informatyki w Zarządzaniu, WSB; lic. Mariusz Wąsik (o.mariuszwasik@gmail.com) – inżynier systemowy, UNISOFT Sp. z o.o.; lic. Piotr Welfler (piotr.welfler@outlook.com) – kierownik zaopatrzenia, Staples Polska Sp. z o.o.; inż. Maciej Kacperski (kacperski.o@gmail.com) – service-desk, Atena SA

zarówno w uruchamianiu kodu w systemie wbudowanym, jak też w konstrukcji węzłów IoT.

Zapotrzebowanie na środowiska symulacji, które stwarzają dogodne warunki dla prawdziwej automatyzacji i ciągłej integracji, jest duże. Doświadczenie pokazuje, że można to osiągnąć za pomocą szybkich platform wirtualnych wraz z modelami sieci symulatorów, w których współdziała wbudowany system.

Alternatywnym rozwiązaniem dla symulacji jest konfigurowanie sprzętu. Standardowa procedura konfiguracji systemu sprowadza się do konfiguracji mikrokontrolera oraz analizy danych testowych uzyskanych za pośrednictwem czujników pomiarowych podłączonych do tego systemu [10]. Z kolei, aby właściwie przetestować wbudowane oprogramowanie, konieczne jest wygenerowanie realistycznych danych na potrzeby tych systemów.

Dane te są zwykle generowane przez modele działające w czasie rzeczywistym. Konfiguracja sprzętu jest kosztowna i tylko nieliczne firmy są w stanie przeprowadzić takie testy. Stosunkowo często stosuje się rozwiązanie polegające na opracowaniu symulatora wykorzystującego interfejs API (*application programming interface*) [11].

Kolejnym podejściem jest uruchomienie oprogramowania bez API. Ma to miejsce wtedy, kiedy się zakłada, że docelowe oprogramowanie jest w znacznej mierze niezależne od modułów systemu docelowego. Wiele firm rozwija podejście łączące elementy obu opisanych metod [2].

Do stworzenia takiego środowiska wymagane jest, aby wirtualna platforma dokładnie odzwierciedlała warunki pracy systemu rzeczywistego. Warunki te z kolei są niezbędne dla wytwarzanego oprogramowania i dla podstawowych instrukcji pozwalających na symulowanie rejestrów urządzeń, map pamięci, przerwań i funkcjonalności urządzeń peryferyjnych.

Metoda projektowania węzłów internetu rzeczy

Na pierwszym etapie formułuje się wymagania w stosunku do procesu projektowania węzła IoT. Jeżeli się przyjmie, że topologia sieci i systemów składa się z prostych węzłów, które zbierają i przesyłają ograniczoną liczbę danych, to stawianie takich wymagań jest łatwe. Jednak w ramach wymagań wobec węzłów należy określić typową architekturę sieci i węzła IoT oraz rodzaj mikrokontrolera, który przetwarza dane i uruchamia pakiety oprogramowania połączone z urządzeniem bezprzewodowym.

Kolejnym etapem jest określenie liczby danych i sposobu, w jaki będą one przesyłane pomiędzy poszczególnymi węzłami. Jeżeli się przyjmie, że urządzenia komunikują się za pośrednictwem sieci, należy wziąć pod uwagę sposób agregacji danych z czujników i wyświetlania informacji. Trzeba także uwzględnić wybór odpowiedniego mikrokontrolera, modułu łączności bezprzewodowej, a także narzędzi programistycznych wykorzystywanych w procesie projektowania mikrokontrolera i urządzeń do transmisji bezprzewodowej.

Kolejnym etapem po określeniu wymagań jest wybór mikrokontrolera. Jest on w dużej mierze konsekwencją funkcji, jaką mają pełnić węzeł IoT i sieć IoT. W przypadku zaawansowanych węzłów, projektowanych za pomocą inteligentnych systemów, odpowiednim wyborem jest jednostka 32-bitowa.

Następny jest wybór łączności bezprzewodowej. Należy tu wziąć pod uwagę koszty projektowania sieci. Projektanci sieci bezprzewodowych muszą także ustalić, czy

jest to rozwiązanie, które najlepiej spełnia potrzeby danej aplikacji.

Jeżeli założą się wysoki standard sieci bezprzewodowych 2,40 Ghz, to można zapewnić wysoką przepływność danych. Należy jednak przyjąć, że transfer na częstotliwości 2,40 GHz ma ograniczony zasięg, słabą penetrację ścian i zwiększa zużycie energii.

Na kolejnym etapie następuje uproszczenie projektu dzięki integracji komponentów. Do tej pory węzeł IoT był omawiany jako suma elementów składowych: mikrokontrolera, czujników pomiarowych oraz urządzeń zasilających. Można jednak potraktować ten węzeł jako jedno urządzenie zamiast używania dyskretnych komponentów.

Przy tworzeniu oprogramowania, które umożliwia integrację urządzeń, można korzystać z urządzeń zintegrowanych. Proces projektowania będzie prostszy po uwzględnieniu wymogów urządzeń o małej mocy.

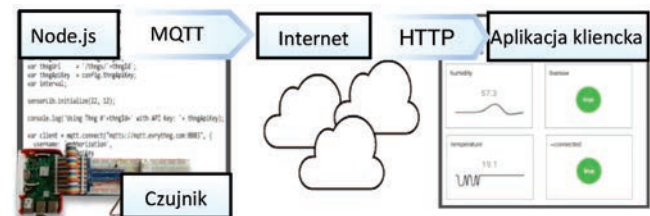
Przykładem takiego urządzenia może być zintegrowany bezprzewodowy mikrokontroler EZR32 z Peryferyjnym Układem Refleks (Peripheral Reflex System), który pozwala urządzeniu peryferyjnym współdziałać ze sobą bez budzenia jednostki centralnej [8].

Eksperyment badawczy – projektowanie węzła IoT

W przeprowadzonym przez autorów eksperymencie badawczym budowa środowiska do symulacji procesów projektowania węzła IoT była realizowana w pięciu etapach.

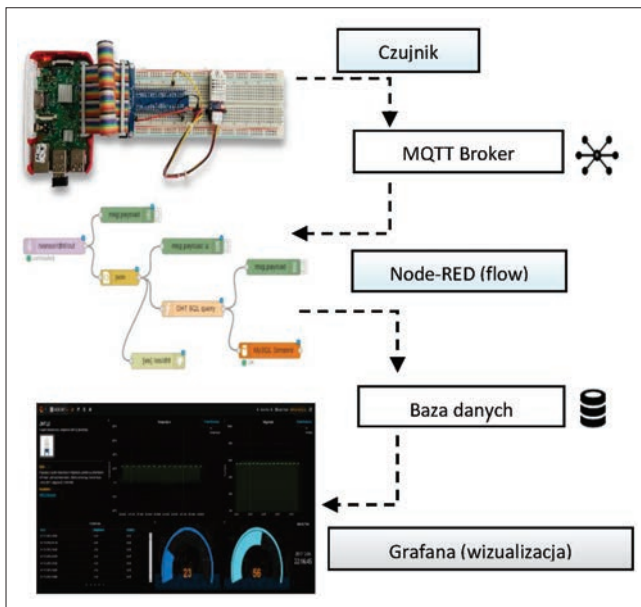
Pierwszy etap obejmował analizę dostępnych technologii na potrzeby stworzenia środowiska symulacyjnego – wizualizacji i sprawdzenia poprawności integracji czujników z Raspberry Pi. Rozważano możliwość wykorzystania technologii projektowych bazujących na systemach Linux i Windows. W pierwszym kroku zastosowano technologie pracujące w systemie operacyjnym Linux Raspbian opartym na dystrybucji Debian.

W fazie drugiej zaprojektowano środowisko symulacyjne. Najpierw wykorzystano technologię umożliwiającą publikację i subskrypcję danych za pomocą lekkiego protokołu transmisji danych MQTT (*message queue telemetry transport*). Na potrzeby tego środowiska przygotowano aplikację napisaną w środowisku Node.js [12]. Następnie aplikację rozszerzono o funkcjonalność łączenia się z usługami stworzonymi na potrzeby internetu rzeczy, takimi jak: <https://thingspeak.com/>, <https://opensensors.com/>, <http://evrythng.com/>. Usługi te służą do przechowywania, wizualizacji i analizy danych pochodzących z zewnętrznych źródeł, np. z czujników. Strukturę środowiska symulacyjnego przedstawiono na rys. 1.



Rys. 1. Struktura środowiska symulacyjnego

Następnie dobrano środowisko do tworzenia aplikacji obsługującej projektowany węzeł – Node RED. Jest to narzędzie programistyczne do łączenia urządzeń sprzętowych, interfejsów API i usług online. Node RED zawiera wiele predefiniowanych modułów wejścia i wyjścia, dzięki którym dalsze projektowanie węzła staje się dużo łatwiejsze.



Rys. 2. Przepływ danych i ich wizualizacja przy podłączeniu czujnika wilgotności (czujnik + MQTT + Node RED + Grafana)

Jednym z modułów wykorzystanych do projektowania przepływu danych był moduł bazy danych MySQL. Baza ta została użyta do rejestrowania wartości pobieranych z czujników. Ostatnim składnikiem środowiska symulacyjnego była technologia do wizualizacji danych.

Na trzecim etapie zastosowano środowisko symulacyjne opracowane do budowy węzła IoT. Wykorzystano siedem czujników pomiarowych podłączonych do Raspberry Pi połączonych poprzez port cyfrowy GPIO. Brano pod uwagę następujące typy czujników: czujnik poziomu cieczy (XKCY25T12V), czujnik odległości (HCRS04), czujnik ruchu (PIR-HCSR501), czujnik dźwięku (Wave-share), czujnik temperatury i wilgotności (DHT22), czujnik natężenia światła (BH1750) oraz czujnik ciśnienia (BMP280).

Podłączono poszczególne czujniki do Raspberry Pi i śledzono przepływ pochodzących z nich danych. Przepływ analizowano z wykorzystaniem środowiska Node Red, a wizualizację danych realizowano z wykorzystaniem technologii Grafana. Przykład działania przepływu danych, w którym podłączono czujnik temperatury i wilgotności, przedstawiono na rys. 2.

Na czwartym etapie wyniki z symulacji stanowiły podstawę do rekonstrukcji węzła IoT. Rozwiązanie to wymagało zastosowania dodatkowego urządzenia – modułu WiFi ESP826601 – niezbędnego do podłączenia Arduino do sieci komputerowej (efekt przeprowadzonej symulacji). Raspberry Pi z zainstalowanym serwerem MQTT wykorzystano jako serwer przechwytyjący dane – tzw. broker. Zastosowano różne typy mikrokontrolera Arduino – UNO, Nano, Pro mini.

Na piątym etapie na podstawie analizy przepływu danych oceniono możliwości modułu ESP826601. Zauważono, że ma on programowalny mikrokontroler, na tyle wydajny, że może z powodzeniem zastąpić Arduino. Ze względu na ograniczoną liczbę portów GPIO i ADC w module ESP826601 wybrano moduł ESP8266-12, którym ma na złączu dziewięć portów GPIO oraz jedną linię ADC. Aby potwierdzić poprawność działania węzła i przydatność dobranych technologii, ponownie zastosowano środowisko symulacyjne, które po kolejnej modyfikacji zostało oparte na maszynie wirtualnej z systemem Linux Ubuntu.

Podsumowanie

Celem artykułu była prezentacja środowiska symulacji procesów projektowania węzłów IoT kierowanej procesami integracji technologii, obejmującej środowiska informacyjne oraz podzespoły techniczne węzła, takie jak mikrokontrolery i czujniki pomiarowe. Zaprezentowano metodę i własne środowisko symulacji procesów projektowania węzła w przeprowadzonym eksperymencie badawczym. Dobierano zarówno środowiska programistyczne, czujniki pomiarowe, jak i mikrokontrolery. Wykazano znaczenie procesów symulacji dla doboru technologii, ich późniejszej integracji, a także dla doboru czujników oraz mikrokontrolerów.

Zaprezentowany przykład, w którym do mikrokontrolera Raspberry Pi podłączono czujniki poziomu cieczy, odległości, ruchu, dźwięku, temperatury i wilgotności, wykazał znaczenie zintegrowanego środowiska symulacyjnego do wizualizacji danych dla lekkiego protokołu MQTT, środowiska do wizualizacji przepływów Node RED i prezentacji danych Grafana. Dzięki zastosowaniu tego środowiska poddano rekonfiguracji węzeł. Zmieniono mikrokontroler w celu optymalizacji procesu pozyskiwania danych i ich przetwarzania na potrzeby ochrony środowiska.

Praca stanowi przykład zaawansowanego projektu studentckiego (podobnie jak Arduino) węzła, który znajduje zastosowanie w budowie węzłów IoT na potrzeby ochrony środowiska. Autorzy dziękują Wyższej Szkole Bankowej za sfinansowanie badań.

LITERATURA

1. Al-Fuqaha A., Guizani M., Mohammadi M., Aledhari M., Ayyash M. "Internet of things: A survey on enabling technologies, protocols, and applications". *IEEE Communications Surveys Tutorials*. 17, 4 (2015): s. 2347–2376, Fourthquarter.
2. Engblom J. "Continuous integration for embedded systems using simulation". *Embedded World Exhibition & Conference*. Szwecja, 2015.
3. Jun Y., Racz C., Tan G. "Evaluation of a sort-based matching algorithm for DDM". *Proceedings of the 16th workshop on Parallel and distributed simulation, PADS*. Washington, DC, USA: IEEE Computer Society, 2002: s. 68–75.
4. Loper M., Louchez A. "The Internet of Things and the Importance of Modeling and Simulation". Georgia Institute of Technology, <https://www.automationworld.com/article/industry-type/all/internet-things-and-importance-modeling-and-simulation>, 2015 (dostęp: 14.10.2018 r.).
5. Dirk S., Puhlmann F., Morrish F., Bhatnagar M. "Enterprise IoT Strategies & Best Practices for Connected Products & Services". O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2015.
6. Law A.M., Kelton D.W. "Simulation Modeling and Analysis". 3rd edition. McGraw-Hill Higher Education, 1999.
7. Luke S., Cioffi-Revilla C., Panait L. "A multiagent simulation environment". *Simulation*. 81, 7 (2005): s. 517–527.
8. D'Angelo G., Ferretti S., Ghini V. "Simulation of the Internet of Things". *Proceedings of the IEEE International Conference on High Performance Computing and Simulation (HPCS)*, 2016.
9. D'Angelo G., Ferretti S., Ghini V. "Simulation of the Internet of Things". *International Conference on High Performance Computing & Simulation (HPCS)*, 2016. DOI: 10.1109/HPCSim.2016.7568309.
10. Kowalczyk Z., Orłowski C. "Advanced Modeling of Management Processes in Information Technology". Springer, ISBN 978-3-642-40876-2, 2014.
11. Orłowski C., Kowalczyk Z. "Modelowanie procesów zarządzania technologiami informatycznymi". PWNT, 2012.
12. Guinard D.D., Trifa V.M. "Internet rzeczy". Polska: Helion, 2017. ■