

Wysokopoziomowy system adresowania urządzeń inteligentnego budynku, pracujący z wykorzystaniem ontologii i Internetu Rzeczy

Jarogniew Rykowski

1. Wprowadzenie

Od początku dziejów ludzie chcieli żyć i pracować w komfortowych warunkach. Pojęcie komfortu ewoluowało w miarę wzrostu możliwości i zapotrzebowania na najnowsze zdobycze nauki i techniki. Ewolucja ta nabrała szczególnego tempa w ostatnich latach, co jest związane z powszechną elektronizacją, a od kilku lat – cyfryzacją naszego życia. Nowe rodzaje coraz bardziej miniaturowych, energooszczędnych i funkcjonalnych urządzeń zmieniają świat i rodzą apetyt na jeszcze większe zmiany.

Komfortowe korzystanie z nowoczesnych urządzeń jest możliwe tylko wtedy, gdy możemy używać tych urządzeń w sposób intuicyjny. Większość ludzi nie akceptuje urządzeń (lub ich niektórych funkcji), z których korzystania musi się wcześniej nauczyć – czy to z lenistwa, czy też (najczęściej) po prostu nie potrafi ich zrozumieć.


Intuicyjne korzystanie z urządzeń oznacza, że interesuje nas efekt pracy, a nie samo urządzenie, które ten efekt wypracowuje. Na przykład sprawne ogrzewanie i wentylacja w pomieszczeniu zwiększają komfort przebywających w nim osób, ale niewiele osób chce zadawać sobie trud (1) wyszukania urządzenia, które jest za to odpowiedzialne, (2) nauczenia się jego obsługi, a wcześniej zrozumienia zasad jego działania, oraz (3) codziennego ustalania trybu pracy. Zamiast tego chcielibyśmy powiedzieć komuś (lub czemuś), że jest zbyt zimno lub przeszkadza nam hałas wentylatora. Intuicyjnie sterowany wentylator, wyposażony w analizator głosu, rozumie wypowiedziane słowa i, podobnie jak człowiek, odpowiednio (w zgodzie z naszymi wyobrażeniami) na nie zareaguje, zwiększając komfort przebywania w pomieszczeniu.

Należy zwrócić uwagę, że intuicyjne sterowanie zawsze stawia na głównym miejscu obiekt wyposażenia, który jest sterowany (lub jego funkcję), a nie elek-

Streszczenie: W pracy opisano nowy sposób wyszukiwania, adresowania i uaktywniania urządzeń inteligentnego budynku. W odróżnieniu od klasycznych metod adresowania, zakładających sztywne powiązanie urządzeń sterujących i sterowanych za pomocą jednoznacznych identyfikatorów (tzw. „parowanie” urządzeń), w przedstawianym podejściu wykorzystywane jest dynamiczne łączenie kontekstowe w połączeniu z modelowaniem ontologicznym. Model obejmuje możliwości urządzeń, które są modelowane nie na poziomie charakterystyki (danych technicznych, sposobu podłączenia, adresu itp.) urządzenia, a na podstawie funkcji, jakie dane urządzenie pełni. Model obejmuje też kontekst, na który składają się te parametry, które są niezależne od wywołania, np. aktualna lokalizacja urządzenia lub osoby, temperatura w pomieszczeniu itp. Ostatnim modelowanym elementem są żądania wywołania odpowiedniej akcji, czyli uaktywnienia urzą-

żeń. Zastosowane modele ontologiczne pozwalają na sformułowanie wymagań i zgłoszenie żądania, następnie dołączenie ograniczeń wynikających z kontekstu i ostatecznie dynamiczny wybór tych (grup) urządzeń, które na podstawie wcześniej zadeklarowanych przez siebie możliwości są w stanie te wymagania spełnić. Urządzenia te są następnie aktywowane, co skutkuje wykonaniem żądanej akcji.

Opisywane podejście zostało zaimplementowane w postaci środowiska SITE (ang. *Semantic Internet-of-Things Environment*). Środowisko to okazało się użyteczne w modelowaniu wzajemnych interakcji między użytkownikami i urządzeniami inteligentnego budynku, szczególnie w kontekście miejsc publicznych, takich jak muzea, szpitale, urzędy itp., gdzie klasyczne sterowanie wyposażeniem budynku przez incydentalnych użytkowników, z wykorzystaniem ich urządzeń personalnych, nie może być stosowane.

 **Abstract:** *The paper introduces a new way of searching for, addressing, and activating devices of an intelligent building. In contrast to classical methods of addressing, aiming in using hardcoded links among controlled and controlling devices by means of unique identifiers (so called “pairing” of devices), the approach is based on dynamic contextual bonding and uses ontological descriptions of devices. The proposed model covers definitions of device possibilities, at the level of the relations among devices and real-world artifacts they are related with rather than at the level of technical (or similar) description of the devices. Possibilities reflect the functions one may observe for each device. The possibilities are automatically enhanced by contextual information (such as current localization of a device/human, communication link, etc.), and compared at run-time with requests coming from humans and other devices.*

Applied ontological models are capable of: (1) formulating device possibilities and (2) incoming requests, (3) filtering of a target device set based on contextual information, and (3) final activation of the selected device(s) that may fulfill the requests based on their predefined possibilities. Activated devices perform certain actions, forming a response to the served request.

The approach has been implemented as SITE (Semantic Internet-of-Things Environment) framework. This framework was found useful in modeling mutual interactions among humans and intelligent-building devices, especially at public spaces such as a museum, a hospital, an office, etc. In such case, classical controllers cannot be used due to incidental (ad-hoc) interaction and no possibility to apply personal devices such as a mobile phone.

troniczny sterownik. Jest to w sprzeczności z powszechnie przyjętą zasadą sterowania urządzeniami inteligentnej instalacji, gdzie sterownik – jego ekran i klawiatura, są tym, na czym należy skupić uwagę, a efekty sterowania uzyskuje się niejako mimochodem. Jednak trudno powiedzieć, że sterowanie oświetleniem za pomocą wielopoziomowego menu z mikrokomputera wiszącego na ścianie jest intuicyjne – jeśli nie jest intuicyjne, nie jest też komfortowe, mimo że w założeniu miało ten komfort polepszyć (mówimy o komforcie korzystania z systemu, a nie tylko wynikającym z efektów pracy instalacji). Jednak wracając do przykładu inteligentnej wentylacji – taki wentylator nie jest widoczny, nie ma też klawiatury i ekranu do sterowania, a jego obsługa sprowadza się do wypowiedzenia polecenia, czyli jest jak najbardziej intuicyjna i przez to komfortowa dla użytkownika.

Intuicyjność kontaktu z urządzeniem powoduje, że w sposób naturalny zaczynamy traktować urządzenie jako dodatkowego domownika i tak też chcemy się do niego zwracać, podświadomie je personalizując lub co najmniej traktując je jak domowe zwierzę. Zatem spodziewamy się, że „inteligencja” wyłącznika światła będzie na poziomie na przykład psa – wyłącznik taki będzie reagował na polecenia głosowe, niesformalizowane, z drugiej strony będzie cierpliwie czekał, aż każemy mu wypełnić swoje obowiązki, chyba że się zepsuje („zachoruje”) i będzie wymagał naprawy („leczenia”), ale wtedy musi „porozmawiać” z administratorem („lekarzem”), wyspecjalizowanym w radzeniu sobie w takich sytuacjach.

Intuicyjne wykorzystanie urządzeń inteligentnej instalacji wymaga zupełnie nowego sposobu wyszukiwania, adresowania i uaktywniania tych urządzeń w ramach sieci inteligentnego budynku. W niniejszym artykule zaproponowano zintegrowane podejście do rozwiązania tych problemów. Główny cel artykułu to zaproponowanie nowego sposobu adresowania i wyszukiwania urządzeń inteligentnej instalacji na takim poziomie, że można korzystać z ich usług (uaktywniać je) bez przeszkolenia, tylko z wykorzystaniem intuicji oraz naturalnych interfejsów typu głos i obraz. Zaproponowane środowisko do implementacji swych funkcji wykorzystuje połączenie trzech idei: Internetu Rzeczy, ontologii oraz wywołań kontekstowych.

Dalsza struktura artykułu jest następująca. W drugim rozdziale przedstawiono podstawowe pojęcia związane z Internetem Rzeczy i cichym przetwarzaniem danych. W trzecim rozdziale przedstawiono krytykę obecnych systemów adresowania urządzeń (węzłów) sieci. Następny rozdział jest poświęcony opisowi architektury środowiska SITE, która jest podstawą implementacji intuicyjnego adresowania urządzeń inteligentnego budynku, oraz bazowym aspektem modelowania ontologicznego i kontekstowego wykorzystywanego do implementacji tego środowiska. Rozdział piąty zawiera wnioski i wskazuje kierunki dalszych prac.

2. Internet Rzeczy i ciche przetwarzanie danych

Znany pisarz *science-fiction* Arthur C. Clarke, któremu przypisuje się między innymi wymyślenie łączności komórkowej i satelitów telekomunikacyjnych, powiedział kiedyś, że „każda dostatecznie rozwinięta technologia jest nieodróżnialna od magii” [1]. Jeśli przeanalizujemy postęp, jaki się dokonał (i nadal dokonuje) w ostatnim półwieczu w dziedzinie nowoczesnych technologii informatycznych i telekomunikacyjnych (ICT – ang.

Information and Communication Technologies), to trudno się z tym zdaniem nie zgodzić. Magia, czy jak kto woli postęp, rządzi się pewnymi prawami, które mogą powodować wykluczenie tych, którzy nie chcą lub nie mogą z „dobrodziejstw” tej magii korzystać. Z tego względu poniżej rozważymy wpływ, jaki ewolucja informatyczno-komunikacyjna wywiera na nas w naszym najbliższym otoczeniu, czyli jak „magicznie” to otoczenie oddziałuje na ludzi (i z wzajemnością, aczkolwiek otoczenie na pewno nie będzie traktować działalności ludzi jako magii).

Podstawowe zasady intuicyjnego sterowania urządzeniami podał znany (już niestety nieżyjący) wizjoner Mark Weiser, wprowadzając definicję „dobrego służącego” – usłużnego i zawsze gotowego, ale niewidocznego podczas oczekiwania na polecenie. Widzimy tylko skutki jego działania, ale sam „służący” stara się pozostać niezauważony i nie zwraca na siebie uwagi.

Według Weisera w historii rozwoju ICT można wyróżnić trzy zasadnicze fale komputeryzacji:

- duże komputery typu „mainframe”, centra obliczeniowe;
- komputery osobiste, kontakt przez GUI/pulpit;
- wszechobecne komputery lub inaczej systemy cichego przetwarzania danych (ang. *ubiquitous computing, calm computing*), ostatnio także nazywane Internetem Rzeczy.

Podczas pierwszej fali komputeryzacji użytkownik, żeby skorzystać z usług komputera, musiał niejako do niego przyjść (jak w znanym przysłowiu z Mahometem i górą). Komputery były na tyle duże i wymagały tak skomplikowanej infrastruktury, że niemożliwe było ich umieszczanie poza specjalnie w tym celu budowanymi pomieszczeniami. Także korzystanie z nich wymagało bardzo silnych narzędzi autoryzacyjnych, w celu limitowania dostępu wyłącznie dla osób uprawnionych. Podczas drugiej fali komputeryzacji komputery stały się znacznie powszechniejsze, a kontakt z nimi możliwy w wielu miejscach – nadal jednak do komputera trzeba było podejść, a także mu się „przedstawić” (nazwa użytkownika, hasło itp.) przez rozpoczęciem interakcji. Tę sytuację odwrócił dopiero Internet Rzeczy – komputery uzyskały możliwość przemieszczania się (na razie głównie biernego, razem ze swymi właścicielami lub użytkownikami, ale samodecydowanie o miejscu, w którym mobilne urządzenie będzie operować, jest już w wielu miejscach faktem), a także stały się na tyle małe, że to one zaczynają niejako przychodzić do użytkowników. Dodatkowo małe i mobilne urządzenia (bo trudno je nazywać klasycznie – komputerami, mimo że są wyposażone w procesor, pamięć i program sterujący) stają się niewidoczne aż do momentu, kiedy nie są komuś (lub czemuś) potrzebne – tak jak wspomniany „dobry służący”. Co więcej, urządzenia te mogą świadczyć usługi całkowicie nieświadomym użytkownikom, nie są też zwykle związane z konkretnymi osobami, a raczej z sytuacją lub miejscem, na terenie którego operują. Zatem świadomy i nieanonimowy użytkownik komputerów z I i II fali komputeryzacji staje się podczas III fali użytkownikiem nieświadomym i potencjalnie anonimowym, a interakcja między komputerem i człowiekiem zaczyna ewoluować od przemyślanej (i przewidywalnej z góry dla obu wzajemnie znających się stron) i nieprzypadkowej do incydentalnej (*ad-hoc*), przypadkowej i anonimowej (a przynajmniej niewymagającej autoryzacji i/lub identyfikacji). Cel, dla którego dane urządzenie istnieje, zaczyna wynikać z aktualnej sytuacji (kontekstu), niekoniecznie musi on być przewidziany od samego początku przez projektantów tego urządzenia.

Mark Weiser, który zajmował się badaniem opisanego powyżej zjawiska od końca lat 80. XX wieku, zainicjował badania na styku socjologii, filozofii, psychologii, psychofizjologii, badań związanych z kulturą i zwyczajami oraz technologii (mikro)komputerowych, zajmując się „niezauważalną technologią (komputerową) na usługach ludzi”. Badacz ten sformułował cztery zasady, znane obecnie jako Zasady Weisera [2]:

- głównym celem komputera jest pomóc człowiekowi w zrobieniu czegoś;
- najlepszy jest ten komputer, który zachowuje się jak dobry służący – jest pomocny, ale niekoniecznie widoczny;
- im więcej człowiek korzysta ze swojej intuicji, tym lepiej/szybciej/naturalniej dochodzi do celu; komputery powinny być wykorzystywane intuicyjnie, a nawet w miarę możliwości nieświadomie;
- technologia powinna wprowadzać uspokojenie, a nie zamieszanie.

W myśl powyższych zasad komputer (lub ogólniej – urządzenie komputerowe) staje się idealnym pomocnikiem człowieka w jego codziennych czynnościach. Mówiąc innymi słowami, technologia umożliwia masowe wspomaganie codziennych czynności za pomocą niezauważalnych urządzeń. Idea ta cieszyła się dużą popularnością już w końcu XX wieku, aczkolwiek ze względu na ograniczenia technologii jej pełne wykorzystanie nie było możliwe – prototypy systemów cichego przetwarzania w rodzaju Ambient Devices i Smart Planet [3] firmy IBM, „Myślących Rzeczy” (ang. *Things That Think*) uniwersytetu MIT [4] czy też urządzenia proponowane przez Alana Kaya i laboratorium Xerox/Apple [5] – tablet, tabliczka dotykowa, palmtop – dopiero teraz, kilkanaście lat później, zaczynają zdobywać rynek.

3. Tradycyjne metody adresowania i wyszukiwania urządzeń

W celu przesłania/odebrania danych do/z urządzenia podłączonego do Internetu, należy a priori znać adres (nazwę, identyfikator) tego urządzenia. W modelu OSI [6] węzły Internetu lokalizują się na podstawie:

- w warstwie aplikacji – nazw DNS lub NetBIOS; nazwy te są interpretowane przez usługę DNS (ang. *Domain Name Service*) i jej specjalizowane serwery usługowe, dostępne w sieci;
- w warstwie sieci – adresów IP oraz numerów gniazd (portów); datagramy IP są przekształcane w warstwie transportowej na pakiety TCP lub UDP i przekazywane do odpowiedniego portu adresowanego węzła sieci;
- w warstwie łącza danych – adresów MAC (ang. *Media Access Control*) – 48-bitowych adresów fizycznych karty (urządzenia) sieciowej; adresy MAC są nadawane na etapie produkcji i w zasadzie powinny być unikalne w ramach całego Internetu.

Za standaryzację adresowania DNS, TCP/IP oraz MAC odpowiadają (między innymi) dokumenty RFC:

- RFC 917 „Internet subnets”, 1984: standardy adresowania podsieci IP oraz routingu w podsieciach;
- RFC 952 „DoD Internet host table specification”: standardy nazywania i reprezentowania nazw komputerów;
- RFC 1166 [7] „Internet Numbers”, 1990: zasady adresowania IP (klasy i rezerwacje adresów);
- RFC 1918 „Address Allocation for Private Internets”: zasady adresowania sieci prywatnych i publicznych (ogólnodostępnych).

3.1. Adresowanie urządzeń w innych typach sieci

W sieciach automatyki przemysłowej adresowanie urządzeń może być zrealizowane na dwa podstawowe sposoby:

- wybór konkretnego urządzenia na zasadzie miejsca jego podłączenia (topologia gwiazdy i dedykowane połączenia między urządzeniami); z reguły takie adresowanie wymaga wykorzystania urządzeń pośredniczących, dołączonych do Internetu i przekazujących dane (ang. *routing*) do/z urządzeń końcowych;
- wykorzystanie architektury magistralowej i sprzętowy wybór adresu fizycznego urządzenia podłączonego do magistrali; adres urządzenia jest albo ustalany przez producenta (np. w technologii 1-Wire), wprogramowany na sztywno do sterownika (układu mikroprocesorowego sterującego urządzeniem) lub wybierany przez użytkownika końcowego np. za pomocą mikroprzełączników wewnątrz urządzenia).

Zatem podobnie jak w przypadku Internetu, także w przypadku sieci automatyki przemysłowej należy z góry znać adres urządzenia, żeby móc nawiązać z nim łączność.

3.2. Adresowanie rozplywowe (bezadresowe) z wykorzystaniem magistrali CAN

W opracowanej w latach 80. XX wieku magistrali CAN [8] zrezygnowano z bezpośredniego adresowania urządzeń dołączonych do tej magistrali. Cała transmisja w magistrali odbywa się rozplywowo, a zainteresowane danymi (współpracą) urządzenia same decydują na podstawie odebranych danych, czy informacje te są do nich skierowane. Magistrala CAN miała w zamyśle sterować pracą urządzeń pokładowych pojazdów i do dzisiaj jest głównie wykorzystywana w tym właśnie charakterze.

Mimo że w tym typie transmisji nie wykorzystuje się fizycznego adresowania węzłów sieci, a interpretacja transmisji odbywa się na zasadzie analizy napływającej informacji, nie jest to w pełni adresowanie semantyczne. Urządzenia magistrali CAN muszą mieć na sztywno zaprogramowane reakcje na określone typy nadsyłanych komunikatów, zatem mimo pozornie dynamicznego charakteru jest to ciągle adresowanie statyczne, a sam system jest stosunkowo mało elastyczny i rozszerzalny.

3.3. Semantyczny Internet i adresowanie jego zasobów

Semantyczny Internet (ang. *Semantic Web*), zainicjowany przez Tima Berners-Lee (twórcę systemu WWW), wykorzystuje dostępne technologie Internetu, w tym jego protokoły adresowe, do reprezentowania i przesyłania informacji w sposób zależny od jej semantyki (znaczenia). Semantyka danych jest reprezentowana przez tak zwane metadane, czyli dane opisujące inne dane [9]. Semantyka jest interpretowana w ramach odpowiedniego kontekstu, umożliwiając dostęp do danych, których lokalizacja a priori nie jest znana. Dzięki temu można powiązać w całość dane o takim samym lub podobnym znaczeniu, rozproszone w węzłach sieci, rozróżnić dane i obiekty, których identyfikatory są identyczne (np. słowa wieloznaczne), a także wnioskować o ukrytym znaczeniu danych.

Semantyczny Internet w założeniu adresuje informację, a nie węzły (urządzenia) sieci. Zatem mimo szerokiego wykorzystania ontologii i kontekstu, nadal nie jest możliwy bezadresowy kontakt z urządzeniami Internetu Rzeczy.

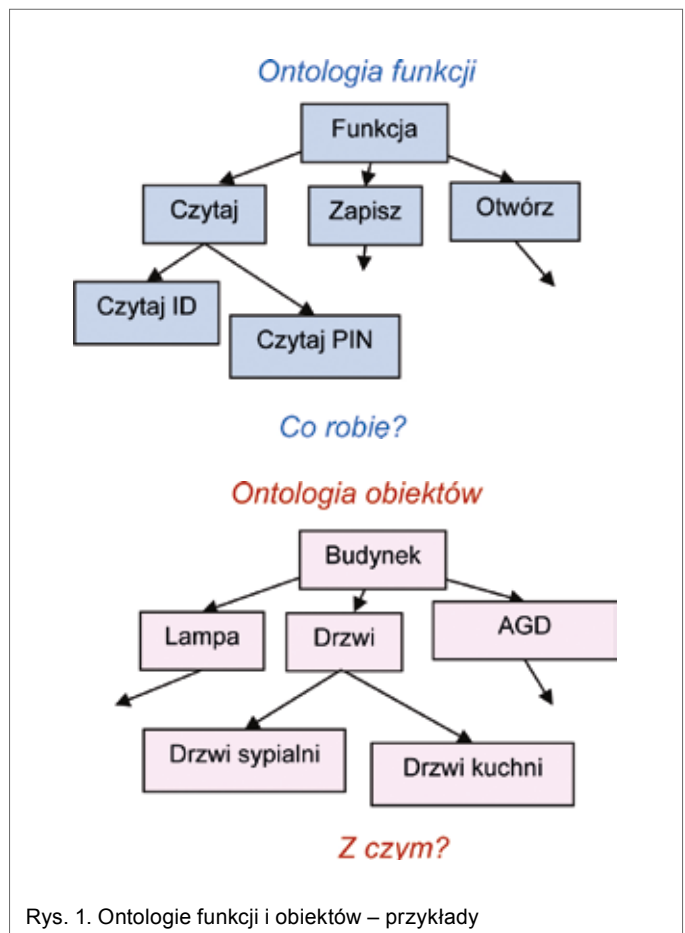
3.4. Adresowanie węzłów i zasobów sieci – ograniczenia aktualnie wykorzystywanych sposobów

Jak wskazano powyżej, praktycznie we wszystkich sieciach (wyjąwszy specjalizowane sieci przemysłowe) węzły są adresowane statycznie, za pomocą unikalnych, nadanych im wcześniej identyfikatorów. W celu zaadresowania danego urządzenia trzeba było z góry znać jego adres w systemie (sieci). Taki sposób adresowania uniemożliwia współpracę *ad-hoc* między urządzeniami, które się „nie znają” wzajemnie i a priori nic o sobie nie wiedzą, mimo że znajdują się np. w tej samej lokalizacji lub służą do podobnych (wzajemnie uzupełniających się, komplementarnych) celów. Celem niniejszej propozycji jest rozszerzenie idei przyświecających twórcom magistrali CAN oraz Semantycznego Internetu w kierunku dynamicznego, wykorzystującego opisy semantyczne adresowania urządzeń Internetu Rzeczy, co wydaje się bardzo obiecującym kierunkiem.

4. Opis rozwiązania

W nowym rozwiązaniu proponuje się wykorzystanie kontekstu oraz opisu ontologicznego do dynamicznego adresowania urządzeń. Rozwiązanie jest dedykowane w szczególności do współpracy *ad-hoc* miniaturowych urządzeń Internetu Rzeczy, pracujących w ramach inteligentnej instalacji np. budynku, biura lub fabryki. W rozwiązaniu ontologicznie modeluje się możliwości urządzeń – są to opisy działań, które urządzenia potrafią wykonać. Ontologicznie jest też modelowany kontekst, czyli sytuacja, w której aktualnie znajduje się dane urządzenie (położenie, warunki otoczenia – temperatura, ciśnienie itp.) oraz nadpływające do systemu żądania uaktywnienia urządzeń. Przez porównanie zgodności zapisów ontologii żądania, kontekstu wywołania i możliwości urządzeń wskazywanych przez ten kontekst można dynamicznie wybrać (zaadresować) urządzenie, które obsłuży to żądanie.

Podstawą reprezentacji kontekstu oraz wyboru urządzeń na podstawie żądań użytkowników jest odpowiednio skonstruowana ontologia modelująca z jednej strony możliwości (funkcjonalność) urządzeń, a z drugiej – nadchodzące żądania wraz z kontekstem ich wystąpienia. Jak wspomniano wcześniej, ontologia [10] oznacza w uogólnionym sensie systemowy opis „tego, co jest” lub „może być” w danym fragmencie rzeczywistości [11]. Modelowane możliwości dotyczą zarówno rzeczywistych, jak i wirtualnych urządzeń (urządzeń złożonych, czyli grup różnorodnych urządzeń traktowanych z punktu widzenia ich adresowania jako całość). Możliwości są modelowane nie na poziomie charakterystyki (danych technicznych, sposobu podłączenia, adresu itp.) urządzenia, a na podstawie funkcji, jakie dane urządzenie pełni w sieci budynku. Na przykład, elektrorzygiel drzwi będzie związany z funkcją otwierania tych drzwi, kontaktron we framudze okna – z funkcją detekcji faktu otwarcia tego okna, przekaźnik w lampie – z funkcją zaświecenia/zgaszenia światła w pomieszczeniu itp. Drugi brany pod uwagę element – kontekst, służy do modelowania sytuacji. Na kontekst składają się te parametry, które są niezależne od wywołania, np. lokalizacja urządzenia (dane GPS lub jednoznaczna nazwa, np. „kuchnia”), aktualne miejsce pobytu użytkownika, temperatura w pomieszczeniu itp. Ostatnim modelowanym ontologicznie elementem są wymagania, czyli postulaty (zgłaszane przez ludzi lub inne urządzenia) składające się na żądanie wywołania odpowiedniej akcji, na przykład otwarcia drzwi. Zastosowane modele ontologiczne pozwalają na sformułowanie wymagań i zgłoszenie



Rys. 1. Ontologie funkcji i obiektów – przykłady

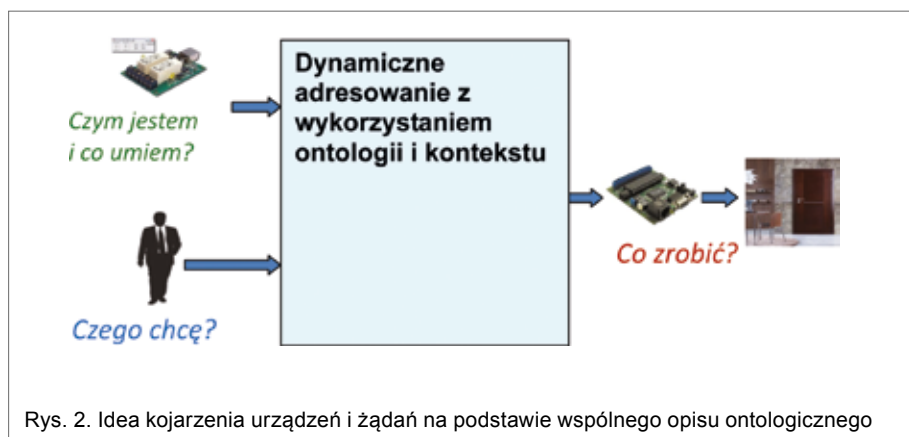
żądania, następnie dołączenie wymagań wynikających z kontekstu i ostatecznie dynamiczny wybór tych (grup) urządzeń, które, na podstawie wcześniej zadeklarowanych przez siebie możliwości, są w stanie te wymagania spełnić. Urządzenia te (jedno lub cała grupa) są następnie aktywowane, co skutkuje wykonaniem żądanej akcji.

Z punktu widzenia opisywanej w dalszej części tekstu implementacji ontologia jest reprezentowana w postaci dwóch grafów acyklicznych i skierowanych (hierarchii), odwziewiedlających (1) czynność, jaką potrafi lub ma wykonać dane urządzenie (tak zwana ontologia funkcji), oraz (2) obiekt świata rzeczywistego, nad którym ta czynność ma być wykonana (tak zwana ontologia obiektów – rys. 1).

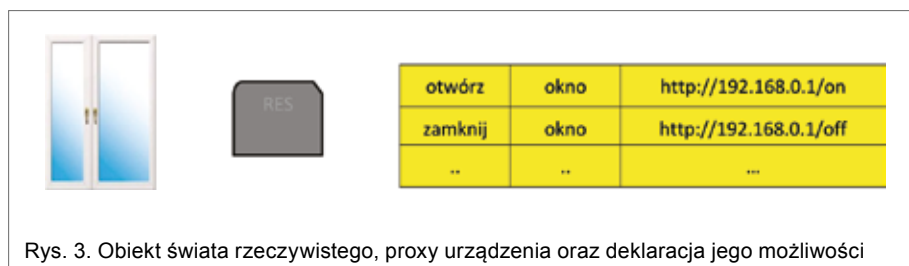
Należy tutaj wskazać na analogię obu ontologii z paradygmatem obiektowym w imperatywnym podejściu do programowania komputerów. Ontologia funkcji jest odpowiednikiem hierarchii dziedziczenia klas. Dla przykładu funkcja „Czytaj PIN” zachowuje wszystkie właściwości funkcji „Czytaj”, aczkolwiek nie jest w jakikolwiek sposób związana z funkcją „Otwórz”. W ontologii funkcji nie występują instancje. Natomiast druga ontologia – obiektów – zawiera w sobie zarówno odpowiedniki klas, jak i konkretne instancje obiektów, rozróżnialne za pomocą unikalnych identyfikatorów. Zatem w przypadku obiektów jest możliwe wskazanie zarówno wszystkich obiektów danej klasy (na przykład wszystkich drzwi), jak i konkretnych wystąpień danej klasy (zawsze liście grafu – w przykładzie z rysunku drzwi sypialni i kuchni). Wybór klasy lub instancji zależy od użytkownika, który adresuje obiekty, zgłaszając żądania dostępu.

Założono, że wymagania i żądania są analizowane pod kątem ich zgodności. Analiza zgodności wymagań i żądań polega na przetwarzaniu obu opisanych wyżej hierarchii w kierunku od liści grafu do jego korzenia (ang. *root*), przy czym za zgodne uważa się te wymagania/żądania, dla których istnieją wspólne, niepuście fragmenty (podgrafy) obu hierarchii. Należy zaznaczyć, że (1) takie podejście eliminuje potrzebę wcześniej wspomnianego statycznego adresowania urządzeń, gdyż polecenie jest zawsze dynamicznie kojarzone z właściwym dla danego kontekstu urządzeniem w dowolnej lokalizacji, w której taka czynność jest możliwa do wykonania, oraz (2) polecenie jest zawsze takie samo bez względu na miejsce, w jakim zostało użyte, co umożliwia dynamiczną zmianę tego miejsca bez konieczności wcześniejszego zaprogramowania tego polecenia i jego obsługi we wszystkich możliwych miejscach, w których może być użyte. W ten sposób zapewniono całościową realizację dwóch głównych celów niniejszej propozycji.

Do systemu można kierować żądania adresowania (wywołania) urządzeń. Jak wspomniano, żądanie może zgłosić człowiek, najczęściej za pośrednictwem swojego urządzenia personalnego, lub inne urządzenie (np. czujka ruchu, która wykryła wejście osoby do pomieszczenia). Po nadejściu żądania (czyli określenia „co chcę zrobić?”) następuje jego interpretacja (rys. 2). W pierwszej fazie są przeszukiwane hierarchie ontologii, w sposób opisany powyżej. Rezultatem tego przeszukiwania jest potencjalna lista urządzeń, które mogą spełniać wymagania żądania. Wcześniej urządzenia te zadeklarowały, co potrafią zrobić („czym jestem i co umiem”). W drugiej fazie zostaje uwzględniony kontekst. Na przykład jeśli kontekst zawiera informację o lokalizacji, z listy urządzeń są usuwane te urządzenia, które nie znajdują się w tej lokalizacji. Okrojona w ten sposób lista jest wynikiem wyszukiwania. Urządzenia, które znalazły się na tej liście, są uaktywniane albo kolejno, albo jest wybierane jedno z nich (w zależności od sytuacji lub preferencji użytkownika określonych w ramach kontekstu). Uaktywnienie urządzenia skutkuje realizacją funkcji związanych z obsługą żądania (odpowiadając na żądanie, czyli pytanie „co zrobić?”). Należy zaznaczyć, że uaktywnione w ramach powyższego procesu urządzenia mogą w czasie tego uaktywnienia zgłaszać własne żądania uaktyw-



Rys. 2. Idea kojarzenia urządzeń i żądań na podstawie wspólnego opisu ontologicznego



Rys. 3. Obiekt świata rzeczywistego, proxy urządzenia oraz deklaracja jego możliwości

niania innych urządzeń (co jest niejako odpowiednikiem wywołania procedur w klasycznych językach programowania).

4.1. Deklarowanie i wykorzystywanie możliwości urządzeń

Poniższy, dość rozbudowany przykład obrazuje wszystkie etapy przygotowania urządzenia do pracy – od rejestracji jego możliwości w katalogu, przez udostępnianie (adresowanie) za pomocą innych urządzeń, po uaktywnienie.

Z punktu widzenia implementacji możliwości urządzenia można zdefiniować jako połączenie trzech elementów: ontologii funkcji skojarzonej z urządzeniem, ontologii obiektów świata rzeczywistego, z którymi urządzenie jest fizycznie związane, oraz adresu URL serwera proxy, który jest odpowiedzialny za uaktywnienie urządzenia (dokładniej – definiowanej możliwości). Elementy te przedstawiono na rys. 3, deklarując dwie możliwości – zamknięcia i otwarcia okna. Należy zwrócić uwagę na semantykę adresów URL uaktywnienia urządzenia – jest ona w pełni zrozumiała wyłącznie dla samego urządzenia, natomiast nie jest znana ani dla osoby wywołującej urządzenie, ani dla katalogu, w którym możliwości są przechowywane. Rejestracja adresu wywołania przez urządzenie uwalnia katalog od konieczności pamiętania i zrozumienia semantyki wywoła-

nia urządzenia – to urządzenie decyduje, jak jego funkcje mają być uaktywnione, rejestrując dla definiowanej dla niego możliwości specyficzny adres URL. Katalog nie rozumie znaczenia elementów tego adresu, ale potrafi uaktywnić serwer dostępowy (proxy) dla tego adresu, tym samym wywołując urządzenie.

Ze względu na znaczne ułatwienie implementacji i fakt, że urządzenia inteligentnego budynku są zwykle niewielkie i mają niewielką moc obliczeniową, jako podstawową metodologię realizacji serwerów proxy urządzeń wykorzystano środowisko REST (ang. *Representational State Transfer*) [12].

Deklarowanie możliwości urządzenia odbywa się za pośrednictwem katalogu (rys. 4). Serwer REST (proxy) urządzenia wysyła odpowiednio sformatowany plik XML z żądaniem rejestracji możliwości skojarzonego z sobą urządzenia, dla semantyki „otwórz” i „zamknij okno” i deklarowanego adresu wywołania (w pierwszym przypadku <http://192.168.0.1/on>). Samo urządzenie nie jest uaktywniane, co jest symbolizowane przerywaną linią strzałek łączących proxy z urządzeniem.

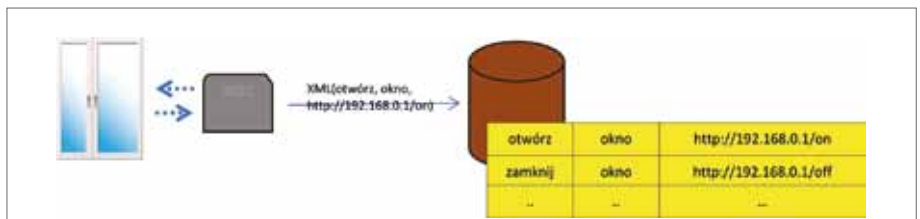
Uaktywnienie urządzenia następuje po przesłaniu do katalogu żądania o semantyce „otwórz okno” (rys. 5). Następuje porównanie semantyki z semantykami wszystkich zadeklarowanych „możliwości”. Po odnalezieniu wiersza

tabeli następuje wyodrębnienie adresu zadeklarowanego dla odszukanej możliwości: – `http://192.168.0.1/on`. Następuje wywołanie serwera REST pracującego na tym adresie. Serwer ten pobiera z adresu wcześniej zadeklarowane przez siebie parametry wywołania REST (w przykładzie – „on”) i na podstawie swojego algorytmu (który jest „czarną skrzynką” dla reszty systemu) wysyła do urządzenia komunikat „1” (na przykład wysterowuje napięciowo wybraną linię sterowania), co skutkuje otwarciem okna. Serwer REST odsyła aktualny stan urządzenia (plik XML) jako odpowiedź na wywołanie, za pośrednictwem katalogu (czerwone strzałki).

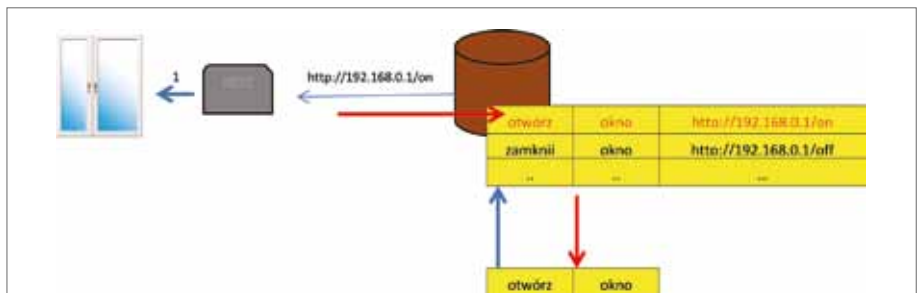
Rozważmy sytuację, gdy aktywne urządzenie ma być uaktywnione ponownie. Jak poprzednio, do katalogu przychodzi żądanie o semantyce „otwórz okno”. Następuje porównanie semantyki żądania z semantykami wszystkich zadeklarowanych możliwości. Po odszukaniu wiersza tabeli następuje wyodrębnienie adresu zadeklarowanego dla odszukanej możliwości: – `http://192.168.0.1/on`. Następuje wywołanie serwera REST pracującego na tym adresie. Serwer ten pobiera z adresu wcześniej zadeklarowane przez siebie parametry wywołania REST (w przykładzie – „on”) i stwierdza, że urządzenie jest już w wybranym stanie. Serwer REST zatem nie uaktywnia ponownie urządzenia, uznając to za niepotrzebne, tylko odsyła aktualny stan urządzenia (plik XML) jako odpowiedź na wywołanie, za pośrednictwem katalogu (strzałki zwrotne na rysunku).

Rozważmy też sytuację, gdy do katalogu przychodzi żądanie o semantyce „otwórz drzwi”. Następuje przeszukanie zadeklarowanych wcześniej możliwości. Ponieważ żadna z nich nie odpowiada semantyce żądania, nie następuje żadne uaktywnienie serwera REST (rys. 7).

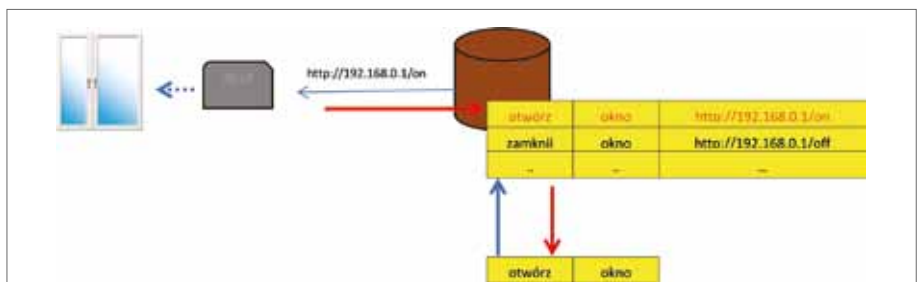
Załóżmy teraz, że do katalogu przychodzi żądanie o semantyce „otwórz budynek”. Zakładamy, że w semantyce „budynek” jest oznaczeniem korzenia drzewa, z którego wywodzą się wierzchołki „okno” i „drzwi” (rys. 1). W tym przypadku także następuje przeszukanie zadeklarowanych możliwości, ale zostają odszukane i uaktywnione dwa urządzenia o możliwościach zgodnych z semantyką żądania: REST 1 (okno) oraz REST 2 (drzwi). Każde z tych urządzeń jest uaktywniane innym adresem URL (wywołaniem REST), także każdy serwer proxy inaczej uaktywnia konkretne



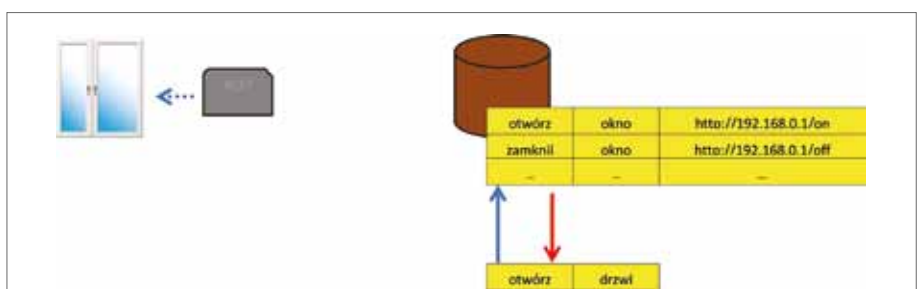
Rys. 4. Rejestracja możliwości przez urządzenie w katalogu SMS



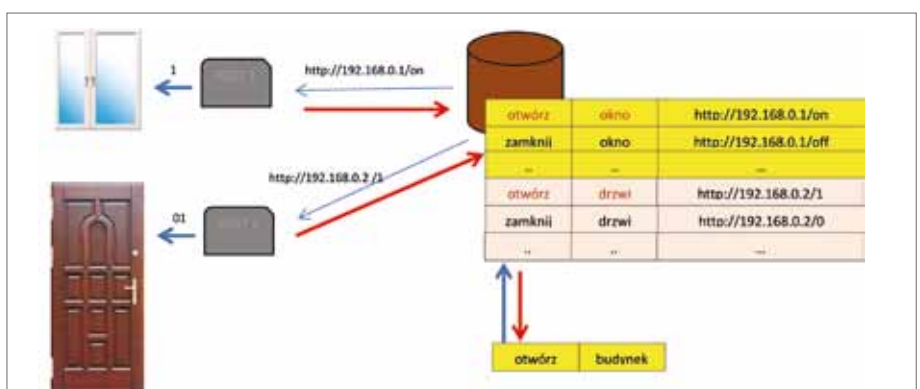
Rys. 5. Uaktywnienie możliwości urządzenia za pośrednictwem przesłania żądania do katalogu



Rys. 6. Uaktywnienie możliwości urządzenia, gdy urządzenie jest już aktywne



Rys. 7. Uaktywnienie możliwości urządzenia, które jest niedostępne



Rys. 8. Grupowe uaktywnienie możliwości wielu urządzeń jednocześnie

urządzenie fizyczne (przesłanie „1” w przypadku okna i „01” w przypadku drzwi). Zbiorcza odpowiedź jest sumą odpowiedzi obu urządzeń i jest przesyłana jako jeden plik XML.

4.2. Zalety proponowanego rozwiązania

Ontologiczne, spójne modelowanie możliwości urządzeń, kontekstu oraz semantyki napływających żądań uaktywnienia urządzeń (czyli skorzystania z oferowanych przez nie funkcji) umożliwia, z punktu widzenia użytkownika, rezygnację ze statycznego adresowania urządzeń i konieczności rozpropagowania adresów (identyfikatorów) urządzeń przed rozpoczęciem przesyłania do nich danych. Przedstawione podejście umożliwia także generyczne zaprogramowanie współpracy z urządzeniami o specyficznych (wymaganych) cechach bez względu na aktualne wartości parametrów tych urządzeń (np. nie licząc samego adresu będą to stan urządzenia, lokalizacja itp.). Raz zaprogramowany algorytm współpracy będzie poprawnie działał w dowolnym kontekście bez potrzeby dokonywania w nim jakichkolwiek zmian. Takie podejście jest szczególnie istotne w przypadku interakcji *ad-hoc* z nieznanymi z góry urządzeniami Internetu Rzeczy – wystarczy umieć wyrazić, jaki cel chce się uzyskać, a metoda osiągnięcia tego celu zostanie wypracowana przez same urządzenia, w sposób najbardziej efektywny z punktu widzenia całego systemu.

Należy zauważyć, że w odróżnieniu od wszystkich innych sposobów adresowania (1) urządzenie/osoba zgłaszająca żądanie nie musi znać fizycznego (statycznego) adresu urządzenia (zatem w skrajnym przypadku wybrane urządzenie może pozostać całkowicie anonimowe) oraz (2) zgłoszenie tego samego żądania w innym kontekście (np. w innym miejscu) skutkuje potencjalnie uaktywnieniem innych urządzeń (przenośność). Zatem za pomocą jednej technologii są osiągnięte dwa cele, które są niedostępne dla aktualnie wykorzystywanych sieci: w pełni dynamiczne adresowanie urządzeń Internetu Rzeczy o identyfikatorach nieznanymi z góry (a nawet całkowicie anonimowymi – przed i po transmisji danych) oraz brak konieczności przeprogramowywania danych urządzeń (programów sterujących, adresów itp.) w przypadku zmiany kontekstu ich działania, w szczególności migracji w inne miejsce sieci.

5. Wnioski końcowe

W artykule opisano nowy sposób wyszukiwania, adresowania i uaktywniania urządzeń inteligentnego budynku. W odróżnieniu od klasycznych metod adresowania, zakładających sztywne powiązanie urządzeń sterujących i sterowanych za pomocą jednoznacznych identyfikatorów (tzw. „parowanie” urządzeń), w przedstawianym podejściu jest wykorzystywane dynamiczne łączenie kontekstowe w połączeniu z modelowaniem ontologicznym. Model obejmuje możliwości urządzeń, które są modelowane nie na poziomie charakterystyki (danych technicznych, sposobu podłączenia, adresu itp.) urządzenia, a na podstawie funkcji, jakie dane urządzenie pełni. Model obejmuje też kontekst, na który składają się te parametry, które są niezależne od wywołania, np. aktualna lokalizacja urządzenia lub osoby, temperatura w pomieszczeniu itp. Ostatnim modelowanym elementem są żądania wywołania odpowiedniej akcji, czyli uaktywnienia urządzeń. Zastosowane modele ontologiczne pozwalają na sformułowanie wymagań i zgłoszenie żądania, następnie dołączenie ograniczeń wynikających z kontekstu i ostatecznie dynamiczny wybór tych (grup) urządzeń, które, na podstawie

wcześniej zadeklarowanych przez siebie możliwości, są w stanie te wymagania spełnić. Urządzenia te są następnie aktywowane, co skutkuje wykonaniem żądanej akcji.

Opisywane podejście zostało zaimplementowane w postaci środowiska SITE (ang. *Semantic Internet-of-Things Environment*) [13]. Środowisko to okazało się użyteczne w modelowaniu wzajemnych interakcji między użytkownikami i urządzeniami inteligentnego budynku, szczególnie w kontekście miejsc publicznych, takich jak muzea, szpitale, urzędy itp., gdzie klasyczne sterowanie wyposażeniem budynku przez incydentalnych użytkowników, z wykorzystaniem ich urządzeń personalnych, nie może być stosowane.

Literatura

- [1] ARTHUR C. CLARKE: *Hazards of Prophecy: The Failure of Imagination*, Profiles of the Future (1962).
- [2] MARC WEISER: *The Computer for the 21st Century*, Scientific American Special Issue on Communications, Computers, and Networks, 1991, <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>
- [3] *What is a smarter planet?* IBM white paper, from <http://www.ibm.com/smarterplanet/us/en/overview/ideas/>
- [4] *Things That Think Consortium*, Massachusetts Institute of Technology, <http://ttt.media.mit.edu/>
- [5] KAY, ALAN, *Powerful Ideas: Useful Tools to Understand the World*, https://itp.nyu.edu/registration/CourseInfo.php?course_id=489
- [6] Open Systems Interconnection (OSI) model, ISO/IEC 7498-1 standard, http://en.wikipedia.org/wiki/OSI_model
- [7] RFC1166 „Internet Numbers”, 1990: zasady adresowania IP (klasy i rezerwacje adresów), <http://tools.ietf.org/html/rfc1166>
- [8] Controller Area Network, http://pl.wikipedia.org/wiki/Controller_Area_Network
- [9] Semantic Web, http://en.wikipedia.org/wiki/Semantic_Web
- [10] Ontologia – definicja encyklopedyczna, na podstawie Wikipedii, <http://pl.wikipedia.org/wiki/Ontologia>
- [11] GADOMSKI A.M.: *Meta-Ontological Assumptions: Information, Preferences and Knowledge (IPK): universal cognitive architecture*, <http://erg4146.casaccia.enea.it/wwwerg26701/gad-dict.htm>
- [12] Representational State Transfer REST, http://pl.wikipedia.org/wiki/Representational_State_Transfer
- [13] RYKOWSKI J., HANICKI P., STAWNIAK M.: *Ontology Scripting Language to Represent and Interpret Conglomerates of IoT Devices Accessed by SOA Services*, in: *SOA Infrastructure Tools – Concepts and Methods*, ed. AMBROSZKIEWICZ, S., J. BRZEZIŃSKI, W. CELLARY, A. GRZECH, AND K. ZIELIŃSKI, Wydawnictwa Uniwersytetu Ekonomicznego w Poznaniu, Poznań, 2010, pp. 235–262.

dr hab. inż. Jarogniew Rykowski, prof. nadzw. UEP
Katedra Technologii Informatycznych, Uniwersytet Ekonomiczny
w Poznaniu, e-mail: rykowski@kti.ue.poznan.pl