

Streszczenie

Celem tej pracy jest zaprojektowanie i udostępnienie prototypu rozwiązania problemu zdalnej akwizycji informacji o wzorcach, występujących w danych prawnie klasyfikowanych jako wrażliwe.

Cel ten motywowany jest minimalizacją powierzchni ataku na dane natury osobistej, a także ograniczeniem kosztów dodatkowej infrastruktury potrzebnej dostawcy do wdrożenia usługi. Proponowany model osiąga to poprzez zaangażowanie odbiorcy, który zyskuje w ten sposób dodatkowe możliwości. W pracy przedstawiono architekturę opracowaną na potrzeby celu szczegółowego w postaci automatycznej certyfikacji elastycznych odbiorców energii elektrycznej.

Dla osiągnięcia celu użyto autorskich algorytmów zaprojektowanych z użyciem wzorców programowania asynchronicznego i współbieżnego, oraz ich prototypowych implementacji.

Słowa kluczowe: DSR, pomiar elastyczności, architektura, programowanie asynchroniczne.

An automated system of crime evaluation

The aim of this thesis is design and publication of a prototype of a solution to the problem of remote acquisition of information about patterns emerging in data classified as sensitive by law.

The mentioned aim is motivated by a vision of lessening the attack surface where data of personal nature is concerned; also - of limiting the cost of additional infrastructure needed by the provider in order to implement the service. The proposed model achieves that through engagement of the user, who in turn gains additional possibilities. In this document an architecture is presented, designed to fulfill a specific purpose of automatizing certification of flexible users of electric energy.

Used for the purpose of achieving the stated goal were original, customly designed algorithms based on patterns of asynchronous and parallel programming, together with their prototypical implementations.

Keywords: DSR, flexibility estimation, architecture, asynchronous programming.

Spis treści

Definicje	5
1 Wstęp	7
2 Cele i założenia pracy	9
Kryteria oceny rozwiązania	10
Zakres pracy	10
Założenia funkcjonalne	10
Ograniczenia realizacyjne	11
Założenia implementacyjne	11
Plan pracy	12
3 Problematyka elastycznego odbioru	13
Typy elastyczności	13
Elastyczny odbiorca	14
Kwestie kontrolowania poboru energii	15
Rola agregatora	17
Program DSR w Polsce	18
4 Projektowanie architektury systemu	21
Podstawowy algorytm badawczo-rozwojowy	21
ArchiMate w analizie architektur	21
Modularyzacja architektury	23
Architektura mgłowa	24
Przetwarzanie współbieżne	25
5 Koncepcja rozwiązania	29
Indeks elastyczności	30
Interesariusze	31
Składowe systemu	31
Urządzenie FlexMeter	32
Moduł scenariuszy	33
Moduł FlexCheck	34
Struktury danych	35
Wzór implementacji	35
Komunikacja zewnętrzna	36
Interfejs odbiorcy	37
Poziomy prywatności	38
Ograniczenia binarnej arytmetyki zmiennoprzecinkowej	38
Rekomendacje dotyczące typu danych Decimal	40

6	Wyniki testów	41
	Funkcjonalność prototypu	41
	Podsumowanie testów funkcjonalności	43
	Wydajność prototypu	43
	Podsumowanie	44
7	Podsumowanie	45
	Proponowane kierunki dalszego rozwoju	46
	Wnioski	46
	Literatura	47

Definicje

- Architektura korporacyjna - architektura opisująca przedsiębiorstwo jako system dla celów analizy lub wprowadzenia zmian; wbrew brzmieniu terminu nie rozumiemy pod nim jedynie “korporacji” w polskim znaczeniu tego słowa [1],
- Agregator - podmiot gromadzący odbiorców; jego celem jest zaoferowanie operatorowi sieci przesyłowej zwiększonej stabilności; w tej pracy również nazywany zamawiającym jako główny zainteresowany korzyściami płynącymi z implementacji opisanego systemu;
- Certyfikacja - proces prowadzony przez agregatora dla celów uszeregowania odbiorców pod względem niezawodności rozumianej jako prawdopodobieństwo skutecznego dostarczenia agregatorowi przez danego odbiorcę produktu o określonych parametrach i w odpowiednim czasie.
- DSR (*demand-side response*) - program angażowania strony popytowej w zapewnienie stabilności sieci elektroenergetycznej,
- Elastyczność - zdolność odbiorcy do modyfikowania swego profilu zapotrzebowania na żądanie [2],
- FMA (*Flexibility Measurement Architecture*) - skrót oznaczający architekturę pomiaru elastyczności opisywaną w niniejszej pracy,
- Indeks elastyczności - miara elastyczności oznaczana symbolem FI , spełniająca warunki: $FI = x | x \in \mathbb{R}^+ \wedge x \leq 1$ [3],
- Metoda szablonowa - jeden ze wzorców projektowych w oprogramowaniu, polegający na udostępnieniu szablonu metody złożonej z elementów, w którym następnie użytkownik może dostosować poszczególne elementy do własnych potrzeb [4];
- Monada - podstawowa jednostka kompozycji obliczeń; monada zwraca wartość typu a , albo *Nothing*,
- Odbiorca - gospodarstwo domowe lub przedsiębiorstwo pobierające energię elektryczną; w kontekście tej pracy również nazywany użytkownikiem w kontekście osoby korzystającej z implementacji przedstawionej w tej pracy architektury dla celu pomiaru własnej elastyczności,
- Pomiar elastyczności - proces wyznaczania indeksu elastyczności,
- Produkt DSR - reakcja odbiorcy o parametrach zgodnych z ustaleniami zawartymi w ramach umowy o świadczenie usługi DSR.
- Próba elastyczności - składowa procesu pomiaru elastyczności odbiorcy,
- Scenariusz - określany przez zamawiającego zbiór warunków, jakie muszą zostać spełnione celem zaliczenia użytkownikowi próby elastyczności,
- Zamawiający - agregator użytkowników,

Rozdział 1

Wstęp

W sierpniu 2015 roku polski rynek energii elektrycznej po raz pierwszy od 35 lat znalazł się w sytuacji wprowadzenia 20. stopnia zasilania [5]. Powodem były przedłużające się susze i związany z nimi niedobór wody niezbędnej do chłodzenia generatorów w elektrowniach.

Na takie sytuacje przewidziany jest mechanizm tzw. stopni zasilania, który określa czasowe limity poboru energii dla podmiotów o zapotrzebowaniu powyżej 300 KW. Limity te są drastyczne i równe dla wszystkich największych odbiorców. 10 sierpnia 2015 roku komunikat o ogłoszeniu stopni zasilania na czas między godziną 10:00 a 22:00 [6] został nadany o 7:55, czyli z wyprzedzeniem wynoszącym 2 godziny i 5 minut [7].

Do czerwca 2017 roku Urząd Regulacji Energetyki wszczął 1,9 tysiąca postępowań wobec firm, które w sierpniu 2015 roku nie dostosowały się do ograniczeń poboru energii elektrycznej [8]. W przypadku 600 przedsiębiorstw zdecydowano o nałożeniu kary, która wyniosła od tysiąca do kilkudziesięciu tysięcy złotych [9]. Ponadto, rozważano przeniesienie odpowiedzialności z podmiotów gospodarczych na osoby bezpośrednio odpowiedzialne za zaniedbania [9].

W lipcu 2018 warunki w środowisku naturalnym i związana z nimi wysokość zapotrzebowania na energię przypomniały specjalistom sytuację z roku 2015 [10]. Przewidywania okazały się słuszne, ponieważ w kilka dni później spółka PSE ogłosiła lokalny stan zagrożenia na terenie dwóch województw w związku z trzecim rekordem popytu na energię w szczycie wiosenno-letnim [11].

W sytuacji gdy gwałtowny wzrost zapotrzebowania na energię w skali zagrażającej ciągłości dostaw może wystąpić z dnia na dzień [12], szans na racjonalniejszy podział zasobów upatruje się w zaangażowaniu odbiorców w program aktywizacji popytu dla utrzymania zagrożonych dostaw energii [13]. Szczególnie interesująca jest możliwość projektowania nowych narzędzi wspierających aktywizację elastycznego odbioru.

Wzrastająca wydajność, energooszczędność i idące za nimi możliwości urządzeń wbudowanych osiągnęły poziom, w którym zasadne jest rozpatrywanie tworzonych rozwiązań z perspektywy późniejszej implementacji nie tylko na wydajnych scentralizowanych serwerach albo w chmurze, lecz także szukanie rozwiązań pośrednich łączących zalety tych dwóch podejść. Wobec rosnącej popularności architektury mgłowej nie sposób zaprzeczyć, że pozwala ona na osiągnięcie niemożliwego dotychczas poziomu zrównoważenia użyteczności, wygody oraz prywatności danych.

System przedstawiony w niniejszej pracy został zaprojektowany dla dostarczenia zainteresowanemu świadczeniem usług DSR odbiorcy środków do oszacowania potencjału DSR własnego gospodarstwa domowego, zakładu produkcyjnego albo kompleksu biurowego. Wobec dostępności narzędzi do szaco-

wania własnej zdolności reagowania na bieżące potrzeby sieci elektroenergetycznej łatwiejsze staje się nie tylko włączenie się w zapewnienie bezpieczeństwa tej sieci, lecz także finansowe zabezpieczenie takiej działalności.

Rozdział 2

Cele i założenia pracy

Celem niniejszej pracy jest opracowanie systemu wspierającego indywidualnego odbiorcę energii elektrycznej w rozpoczęciu świadczenia usług określanych mianem Demand Side Response, czyli reakcji strony popytowej na sygnał operatora systemu, w oparciu o własną elastyczność.

Cel pracy rozpatrywany jest zatem w dwóch aspektach. Pierwszym aspektem jest identyfikacja potrzeb związanych z szacowaniem elastyczności odbiorców energii elektrycznej. Drugim z aspektów jest zaprojektowanie architektury systemu dla wsparcia odbiorcy w podejmowaniu decyzji związanych z przystępowaniem do programu DSR umożliwionym odbiorcy przez agregatora.

Opracowany system ma pozwalać odbiorcy na zautomatyzowane oszacowanie możliwości dostosowania własnego zapotrzebowania na energię do potrzeb sygnalizowanych przez operatora tego systemu. Usługi takie mogą być świadczone w ramach współpracy odbiorców z ich przedstawicielem, tzw. agregatorem. Zatem opracowywany system musi się też wpisywać w część infrastruktury informatycznej agregatora.

Podstawowym założeniem jest dostępność odpowiedniej infrastruktury po stronie odbioru, czyli tak zwanego inteligentnego licznika, który dostarczał będzie do systemu dane o bieżącym zużyciu energii.

Oszacowanie elastyczności odbiorcy z użyciem systemu w zaprojektowanej architekturze może się odbywać jedynie w ramach grupy świadczącej usługi DSR pod nadzorem Agregatora.

Dla osiągnięcia głównego celu pracy zrealizowano następujące cele szczegółowe:

- Identyfikacja i wybór parametrów pozwalających jak najlepiej określić i mierzyć elastyczność odbiorcy energii,
- Określenie charakterystycznych wzorcowych profili zmian zużycia energii i opisanie ich w odniesieniu do zdefiniowanej miary elastyczności,
- Opracowanie koncepcji szacowania elastyczności poprzez odniesienie do wzorcowych profili elastyczności - szacowanie to może być realizowane zarówno indywidualnie, jak też w sposób automatyczny może być realizowane przez agregatora w stosunku do wszystkich podmiotów, które reprezentuje,
- Zaprojektowanie architektury systemu umożliwiającej realizację opracowanej koncepcji,
- Dostarczenie odbiorcy przy pomocy systemu informacji istotnych dla zawarcia obopólnie korzystnej umowy współpracy z agregatorem.
- Wykonanie i przedstawienie wzorcowej, prototypowej implementacji kluczowego modułu oprogramowania systemu oraz opisanie jego funkcjonowania.

Przydatność opracowanego rozwiązania oceniono przeprowadzając proste symulacje pracy z systemem. Przyjmując pewne założenia o możliwościach modyfikacji własnego zużycia pokazano, że opracowane koncepcje w sposób prawidłowy klasyfikują i wspomagają ocenę elastyczności.

Kryteria oceny rozwiązania

Za wypełniające założenia pracy rozwiązanie problemu uznaje się przedstawienie projektu architektury systemu oferującego opisaną funkcjonalność oraz demonstrację zastosowania takiego systemu w roli narzędzia wspierającego elastycznego odbiorcę w określeniu własnej elastyczności.

Oszacowanie elastyczności odbiorcy z użyciem systemu w zaprojektowanej architekturze należy przeprowadzić w sposób użyteczny dla odbiorcy w procesie dołączania do grupy świadczącej usługi DSR pod nadzorem Agregatora.

Dane pozyskane przy pomocy systemu w zaproponowanej architekturze powinny dostarczać odbiorcy informacji istotnych dla zawarcia obopólnie korzystnej umowy współpracy z Agregatorem.

Zakres pracy

Tematyka podjęta w pracy dotyczy innowacyjnej koncepcji aktywizacji odbiorców energii - koncepcji, która nie była jeszcze wdrażana na szerszą skalę w polskich warunkach.

W związku z tym, pominięto w pracy szereg innych typowych problemów rozwiązanych dotychczas w sposób satysfakcjonujący, jak na przykład rozpatrywanie niektórych zagadnień w dziedzinie przetwarzania danych. W części implementacyjnej skupiono się na opracowaniu fragmentu systemu, który zostanie zainstalowany w bezpośredniej styczności z urządzeniem pomiaru zużycia energii odbiorcy.

Przyjmuje się natomiast spełnienie następujących założeń istotnych zarówno ze względu na aspekty funkcjonalne jak i nefunkcjonalne.

Założenia funkcjonalne

Przeznaczeniem systemu jest umożliwienie odbiorcy oszacowania własnej elastyczności w warunkach określonych przez Agregatora.

Możliwa ma być instalacja głównego modułu sprzętowego zaprojektowanego systemu w obudowie licznika energii elektrycznej, lub ewentualnie na tej obudowie. Założenie to wnosi wspomniane dalej ograniczenia dotyczące przestrzeni dostępnej do celów montażu urządzenia.

Główną koncepcją, na której oparto się w pracy, jest odniesienie się do wzorcowych profili modyfikacji zużycia, dla możliwie zróżnicowanych warunków funkcjonowania odbiorcy, określanych dalej scenariuszami. Bardziej szczegółowo koncepcja ta została wyjaśniona w rozdziale poświęconym koncepcji rozwiązania.

Funkcje te umożliwiać mają szczególnie:

- Możliwość bieżącego monitorowania danych o zużyciu przez agregatora oraz przez odbiorcę,
- Wspieranie odbiorcy w definiowaniu reprezentatywnych scenariuszy opisujących środowisko funkcjonowania odbiorcy,
- Wspieranie agregatora w definiowaniu reprezentatywnych profili reakcji odbiorcy na zgłoszenie potrzeby realizacji usługi,
- Wspieranie dopasowania reakcji odbiorcy do reprezentatywnego profilu elastycznego zużycia,

- Umożliwienie komunikacji od agregatora do odbiorcy celem przesłania informacji o potrzebie modyfikacji zużycia,
- Umożliwienie własnego odczytu oceny elastyczności przez odbiorcę, jak również odczytu tej oceny przez agregatora,

Opracowany system ma charakter modułowy, składa się z kilku elementów realizujących odpowiednie funkcjonalności zarówno po stronie odbiorcy, jak i agregatora.

Główną częścią opracowywanej architektury jest moduł przeznaczony do szacowania elastyczności odbiorcy przez połączenie danych o jego zużyciu energii z innymi wymaganymi informacjami. Praca jest poświęcona opisowi oraz ograniczonej do obliczeń na profilach elastyczności oraz podstawowej komunikacji z użytkownikiem i agregatorem realizacji tej części systemu.

Ograniczenia realizacyjne

Docelowe środowisko pracy urządzenia charakteryzuje się następująco:

- Jest środowiskiem o ograniczonej przestrzeni i limitowanych możliwościach wypromieniowywania ciepła,
- Stabilne połączenie z urządzeniem nie może być zagwarantowane - kanał transmisji ma ograniczoną przepustowość, a jego niezawodność jest niska, co objawia się perspektywą zerwania połączenia w dowolnej chwili,
- Możliwa jest ingerencja osób trzecich w wymianę danych pomiędzy dostawcą a odbiorcą.

Założenia implementacyjne

Następujące ograniczenia wynikają z dostępnego środowiska narzędziowego oraz ograniczonego dostępu do informacji o komercyjnych rozwiązaniach analitycznych.

W doborze narzędzi do realizacji projektu kierowano się prostotą zastosowania i ewentualnej rozbudowy o potrzebną funkcjonalność, oraz kompletnością dostępnej dla autora dokumentacji produktu. W związku z tym założeniem w dziedzinie oprogramowania rozpatrywano przede wszystkim rozwiązania o otwartych źródłach.

Jako jednostkę czasu przyjmuje się godzinę wszędzie tam, gdzie nie zaznaczono inaczej. Oznacza to, że w pracy odnosimy się do poboru energii uśrednionego w obrębie godziny. Takie założenie wynika z zasad realizacji usługi DSR, do której odniesiono się w pracy.

W pracy nie rozważa się ekonomicznych aspektów ewentualnego wdrożenia efektów prac, ponad minimum niezbędne do oceny rozwiązania w dziedzinie technicznej. Skupiono się w niej na wspomaganiu szacowania parametrów oferty odbiorcy w oparciu o pulę możliwości zdefiniowanych przez agregatora tak, aby zapewnić zrealizowanie usługi redukcji na wezwanie OSP, natomiast pominięto rozważanie kwestii związanych z opracowywaniem tak zwanych strategii ofertowych, czyli wspomagania takiej budowy oferty, która zmaksymalizuje szanse jej przyjęcia przez OSP.

Wyniki niniejszej pracy mają charakter badawczy i prototypowy. W pracy pominięto kwestie związane z odpornością przygotowanego oprogramowania na celowe próby ingerencji w stabilność jego wykonania, a także wszelkie problemy związane z bezpieczeństwem informacji inne niż zagadnienie autentyczności i spójności danych przesyłanych pomiędzy odbiorcą a dostawcą. Założono, że fizyczną integralność systemu wdrażanego w oparciu o wyniki pracy zapewni podmiot odpowiadający za implementację.

Zakłada się ponadto zapewnienie przez podmiot implementujący dostępności przy pomocy łączności bezprzewodowej odczytów bieżącego zużycia energii w rozdzielczości godzinowej, a także danych dotyczących stanu, historii i przewidywań pogody oraz prognoz zużycia dla danego odbiorcy.

Plan pracy

Dalsza część tej pracy zorganizowana jest następująco:

W rozdziale trzecim zdefiniowano i osadzono w kontekście podstawowe pojęcia związane z problematyką elastycznego odbioru. Następnie zidentyfikowano i omówiono potrzeby występujące w poszczególnych poddziedzinach zadania. Rozdział zakończony został podsumowaniem i wnioskami dotyczącymi ostatecznego kształtu projektowanej architektury systemu.

Rozdział czwarty poświęcony został przeglądowi metod i narzędzi wybranych do realizacji postawionego problemu.

Rozdział piąty przeznaczono na opis autorskiej koncepcji realizacji obranego w pracy celu oraz przedstawienie sposobu wykorzystania zaprojektowanego systemu przez użytkownika.

W rozdziale szóstym zaprezentowano wyniki testów opracowanego rozwiązania.

W rozdziale siódmym zawarto podsumowanie przeprowadzonych prac pod kątem wypełnienia wstępnych założeń, przedstawiono końcowe wnioski oraz zaproponowano drogi przyszłego rozwoju opracowanego systemu.

Rozdział 3

Problematyka elastycznego odbioru

Program aktywizacji strony popytowej (*Demand-Side Response, DSR*) pozwala na większe, niż w przypadku stopni zasilania, wykorzystanie mechanizmów wolnego rynku.

Według założeń programu, w sytuacji nagłego niedoboru energii niektóre podmioty dobrowolnie rezygnują z zaspokojenia części swego zapotrzebowania na rzecz pozostałych podmiotów. Wstrzymujący się od części konsumpcji otrzymują odpowiednie finansowe wynagrodzenie, a jego wymiar zależy od wielkości dostawy i jej zgodności z umową.

Mechanizmy motywowania odbiorców mogą być w ogólnym przypadku rozbudowane i złożone, tematyka ta wykracza poza obszar zainteresowania tej pracy.

Przejsięcie do modelu rynkowego, w którym odbiorca współuczestniczy w zapewnianiu stabilności dostaw energii dla wszystkich uczestników rynku, trwa od lat i jest wskazywane jako jedna z najbardziej korzystnych z bieżących perspektyw w dziedzinie energetyki [14].

Typy elastyczności

W swojej pierwszej pracy na temat DSR, w roku 1985 autor koncepcji Clark W. Gellings zidentyfikował sześć głównych typów potrzeb dotyczących charakterystyki dziennego poboru energii elektrycznej ???.

Przedstawiono je na Rys. 3 na podstawie pracy ???.

Cele zarządzania stroną popytową, opracowanie własne za ???

Zilustrowane na Rys. 3 obciążenie szczytu oznacza ograniczenie poboru w okresie największego zapotrzebowania do zadanej wielkości. Celem przedstawionego następnie wypełnienia dolin jest podniesienie wykorzystania energii poza okresem szczytowego zapotrzebowania. Następne w kolejności przesunięcie obciążeń związane jest z połączeniem dwóch pierwszych zachowań przez rozdysponowanie części obciążenia w szczycie do czasu przed tym szczytem oraz bezpośrednio po nim.

Dwa dalsze rodzaje charakterystyk z Rys. 3 niosą ze sobą konieczność odpowiednio: zwiększenia oraz zmniejszenia popytu na energię w szczycie, przy dopuszczeniu idącej za tym zmiany zapotrzebowania w okolicach szczytu - stosunkowo niewielkiej i zgodnej co do typu ze zmianą w szczycie.

Ostatni z celów przedstawionych na Rys. 3 to przypadek ogólny, określany mianem kształtowania krzywej obciążenia. O kształtowaniu krzywej obciążenia mówimy gdy oczekiwany kształt krzywej

obciążenia jest bardziej złożony niż w przypadku poprzednich pięciu wariantów. W takich okolicznościach możliwe jest wpływanie na odbiorców przy pomocy zestawu sposobów dobranych tak, aby otrzymana na koniec krzywa obciążenia posiadała cechy jak najbardziej zbliżone do oczekiwanych.

Elastyczny odbiorca

Pod pojęciem odbiorcy energii rozumie się obiekt pobierający z sieci elektroenergetycznej określoną ilość energii w zadanym okresie pomiaru.

Przyjmąwszy symbol P dla oznaczenia mocy pobieranej przez określonego odbiorcę O , funkcję obciążenia sieci przez odbiorcę O przedstawiamy w ogólności w postaci:

$$P = P(X) \quad (3.1)$$

Zmienna X w równaniu oznacza zbiór wszystkich zmiennych, od których zależy pobór energii w chwili t . Zgodnie z 3.1 do zbioru zmiennych X należy zatem między innymi zmienna czasowa t . Dla oznaczenia pozostałych zmiennych, od których zależy pobór energii w chwili t , używany jest symbol $X' \equiv X \setminus \{t\}$.

Przykładowe obciążenie generowane przez odbiorcę w ciągu doby ilustruje krzywa przedstawiona na Rys. 3.

Przykładowa krzywa dziennego obciążenia.

Odbiorca nieelastyczny wykazuje charakterystykę odbioru w przybliżeniu niezależną od warunków panujących w sieci elektroenergetycznej. Upraszczając zatem równanie 3.1 dla przypadku odbiorcy nieelastycznego O_n otrzymujemy funkcję P_n następującej postaci.

$$P_n = P(t) \quad (3.2)$$

W postaci tej funkcja P zależy wyłącznie od czasu.

Odbiorca elastyczny to odbiorca, który w przeciwieństwie do poprzedniego przypadku jest zdolny do modyfikowania profilu swego zapotrzebowania w wypadku zaistnienia takiej potrzeby. Funkcja zapotrzebowania odbiorcy elastycznego P_e uwzględnia warunki panujące w sieci elektroenergetycznej, zmieniając się w zależności od nich oraz od czasu t .

$$P_e = P(X', t) \quad (3.3)$$

Reakcję elastycznego odbiorcy energii przedstawiono na Rys. 3 w opracowaniu własnym za pracę ???.

Warianty modyfikacji profilu zapotrzebowania na energię elastycznego odbiorcy, opracowanie własne za ???

Na Rys. 3 widzimy kilka możliwych typów odpowiedzi elastycznego odbiorcy na sygnał operatora systemu. Zaznaczony na rysunku tzw. efekt odbicia nawiązuje do wzorca profilu zużycia, w którym odbiorca zmniejsza swoje zapotrzebowanie w zadanym przedziale czasu aby następnie zwiększyć je tuż po nim.

Aktywność odbiorców elastycznych sprawia, że mówić możemy o możliwości wygenerowania spadku zapotrzebowania na energię w określonym wymiarze i na określony czas. Mówimy wtedy o dostarczeniu do sieci “produktu” w sensie zwolnienia π kW mocy na czas δ minut z bezwładnością startu ρ kW/min, przy dostarczeniu do systemu energii w całkowitej ilości ϵ kWh ???.

Ilustrację koncepcji produktu DSR przedstawiono na Rys. 3 w oparciu o pracę ???.

Produkt DSR, opracowanie własne w oparciu o ???

Problemem związanym z prowadzeniem programu aktywnego popytu jest oszacowanie wielkości spodziewanej rzeczywistej redukcji ???. Trudno jest natomiast wskazać jednoznaczną miarę elastyczności, jak ma to miejsce w tradycyjnej definicji ekonomicznej ze względu na dynamiczny charakter tego procesu ???.

W źródłach dotyczących grupowania elastycznych odbiorców proponowane są podejścia, których zestawienie zawarto w tabeli 3.1. Szerzej temat ten omawia praca [15].

Tabela 3.1: Porównanie opisanych w literaturze przykładów metod szacowania elastyczności.

Rok	Autorzy pracy	Miara elastyczności
2013	Mangesius i inni [16]	Prawdopodobieństwo wytrzymania określonego stałego obciążenia
2013	Ulbjg i Andersson ???	Wielkość i szybkość zmiany konsumpcji energii
2018	Junker i inni [3]	Koszt ignorowania zmian cen energii
2018	Wang i inni [17]	Zmiana zapotrzebowania na żądanie w danej porze dnia

Podejść do szacowania elastyczności jest wiele, co widzimy w tabeli 3.1. Główną trudnością zaangażowania odbiorcy w te metody jest jednak skomplikowanie modeli, na których się opierają. Szansą jest tu bezpośrednie szacowanie elastyczności na podstawie rzeczywistego zużycia przez samego odbiorcę.

Bazując na wnioskach z literatury możemy jednak zdefiniować indeks oraz funkcję, których będziemy używać do zdefiniowania miary elastyczności.

Kwestie kontrolowania poboru energii

Zauważono ???, że osoby donoszące o trudnościach finansowych związanych z konsumpcją energii elektrycznej, były przygotowane do poświęcenia większej uwagi kwestiom zużycia energii, a także do redukcji swego zapotrzebowania. Spostrzeżenia tego dokonano podczas analizy wyników sondażu dotyczącego systemu DSR, przeprowadzonego pośród 2441 reprezentantów populacji Wielkiej Brytanii.

Autor analizy dodaje, że wspomniane osoby były równocześnie mniej skłonne do udostępniania danych o swoim zużyciu energii, przywołując w tym miejscu hipotezę, że ci posiadający mniejszy wpływ na społeczeństwo mogą postrzegać siebie jako bardziej wrażliwych na wykorzystanie.

Pośród osób wyrażających obawy finansowe spodziewane jest również, że dla części z nich finansowe korzyści z technologii DSM (Demand-Side Management, nadzbiór DSR), nie będą od razu dostrzeżone - albo też nie będą wydawały się dostatecznie wiarygodne.

Obserwacje na rynku w Stanach Zjednoczonych wskazują na sceptycyzm dotyczący zagadnienia, czy energetyczne technologie przyszłości przyniosą ogólną redukcję kosztów, ograniczenie uwagi użytkownika wymaganej przez system oraz zniwelowanie kosztów ukrytych inwestycji w technologie energetyczne ????. Pośród ukrytych kosztów inwestycji znajdują się na przykład te wiążące się z potrzebą okresowej wymiany i napraw urządzeń wspomagających zarządzanie energią po stronie odbiorcy.

Jedną z najbardziej znaczących dla konsumentów stron aktywności DSR wydaje się być nie strona finansowa - w ujęciu własnej finansowej korzyści - a raczej współdziałanie dla wspólnego dobra. Jedno z badań wskazało, że niektórzy konsumenci byliby chętni przyjąć stały monitoring swego zużycia energii, nie oczekując za to wynagrodzenia w jakiegokolwiek formie jeżeli mogłoby to pomóc w znanym czy nawet nie znanym im szczegółowo celu. Jako wystarczającą formę zapewnienia prywatności odczuwają oni bycie "jedną małą cyfrą pośród 10 milionów ludzi" ???.

Nie brakowało jednak respondentów, którym bliska była troska o zapewnienie należytej prywatności. Oczekiwali oni adekwatnej ochrony swoich danych, usuwania ich po określonym czasie, a także szansy na własny wgląd do nich ????. Zatem oczekiwania odbiorców dotyczące ochrony danych o własnym zużyciu energii są zróżnicowane i mogą mieć realny wpływ na decyzje powiązane z potrzebą udostępnienia takich danych.

Jedną ze znaczących przyczyn zainteresowania udziałem w DSM związana jest z możliwością precyzyjniejszego samodzielnego kontrolowania własnego zużycia - w opozycji do bycia automatycznie kontrolowanym. W badaniu przeprowadzonym na terenie kilku krajów europejskich jedna z respondentek wskazała, że "wspaniale byłoby mieć jakiś wgląd w to, jak wiele pieniędzy jest oszczędzane, oraz jak wiele energii jest oszczędzane. Miło jest wiedzieć, jaki jest mój wkład w czystsze środowisko." ???.

Z badań społecznych przeprowadzonych na zlecenie PTPiREE w ramach projektu „Inteligentne sieci – dla domu środowiska i gospodarki” wynika, że dla Polaków w relacjach z dostawcami energii elektrycznej najistotniejszym czynnikiem są aspekty ekonomiczne – cena i możliwość oszczędzania. Na dalszy plan schodzą komfort, bezpieczeństwo i ekologia ???.

Jak zaznaczono w pracy [18], poprzez uzyskanie dostępu do danych z inteligentnego licznika można wywnioskować odpowiedzi na wiele pytań dotyczących osobistych, potencjalnie głęboko prywatnych, zachowań użytkowników. Podczas gdy niektóre z tych odpowiedzi mogą się wydawać nieszkodliwe, jak np. pora oglądania telewizji, inne mogą być dość dotkliwe, jak np. obserwacja, że w domu obecny jest noworodek.

Pytania o prywatność w istotny sposób wpłynęły na ostateczny kształt rozwiązania tej pracy. Przykładowe z nich przytoczono w tabeli 3.2 za [18].

Rozwiązania inteligentnego pomiaru zużycia energii przechowujące dokładne dane o zużyciu u dostawcy energii wciąż należą do najpopularniejszych [19]. Także w roku 2018 podstawową architekturą rozwiązań monitorowania zachowań konsumentów było wysyłanie ich danych dotyczących zużycia energii do zewnętrznego centrum przetwarzania, gdzie następnie dane były analizowane [20].

Tabela 3.2: Prywatne pytania oraz odpowiedzi odsłaniające przez szczegółowe dane o zużyciu energii elektrycznej, za [18].

Pytanie	Możliwości odpowiedzi	Rozdzielczość czasowa
Czy byłeś w domu w trakcie zwolnienia lekarskiego?	Tak: Wysokie zużycie energii w ciągu dnia. Nie: Niskie zużycie energii w ciągu dnia.	Godzina/Minuta
Czy się dobrze wyspałeś tej nocy?	Tak: Brak zużycia energii w ciągu nocy przez co najmniej 6 godzin. Nie: Losowe użytkowanie energii w ciągu nocy.	Godzina/Minuta
Czy oglądałeś mecz wczoraj wieczorem?	Tak: Aktywność urządzeń odpowiadająca czasowi transmisji meczu. Nie: Brak korelacji z aktywnością urządzeń elektrycznych w czasie transmisji meczu.	Minuta/Sekunda
Czy wyszedłeś do pracy później niż zazwyczaj?	Tak: Ostatnie użytkowanie urządzeń elektrycznych później niż przewidywany przez Google Maps czas drogi do pracy. Nie: Ostatnie użytkowanie urządzeń elektrycznych z wystarczającym zakresem czasu na dotarcie do miejsca pracy.	Minuta
Czy zostawiasz dziecko samo w domu?	Tak: Aktywność odpowiadająca jednej osobie. Nie: Równoczesne użytkowanie urządzeń elektrycznych w różnych częściach domu.	Minuta/Sekunda
Czy jadasz gorące czy zimne śniadanie?	Gorące: Duża aktywność urządzeń elektrycznych rano (kuchenka mikrofalowa/ekspres do kawy/toster). Zimne: Brak porannej aktywności urządzeń elektrycznych.	Sekunda

Podsumowując, przed rozwiązaniami dla energetyki stoi szansa związana z możliwością otwierania chętnym odbiorcom dróg aktywnego zaangażowania, a pomoc w dobrym wykorzystaniu tej szansy może zapewnienie ochrony danych odbiorców na poziomie dostosowanym do ich potrzeb i oczekiwań.

Rola agregatora

Podmiot grupujący odbiorców energii nazywany jest w literaturze agregatorem ???.

Ze względu na minimalne wymagania programów DSR prowadzonych przez OSP dotyczące mocy umownej uczestników rynku, udział w przedstawionych programach DSR możliwy jest przede wszystkim w wypadku zgrupowania wielu odbiorców. Próg ten wynosi obecnie 300 kW, co przy założeniu 12,5 kW mocy umownej na mieszkanie i 30 kW mocy umownej dla domu jednorodzinnego jest porównywalne do całkowitego zapotrzebowania na moc grupy 24 mieszkań lub 10 domów jednorodzinnych.

Tworzenie takich zespołów oraz wspieranie ich członków w wypełnianiu wspólnych zobowiązań i zapewnieniu opłacalności przedsięwzięcia jest rolą agregatora. Agregator rolę swoją wypełnia łącząc

odbiorców tak, aby oferowany przez nich produkt DSR spełniał oczekiwania dotyczące parametrów usług redukcji na żądanie.

Relacje pomiędzy agregatorem, OSP oraz odbiorcami energii przedstawiono na Rys. 3.

Relacje pomiędzy głównymi interesariuszami projektu, opracowanie własne.

Celem szacowania elastyczności odbiorców jest dostarczenie agregatorowi użytecznych informacji o wypadkowej elastyczności zarządzanej przez niego grupy odbiorców.

Agregator potrzebuje tych informacji by móc przystąpić do przetargu organizowanego przez OSP, oferując produkt zgodny z faktycznymi możliwościami technicznymi członków grupy odbiorców.

Oczekiwanym przez agregatora efektem szacowania elastyczności jest opisanie elastyczności odbiorcy parametrem albo zbiorem parametrów, dającym możliwość porządkowania odbiorców rosnąco lub malejąco pod względem elastyczności. Koncepcja szeregowania odbiorców wynika z potrzeb agregatora, który dla podejmowania dobrze umotywowanych decyzji może szacować dostępną w zarządzanej grupie odbiorców elastyczność oraz pewność jej dostarczenia w razie potrzeby.

Z roli agregatora wynika dla niniejszej pracy to, że inteligentny licznik może być własnością operatora, który inwestuje u odbiorców wyrażających chęć udziału w DSR. Do takiego rozwiązania nawiązują założenia tej pracy. Wśród potrzeb agregatora znajdują się narzędzia wspomagające szacowanie, jakimi zasobami elastyczności dysponuje.

Program DSR w Polsce

Z punktu widzenia OSP w przypadkach niedoboru energii istotny jest koszt oraz szybkość reakcji podmiotów za nią odpowiedzialnych. Porównanie dostępnych strategii zapobiegania niedoborom energii i regulowania zasięgu ich skutków gdy wystąpią przedstawiono na Rys. 3 za pracą [21].

Strategie łagodzenia skutków niedoborów energii elektrycznej, za [21].

Jak zaznaczono na Rys. 3, jednym z powodów zainteresowania OSP aktywnym popytem (DSR) jest wysoka efektywność kosztowa, wyższa niż w przypadku ulepszeń w infrastrukturze.

Na polskim rynku energii OSP prowadzi trzy programy, których podstawowe warunki zestawiono w tabeli 3.3.

Tabela 3.3: Programy DSR prowadzone przez PSE, za [22].

Cecha / Program	Program gwarantowany	Program bieżący	Program bieżący uproszczony
Opłata za gotowość	Tak	Nie	Nie
Odpowiedź na wezwanie do złożenia propozycji sprzedaży	Obowiązkowo	Dobrowolnie	Dobrowolnie

Cecha / Program	Program gwarantowany	Program bieżący	Program bieżący uproszczony
Reakcja na polecenie redukcji, o ile zaakceptowano propozycję sprzedaży	Obowiązkowo	Obowiązkowo	Dobrowolnie
Obowiązek testowania	Tak	Nie	Nie
Kary za niewykonanie redukcji	Tak	Tak	Nie

Zestawione w tabeli 3.3 programy różnią się przede wszystkim poziomem odpowiedzialności odbiorcy za wykonanie redukcji, oraz związanym z tą odpowiedzialnością wynagrodzeniem.

Program bieżący uproszczony jest programem niezobowiązującym, który pozwala odbiorcy na zapoznanie się z systemem, zaangażowanie się w świadczenie usług DSR bez ponoszenia ryzyka, jak również na czerpanie zysków w przypadku wykonania redukcji na żądanie operatora, jednak bez obowiązku wykonania takiej redukcji.

Program bieżący wprowadza obowiązek wykonania redukcji i za pomyślne wypełnienie tego obowiązku oferuje wynagrodzenie finansowe, jednakże pozostawienie odbiorcy w ciągłej gotowości do redukcji nie jest w nim kompensowane finansowo.

Finansową kompensację faktu ciągłej gotowości odbiorcy do wykonania redukcji na żądanie w połączeniu z wynagrodzeniem za wykonanie redukcji oferuje program gwarantowany. Z tej perspektywy jest to program najbardziej atrakcyjny finansowo, lecz by udział w nim miał szanse przyniesienia realnego zysku potrzebna jest

Niniejsza praca skupia się na działalności agregatora w wymagającym największego zaangażowania ale i potencjalnie najbardziej dochodowym z zestawionych w tabeli 3.3 programów DSR programie gwarantowanym (PG).

PTPiREE, do którego należą najwięksi polscy OSD oraz jedyny OSP, podkreśla preferencję dla Elektronicznego obiegu dokumentów wszędzie tam gdzie to możliwe ???.

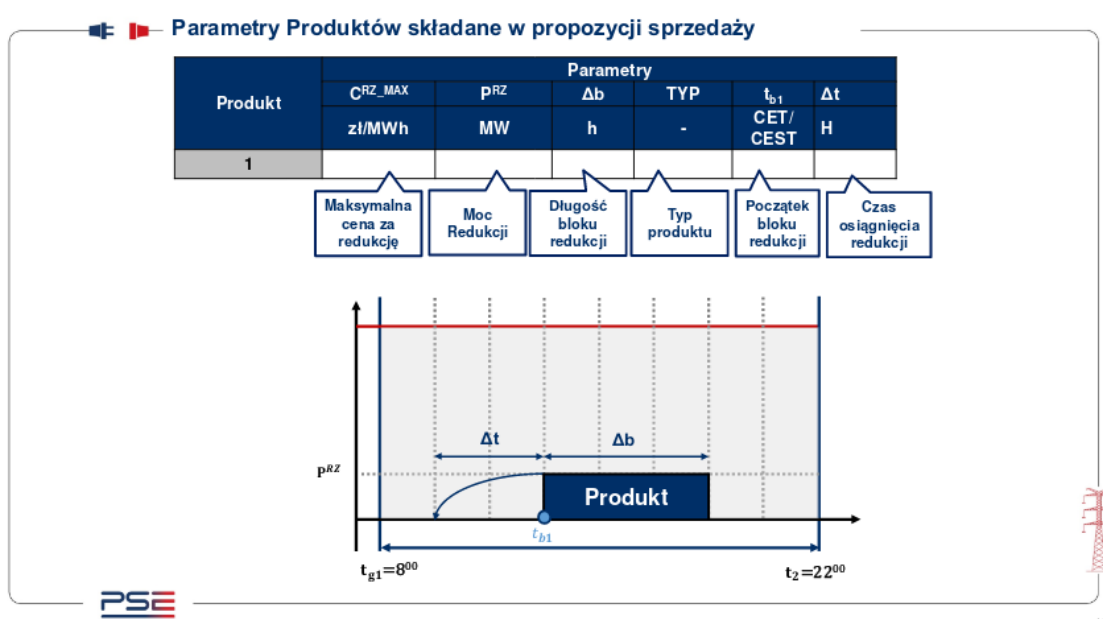
Bieżący schemat procedury kwalifikacji uczestnika rynku do programu DSR przedstawia Rys. 3. Schemat pochodzi z prezentacji aktualnych reguł aukcji DSR na polskim rynku energii elektrycznej. Celem niniejszej pracy jest opracowanie narzędzi, wspomagających pozytywną weryfikację oferty udziału w programie DSR. Na rysunku Rys. 3 oznaczono ten obszar symbolicznie zieloną linią przerywaną.

Przebieg certyfikacji elastycznego odbiorcy z zaznaczeniem obszaru zainteresowania pracy, źródło: [22]

W trakcie procesu certyfikacji, elastyczny odbiorca deklaruje kluczowe parametry swojej oferty, określając między innymi górne ograniczenie dotyczące ceny produktu dla OSP, czas redukcji oraz czas wymagany na dostarczenie produktu do systemu przesyłowego. Te parametry to:

- Maksymalna cena za redukcję C^{RZ_MAX} [zł/MWh],
- Moc redukcji p^{RZ} [MW],
- Długość bloku redukcji δb [godz.],
- Typ produktu,
- Początek bloku redukcji t_{b1} (CET/CEST),
- Czas osiągnięcia redukcji δt [godz.].

Parametry te przedstawia w sposób zbiorczy Rys. 3.1. Właściwe ich zdefiniowanie przez uczestnika programu jest kluczowe: parametry te mają wpływ na ocenę atrakcyjności oferty. W przypadku zaakceptowania oferty uczestnik musi oczywiście być gotowy do realizacji usługi zgodnie z podanymi przez siebie parametrami.



Rys. 3.1: Parametry do podania przez Obiekt Redukcji w propozycji sprzedaży, źródło: [22]

Sam problem wyboru najlepszych ofert przez OSP jest wyzwaniem. Jak wskazują autorzy pracy [13] zadanie to może być rozwiązane w następujący sposób. Oferta każdego z odbiorców, wraz z jej konkretnymi parametrami: wymiarem, ceną oraz jakością, nazywana jest dalej “produktem”. Na procedurę wyboru składają się następujące kroki:

1. Wybranie najtańszego zestawu dostępnych produktów DSR. Rozwiązanie szczególnego przypadku “problemu plecakowego” przy użyciu puli produktów oferowanych przez elastycznych odbiorców;
2. W razie niemożności skomponowania puli wystarczającej, wyznaczenie puli maksymalizującej uzyskaną moc;
3. Wyznaczenie puli maksymalizującej uzyskaną moc.

Praca skupia się na wspomaganie szacowania parametrów agregatora w oparciu o pulę możliwości zdefiniowanych przez odbiorców tak, aby zapewnić zrealizowanie usługi redukcji na wezwanie OSP. Nie rozważa się natomiast kwestii związanych z opracowywaniem tak zwanych strategii ofertowych, czyli wspomaganie takiej budowy oferty, która zmaksymalizuje szanse jej przyjęcia przez OSP.

Za czas notyfikacji przed wydaniem polecenia redukcji przyjmuje się 60 minut ???.

Rozdział 4

Projektowanie architektury systemu

Właściwy dobór narzędzi jest ważnym etapem rozwiązywania problemu. Zadaniem narzędzi w rękach projektanta jest połączenie oczekiwań z konkretnymi, ściśle zdefiniowanymi krokami, które należy podjąć dla osiągnięcia celu. Zastosowanie dobrze znanych metod i modeli formalnych w opisie jest gwarancją jakości rozwiązania, a także sprzyja rozpoznawaniu w nim wzorców konstruktywnych i destruktywnych (*patterns*, *anti-patterns*).

O znaczeniu podejścia polegającego na łączeniu prostych rozwiązań o znanych właściwościach w większe struktury według sprawdzonych modeli można wnioskować popularności, jaką cieszy się pierwsza publikacja podsumowująca dziedzinę projektowania w oparciu o wzorce projektowe [4]. Podsumowanie odwołań do tej pozycji przygotowane przez bibliotekę stowarzyszenia ACM w lutym 2019 roku zawierało przeszło 5400 prac ???.

W niniejszym rozdziale przedstawiono najważniejsze narzędzia z dziedziny projektowania architektury systemu. Zidentyfikowane użyteczne koncepcje posłużyły do opracowania rozwiązań na potrzeby tej pracy.

Podstawowy algorytm badawczo-rozwojowy

Przypomnijmy podstawową procedurę porządkującą etapy rozwiązywania problemów natury inżynierskiej.

Metoda ta składa się z następujących kroków [23]:

1. Zrozumienie problemu,
2. Zbudowanie matematycznego zrozumienia problemu,
3. Rozważenie problemu przy użyciu skonstruowanego wcześniej modelu,
4. Wprowadzenie poprawek do modelu - jeśli są potrzebne, aż w końcu
5. Znalezienie (obliczenie) rozwiązania.

ArchiMate w analizie architektur

Za podstawowy cel przyświecający wypracowaniu ujednoliconego języka opisu architektur korporacyjnych ArchiMate autorzy podają zaspokojenie rynkowego zapotrzebowania na jednolity i kompleksowy język opisu architektur korporacyjnych [24].

Standard ArchiMate dostarcza zbiór elementów wraz z najczęstszymi zależnościami pomiędzy nimi, a także odpowiadającą tym pojęciom ikonografią dla graficznego modelowania tworzonej architektury.

Specyfikacja języka ArchiMate w obowiązującej obecnie wersji 3.0.1 [24] definiuje kluczowe dla tego języka pojęcia w następujący sposób:

1. Podstawowa platforma analityczna ArchiMate (*ArchiMate Core Framework*) - struktura referencyjna, używana do klasyfikacji elementów głównego języka ArchiMate. Składa się z trzech warstw i trzech aspektów;
2. Centralny język ArchiMate (*ArchiMate Core Language*) - centralna część języka ArchiMate, definiująca koncepty i zależności dla modelowania Architektury Korporacyjnych;
3. Aspekt - klasyfikacja elementów oparta na niezależnych od warstwy cechach mających odniesienie do spraw istotnych dla poszczególnych interesariuszy. Jest on stosowany do pozycjonowania elementów w meta-modelu ArchiMate;
4. Atrybut - własność związana z elementem lub relacją w języku ArchiMate;
5. Koncept - element albo relacja;
6. Zgodność - wypełnienie określonych wymagań;
7. Zgodna implementacja - implementacja wypełniająca wymagania zgodności zdefiniowane w klauzuli zgodności w standardzie ArchiMate;
8. Element złożony - element składający się z innych elementów z rozmaitych aspektów i warstw w języku modelu;
9. Element - podstawowa jednostka w meta-modelu ArchiMate. Używa się jej do definiowania i opisu części tworzących architekturę korporacyjną oraz zbiory ich cech charakterystycznych;
10. Warstwa - abstrakcja platformy analitycznej ArchiMate, w której przedsiębiorstwo może być odwzorowane;
11. Model - zbiór konceptów w kontekście struktury językowej ArchiMate;
12. Relacja - połączenie pomiędzy konceptem źródłowym i docelowym; klasyfikowane jako: strukturalne, zależnościowe, dynamiczne lub inne.

Język ArchiMate posiada ważne zalety, które przedstawiono na Rys. 4 za grupą The Open Group [25].

Najważniejsze możliwości otwierane przez język ArchiMate, za: The Open Group.

Najważniejsze elementy języka ArchiMate to:

1. Model,
2. Koncept:
3. Element,
4. Relacja,
5. Łącznik relacyjny,
6. Element zachowania,
7. Element struktury,
8. Element motywacji, oraz
9. Element złożony.

Zależności pomiędzy tymi konceptami ilustruje za [24] Rys. 4.

Zależności pomiędzy najbardziej ogólnymi konceptami języka ArchiMate, źródło: The Open Group.

Model języka składa się z trzech warstw:

1. Warstwy biznesowej,
2. Warstwy aplikacyjnej, a także

3. Warstwy technologicznej.

Z tych trzech warstw oraz trzech aspektów, tworzony jest kompletny rdzeń platformy ArchiMate, jak przedstawiono na Rys. 4.

Struktura rdzenia platformy analitycznej ArchiMate, źródło: The Open Group.

Ogólny schemat platformy ArchiMate w najnowszej wersji 3.0.1 przedstawiono na Rys. 4 za specyfikacją [26].

Przykład użycia specyfikacji ArchiMate w modelowaniu relacji pomiędzy motywatorami działań a strategicznymi celami; autorstwo oryginału: The Open Group.

Przygotowanie modelu jest pierwszym krokiem w stronę ucieleśnienia systemu rozwiązującego konkretny problem. Im dokładniej system zostanie odwzorowany, tym wcześniej wykryjemy ukryte w projekcie wady i tym skuteczniej i precyzyjniej będzie można zaproponować poprawki. Modelowanie służy także lepszemu zrozumieniu natury problemu; jak mawiał noblista, fizyk teoretyk i pionier w dziedzinie planowania obliczeń prof. Richard Feynman - "Czego nie potrafię zbudować, tego wciąż nie rozumiem."

Język modelowania UML w wersji 2.0 służy opisowi systemów w spójny i jednoznaczny sposób. Ułatwia on w ten sposób dyskusję nad ostatecznym kształtem systemu pomiędzy osobami z różnych dziedzin życia, na przykład inżynierami, matematykami oraz przedsiębiorcami.

Modularyzacja architektury

Modułowa budowa systemu ma za zadanie ułatwić późniejsze rozbudowywanie systemu o nowe elementy. Upraszcza ona również ulepszanie i dostosowywanie istniejących elementów prototypu do własnych potrzeb. Więcej zalet modularnej budowy systemu takich jak zwiększona niezawodność i prostota diagnostyki błędów, na przykładzie prototypowego systemu monitorowania rozproszonego opartego na Bezzałogowych Statkach Powietrznych (BZP), przytacza praca ???.

Motywnym przewodnikiem wspomnianej wcześniej książki "Wzorce architektoniczne" jest preferencja kompozycji ponad polimorfizm. Autorzy książki zachęcają do tworzenia dobrze zdefiniowanych wymiennych elementów wszędzie tam, gdzie to możliwe, dla polimorfizmu zostawiając rolę rozwiązania ostatecznego - do użycia jedynie tam, gdzie jest to nie do uniknięcia.

Architektura modułowa ucieleśnia tę zasadę, pozwalając na zastępowanie w razie potrzeb elementów nowymi, a także ogranicza zasięg skutków awarii do pojedynczych komponentów i, ewentualnie, ich sąsiadów w drzewie zależności.

Pisanie mikro-usług przechowujących stan jest trudne [27]. Typowo sposób, w jaki tworzone są takie mikro-usługi by być jak najprostszymi do wdrożenia, polega na oddelegowaniu przechowywania stanu usługi do zewnętrznej bazy danych, przez co efektywnie baza danych staje się sama w sobie usługą przechowującą stan. Takie podejście oprócz względnej prostoty wdrożenia pociąga za sobą szereg negatywnych konsekwencji. Spośród tych konsekwencji najpoważniejszą jest to, że nie zawsze możliwe jest osiągnięcie w ten sposób zamierzanego rezultatu [27].

Wspomniany wyżej fizyk teoretyczny i noblista Richard Feynman przytoczył w jednym z wywiadów, że złożone rozwiązania istniejące w naturze nie są "skomplikowane", a są raczej kompozycjami złożonymi z wielu małych, nieskomplikowanych elementów pełniących nieskomplikowane funkcje [28].

Jeśli zatem szukamy czegoś, co można nazwać "esencją inżynierii", a więc i programowania, to jednym z głównych kandydatów jest *komponowanie* poprzez łączenie komponowalnych abstrakcji [29].

Mając to na uwadze możemy przyjąć swoistą “regułę modularności” i porzucić próby nadania swemu rozwiązaniu pozornej “doskonałości” poprzez dostosowywania go do wszelkich możliwych przypadków i zamiast tego skupić się na dobrym pełnieniu jednej funkcji oraz łatwości łączenia elementu z innymi, jak ma to miejsce przy projektowaniu systemów według tzw. “filozofii uniksowej” [30].

Architektura mgłowa

Niniejsza praca wykorzystuje nowatorską koncepcję architektury “mgły”. Wykorzystuje ona rozwijające się możliwości miniaturyzacji komputerów. Jej istotnym elementem są współczesne komputery jednopłytkowe, które będąc urządzeniami wbudowanymi według rozmiarów oferują programiście dostęp do konsoli tekstowej zgodnej ze standardem POSIX.

Schemat architektury “mgły”, w której pracuje zaprojektowany moduł, przedstawiono za pracą ??? na Rys. 4.

Architektura mgły, za ???

Przykładem miniaturowej platformy obliczeniowej jest komputer NanoPi, który w wersji NEO4 umożliwia nawet wspomaganie przetwarzania za pomocą zamontowanego na pokładzie GPU. Charakterystyczne jest to, że poziomy przekrój takiej jednostki ma wymiary 60x45 mm, co czyni ją niewiele większą od pudełka zapalek.

Płytki oparte na wielordzeniowych jednostkach obliczeniowych zaprojektowanych przez przedsiębiorstwo ARM oferują obsługę w pełni funkcjonalnych systemów operacyjnych, z systemem Linux włącznie. Dostępne interfejsy komunikacyjne to między innymi: Wi-Fi 802.11n oraz Bluetooth 4.0. Pamięć operacyjna to 0,5-1 GB, a rolę pamięci trwałej pełnią nośniki w postaci kart microSD. Do dyspozycji na urządzeniu mamy również interfejsy przeznaczone do komunikacji z innymi urządzeniami wbudowanymi takie jak MIPI-CSI, UART, I2S, I2C, SPI oraz GPIO.

Na tego typu urządzeniach opiera się rosnąca popularność architektury mgłowej do przetwarzania i dystrybucji informacji ???. Architektura ta zakłada przetwarzanie części informacji na inteligentnych urządzeniach znajdujących się w skrajnych punktach sieci oraz wymianę już przetworzonych informacji pomiędzy tymi urządzeniami a “chmurą”.

Ważnymi celami do wypełnienia przez architekturę “mgły” są: minimalizacja opóźnień transmisji, zapewnienie odpowiedniego pasma transmisji oraz zadbanie o odpowiedni zakres ogólności opracowywanych rozwiązań ???.

W pracy tej założono, że integralność urządzenia implementującego zaprojektowaną architekturę zostanie zabezpieczona przez agregatora. Komunikacja tego systemu sięga jednak poza urządzenie przez publiczne kanały transmisji. Rozważmy szczegółowo model zagrożenia występujący w przypadku takiej architektury za pracą ???. Komunikacja zachodzi tu przy użyciu dostępnego publicznie kanału transmisji takiego jak sieć bezprzewodowa.

Jak widzimy na Rys. 4, zakładany początkowo tor wiadomości wiedzie od dostawcy wprost do odbiorcy. Dzieje się tak gdy kanał transmisji wiadomości jest z założenia niedostępny dla osób trzecich. W praktyce jednak zarówno w przypadku transmisji bezprzewodowej, jak i przewodowej, znane są sposoby ingerowania w dowolnie wybraną transmisję z pozycji osoby trzeciej zafałszowując lub uniemożliwiając komunikację.

Normalny tok transmisji wiadomości od dostawcy bezpośrednio do odbiorcy.

W niniejszej pracy w szczególności skupiono się na ewentualności zafałszowania komunikacji. Motywacją dla priorytyzacji rozpatrzenia takiego scenariusza było założenie, iż znacznie bardziej niebezpieczne dla prawidłowego funkcjonowania systemu opartego o architekturę FMA jest naruszenie integralności komunikacji niż jej uniemożliwienie.

Na Rys. 4 widzimy schemat ingerencji niepowołanej osoby trzeciej w transmisję.

Schemat ideowy ingerencji niepowołanej trzeciej strony w komunikację pomiędzy dostawcą a odbiorcą.

Jak przedstawiono na Rys. 4, dostawca zamierza nawiązać połączenie z odbiorcą, wysyłając do niego dane po wspólnym kanale transmisji. Na drodze wiadomości zdążającej do odbiorcy pojawia się intruz, który przechwytuje wiadomość i korzystając z dostępu do kanału transmisji przekazuje do odbiorcy jej zmodyfikowaną wersję.

W praktyce przedstawiony scenariusz oznacza, że dostawca i odbiorca nie mogą bezpiecznie się komunikować bez ustanowienia prywatnego, dostępnego wyłącznie dla nich dwójga kanału transmisji.

Metoda wymiany kluczy Diffiego-Hellmana ??? opisuje praktyczne rozwiązanie problemu komunikacji za pomocą nośnika dostępnego publicznie do odczytu, zapisu i nadpisywania ????. Nadpisywanie może mieć miejsce gdy atakujący przerwie kanał transmisyjny pomiędzy dostawcą a odbiorcą, wstawiając pomiędzy nimi własne urządzenie nadawczo-odbiorcze, nie przekazujące pakietów danych bezpośrednio, lecz zamiast tego - pozwalające na bieżąco modyfikować te pakiety. W praktyce takie zachowanie oznacza zatem nadpisywanie danych rozsyłanych przez rozpatrywany nośnik nie zabezpieczony w inny sposób.

Przetwarzanie współbieżne

Przetwarzanie współbieżne pozwala wykorzystać możliwości obliczeniowe coraz bardziej popularnych współcześnie wieloprocessorowych jednostek przetwarzania danych. W obliczeniach jest ono bardzo przydatne, a w wypadku niniejszej pracy - okazało się niezbędne dla zrealizowania celu dotyczącego implementacji przetwarzania danych wrażliwych u użytkownika przy użyciu wbudowanych jednostek obliczeniowych.

Programowanie w ujęciu funkcyjnym stanowi następny po programowaniu obiektowym krok w kierunku ujednolicenia struktury oprogramowania. W programowaniu funkcyjnym chodzi o dochodzenie do nieskomplikowanych rozwiązań - prostych w zrozumieniu, diagnostyce oraz naprawach.

Podstawą dla funkcyjnego programowania jest rachunek lambda z typami prostymi, zaproponowany przez Alonzo Churcha w 1940 roku w wyniku prac w dziedzinie podstaw matematyki [31]. Przy pomocy tego rachunku znaczna część wiedzy technicznej może być wyrażana w nim bardzo naturalnie [32].

Jedną z najbardziej znanych implementacji rachunku lambda dla celów programowania komputerów jest Haskell [33] i związany z nim kompilator GHC. Niemniej jednak, niektóre elementy tworzące rdzeń funkcyjnego paradygmatu programowania takie jak wzorzec strażnika w obliczeniach czy wyrażenia listowe, od dawna już znaleźć można w popularnych językach programowania trzeciej generacji: Python oraz JavaScript.

Funkcyjne podejście do programowania posiada ważne wymagania takie jak wyższy niż w przypadku programowania w językach imperatywnych próg wiedzy, który trzeba przekroczyć by zacząć konstruować użyteczne w praktyce rozwiązania. Przekroczenie tej bariery niesie ze sobą jednak na-

stępujące korzyści, istotne z punktu widzenia inżyniera oprogramowania oraz badacza zgłębiającego dziedzinę nauk informacyjnych:

- referencyjna przeźroczystość tworzonych funkcji, skutkująca większą modularnością tworzonych rozwiązań niż w przypadku języków imperatywnych,
- wyższa czytelność i ekspresywność pisanego kodu (po przekroczeniu wyżej wspomnianego progu wiedzy), oraz
- lepsze odwzorowanie typowych, popularnych wzorców tworzenia oprogramowania w języku, skutkujące łatwiejszym i bardziej precyzyjnym zrozumieniem specjalizacji przez inżynierów oraz pozwalające architektom systemów na ogłęd systemu z wyższego poziomu abstrakcji.

Oparcie platformy systemowej na podejściu funkcyjnym pozwoliło jednej z gałęzi projektu GNU fundacji Free Software Foundation [34] otrzymać system operacyjny oferujący następujące właściwości [35]:

- rozmiar pakietu dystrybucyjnego nie przekraczający 200 MB,
- rozmiar pakietu źródłowego wynoszący 23,2 MB,
- czas kompilacji kompletnego systemu operacyjnego wynoszący kilka minut [36],
- odporną na błędy architekturę dzięki zastosowaniu transakcyjnego modelu aktualizacji pozwalającego przywrócić system do poprzedniego stanu w dowolnym momencie [37], a także dzięki uniezależnieniu skutków aktualizacji od poprzedniego stanu systemu [38];
- spójność systemu dzięki deklaratywnej naturze instalacji i konfiguracji [37],
- ujednolicone zarządzanie pakietami języków wyspecjalizowanych (*DSL*) przy wykorzystaniu pojedynczego menedżera pakietów dla środowisk deweloperskich takich jak Python czy TeXlive [37],
- korzystanie domyślnie bezpośrednio ze źródeł pakietu podczas jego instalacji, co wiąże się również z
- możliwością zastosowania wirtualizacji o *przeźroczystym* charakterze uruchamianych obiektów dzięki możliwości kompilowania ich w locie z kodu źródłowego tuż przed uruchomieniem [37].

Badania nad zarządzaniem zależnościami na sposób w pełni funkcyjny już trwały gdy projekt GNU Guix powstał jako gałąź projektu Nix. Obecnie Guix jest już oddzielnym projektem - menedżerem pakietów oferującym ponad 9 tysięcy pakietów. Guix może być wdrożony na dowolnej istniejącej dystrybucji systemu operacyjnego Linux takiej jak Debian [39].

Użytecznej techniki do strukturyzowania programów funkcyjnych dostarcza ujęcie funkcji w formie monady, a esencja wprowadzania efektów ubocznych może zostać ujęta poprzez użycie monad w czystym języku funkcyjnym [40].

Monada zwraca wartość typu *a*, albo *Nothing*, o czym ogólnie mówimy, że monada zwraca wartość typu *Maybe a*. Skutki działania monady nie są widoczne na zewnątrz, stąd monada nazywana jest również elementem procesu obliczeniowego “czystego” w sensie skutków ubocznych ???.

Zgodnie z przedstawioną definicją, monada rozumiana jest w tej pracy jako podstawowa jednostka kompozycji obliczeń i tylko w taki sposób. W niniejszej pracy celowo pominięto matematyczny rygor związany z pochodzeniem i definicją monady na rzecz skupienia się na korzyściach płynących z wykorzystania monad jako elementarnej jednostki kompozycji programów komputerowych.

W dziedzinie projektowania potoków przetwarzania danych monady są cenne ze względu na dostępność dobrze zdefiniowanych reguł ich komponowania, co z kolei znacznie upraszcza projektowanie, implementację oraz następujące potem przetwarzanie danych oparte na monadach, a także ułatwia często bardzo trudne zapewnienie spójności pomiędzy tymi trzema aspektami w czasie [29].

Funkcje wykonujące dużą ilość rozmaitych zadań są trudne do przetestowania. Aby zredukować ograniczenia możliwości rozwoju tworzonego oprogramowania, dobrą drogą jest redukcja złożoności tego oprogramowania w ważnych dziedzinach jego powstawania.

Program napisany w języku funkcyjnym i skompilowany przy użyciu kompilatora GHC wykonuje się z prędkością porównywalną do programów napisanych w językach C czy C++. Różnica na korzyść C/C++ wydaje się być znaczniejsza w zastosowaniach, w których program przez większość czasu wykonuje bardzo małą część kodu, co ma miejsce między innymi w koderach wideo. Tym niemniej, kodery takie zawierają często rutyny niezależne od siebie, a więc takie, które teoretycznie można wykonywać równolegle. W przypadku zastosowania równoległego wykonywania takich funkcji, korzyść z użycia języka C/C++ przegrywa z modularnością u łatwością zrównoleglania kodu w językach funkcyjnych takich jak Haskell [41].

Częstym przypadkiem jest, że rozważana aplikacja poświęca 80% czasu na wykonywanie 20% kodu. Oznacza to, że duża część nakładu roboczego aplikacji pochodzi często z jednego lub ewentualnie kilku miejsc, tzw. “wąskich gardeł”. Odpowiednie zastosowanie profilowania aplikacji i następujące po nim przepisanie wytypowanych funkcji z języka funkcyjnego do języka C w celu późniejszego wykorzystania ich w głównym module zaimplementowanym w języku Haskell może zapewnić aplikacji zarówno stabilność, szybki rozwój, jak i łatwość późniejszego utrzymania i dalszej rozbudowy. Obecnie czas pracy programisty jest przecież kosztowniejszy niż czas pracy procesora [41]. Co więcej tam, gdzie optymalizacja algorytmu ma większy wpływ na ogólną wydajność aplikacji, szybszy rozwój aplikacji w języku Haskell w porównaniu do platformy C++ pozwala teoretycznie zapewnić więcej czasu na rozważania samego algorytmu [41].

Programowanie funkcyjne, mimo swoich wyzwań, niesie ze sobą ważne praktyczne zalety. Wszechobecna rekurencja pozwala na zgrabne rozwiązywanie wielu powszechnych problemów; programowanie przy użyciu stałych zapobiega całej klasie trudnych do wyśledzenia błędów; oparcie struktury rozwiązania na funkcjach pozwala na uniknięcie rozrostu “szytych na miarę” klas i interfejsów na rzecz automatycznego stosowania powtarzalnych, sprawdzonych wzorców do ich budowy jedynie w razie potrzeby; rygor typów natomiast znacznie pomaga w tworzeniu niezawodnego i prostego do analizy i modyfikacji oprogramowania [41].

Transformacja rozwiązania imperatywnego w deklaratywne może na przykład przebiegać następująco, zgodnie z formułą zaproponowaną przez Joshuę Backfielda [42]:

1. Wprowadzenie funkcji wyższego rzędu,
2. Konwersję istniejących metod w funkcje czyste,
3. Konwersję pętli do rekursywnych/ogonowo rekursywnych metod (jeśli możliwe),
4. Konwersję zmiennych mutowalnych w niemutowalne,
5. Użycie dopasowania wzorców (jeśli to możliwe).

Znaczne korzyści płyną już z postawienia pierwszego z tych kroków, gdzie to zyskujemy natychmiast możliwość wykorzystania ponownego użycia kodu. Po zastosowaniu takiego podejścia mamy szansę zacząć zauważać pętle, gdzie możliwe będzie wyabstrahowanie logiki rekursywnej do bardziej powtarzalnej i modularnej postaci funkcji wyższego rzędu przekazywanej do logiki zapętlenia [42].

W modelowaniu współbieżnych procesów pomocny jest model zwany sieciami Petriego ????. Kluczową zaletą tego modelu jest jego dwojaka natura - z jednej strony matematyczna, a z drugiej graficzna ????. Oznacza to, że każdy model matematyczny sporządzony z użyciem tego formalizmu możemy zilustrować graficznie, a każdy model graficzny możemy analizować z użyciem automatycznych narzędzi matematycznych.

Rozdział 5

Koncepcja rozwiązania

Z analizy wymagań dotyczących szacowania energii płyną wnioski, że potencjał aktywizacji strony odbioru jest duży, a mali detaliczni konsumenci mogliby uczestniczyć w DSR przy wsparciu reprezentującego ich agregatora. Potrzeba narzędzi, które umożliwiłyby oszacowanie wielkości tego potencjału.

Zaprojektowana architektura systemu, nazywana dalej *FMA* (*Flexibility Measurement Architecture*), uwzględnia potrzeby odbiorcy oferując mu narzędzie wspomagające go w szacowaniu własnej elastyczności. Agregatorowi architektura FMA daje możliwość zautomatyzowania procesu szacowania elastyczności odbiorców, zgodnie ze zidentyfikowanymi wcześniej wymaganiami.

Architektura FMA uwzględnia możliwość zainstalowania modułu obliczeniowego FlexMeter, dla którego prototypowy moduł programowy FlexCheck opracowano w ramach tej pracy, zarówno w siedzibie Agregatora, jak i w siedzibie odbiorcy. Niezależnie od tego faktu, system może pracować w dwóch trybach: odbiorcy oraz agregatora.

Oparty na architekturze FMA system nosi nazwę *FSCP* (*Flex Self-Check Plus*), a jego schemat przedstawiono na Rys. 5.

Fizyczne składowe systemu FSCP w architekturze FMA, opracowanie własne.

Komponenty wchodzące w skład systemu przedstawiono bardziej szczegółowo na Rys. 5.

Składowe zaprojektowanego autorskiego rozwiązania FSCP w architekturze FMA.

Na system składają się następujące elementy (moduły):

- Bufor danych o zużyciu w postaci licznika inteligentnego zainstalowanego w siedzibie odbiorcy,
- Urządzenie FlexMeter, a w nim:
 - Komponent buforujący pochodzące od agregatora definicje scenariuszy, dane meteorologiczne oraz prognozy zużycia energii dla odbiorcy,
 - Moduł scenariuszy, czyli klasyfikator identyfikujący scenariusze na podstawie danych od agregatora,
 - Nierelacyjna baza danych,
 - Moduł programowy *FlexCheck* szacujący elastyczność,
 - Interfejs API wykorzystywany przez aplikację odbiorcy,
 - Interfejs API udostępniający dane agregatorowi.
- Aplikacja przeglądarkowa wykorzystująca interfejs API odbiorcy,

Połączenie elementów systemu FSCP z istniejącą infrastrukturą odbiorcy odzwierciedlono na Rys. 5.

Diagram wdrożenia pakietu SCFP.

Ideę instalacji zaprojektowanego w ramach tej pracy modułu FlexMeter, którego szczegółowy opis znajduje się w podrozdziale 5.4, zilustrowano na Rys. 5. Polega ona na instalacji blisko licznika energii z dostępem do łączności internetowej. Urządzenie przedstawione na obrazku reprezentuje moduł FlexMeter.

Idea instalacji instalacji modułu sprzętowego FlexMeter z zainstalowanym oprogramowaniem FlexCheck; w opracowaniu wykorzystano oryginalną fotografię pochodzącą z pracy ???.

Indeks elastyczności

Podstawową właściwością odbiorcy, który mógłby aktywnie uczestniczyć w rynku energii jest jego zdolność do modyfikacji własnego zużycia energii, określana jako elastyczność ???. Poprzez profilowanie odbiorców rozumiemy przypisanie każdemu odbiorcy zestawu parametrów opisujących jego elastyczność. Celem w wyborze zbioru parametrów do opisu profilu użytkownika było umożliwienie agregatorowi praktycznego wnioskowania o elastyczności grupy wielu odbiorców na podstawie danych pochodzących od każdego odbiorcy z osobna.

Na początku rozpatrzono indeks elastyczności, występujący dalej również pod symbolem FI. Jest on koncepcją pozwalającą na najprostsze opisanie elastyczności danego odbiorcy. Wartość FI obliczania jest za pomocą formuły 5.1.

$$FI = 1 - \frac{E_1}{E_0} \quad (5.1)$$

We wzorze 5.1, wartość E_0 oznacza typowe zużycie energii w danym scenariuszu, a wartość E_1 symbolizuje rzeczywiste zużycie zmodyfikowane na żądanie.

Podany wzór 5.1 dla celów obliczeń uszczegóławiamy dla wyróżnienia wartości dyskretnego szeregu czasowego zawierającego wyniki pomiaru mocy pobranej przez odbiorcę w jednostce czasu. Rezultatem jest formuła 5.2.

$$FI = 1 - \sum_{n=1}^{24} \frac{E_1[n]}{E_0[n]} \quad (5.2)$$

Formuła 5.2 wymaga częstego dzielenia liczb zmiennoprzecinkowych, dlatego w obliczeniach zastosowano ostatecznie formułę 5.3, w której operacje dzielenia zostały wyodrębnione do wykonania jedynie raz w pojedynczym procesie obliczeniowym.

$$FI = 1 - \frac{\sum_{n=1}^{24} E_1[n]}{\sum_{n=1}^{24} E_0[n]} \quad (5.3)$$

Poprzez przekształcenie formuły 5.2 do postaci 5.3 uzyskano poprawę wydajności oraz precyzji obliczeń.

Operacji dzielenia unikamy także w zastosowaniu jako miary elastyczności miary FL opartej na porównaniu obydwu przebiegów:

$$FL = E_1 - E_0 \quad (5.4)$$

Indeks elastyczności w przypadku tej miary ma postać:

$$FLI = \sum_{n=1}^{24} E_1[n] - E_0[n] \quad (5.5)$$

Zarówno w przypadku FI, jak i FLI, uogólnienia dokonuje się poprzez wyznaczenie wektora złożonego z wartości indeksu elastyczności w poszczególnych godzinach doby.

Prototypowe rozwiązanie zostało dla uproszczenia oparte na mierze FL/FLI, a podstawą wyboru obydwu z przedstawionych miar była oferowana przez nie możliwość demonstrowalnego dla odbiorcy szacowania elastyczności w odniesieniu do przebiegów przez niego deklarowanych lub przewidywanych przez agregatora.

Interesariusze

Na Rys. 5 przedstawiono przykładowe zadania odbiorcy oraz agregatora reprezentującego tego odbiorcę. Zilustrowana została możliwość szacowania własnej elastyczności w kontekście przykładowych zadań agregatora przy zgłaszaniu gotowości do świadczenia usługi DSR dla Operatora Systemu Przesyłowego (OSP).

Zadania składowe procesu wyboru usługodawców DSR przez OSP.

Składowe systemu

Poszukiwana była architektura dla systemu umożliwiającego zautomatyzowane profilowanie konsumentów energii elektrycznej pod kątem elastyczności.

Na Rys. 5 widzimy połączenia pomiędzy elementami pakietu oraz wyróżnione podsystemy.

Połączenia pomiędzy komponentami zestawu Flex Self-Check Plus.

Na Rys. 5 widzimy połączenia pomiędzy elementami pakietu oraz wyróżnione podsystemy.

Podsystemy FSCP wraz z połączeniami między nimi.

Następujące elementy systemu są istotne z perspektywy odbiorcy:

1. Aplikacja odbiorcy: może być to aplikacja mobilna albo internetowa. W praktyce możliwe jest połączenie wykorzystania obu tych technologii niskim kosztem przy użyciu interfejsów API.
2. Urządzenie FlexMeter: łączność z tym urządzeniem musi być dostępna w celu przeprowadzenia iteracji szacowania elastyczności, zwanej dalej próbą elastyczności.

Dzięki zastosowaniu w węźle konsumenckim w pełni funkcjonalnego komputera, rozwiązanie oparte o architekturę FMA może pracować pod kontrolą systemu operacyjnego ogólnego zastosowania takiego, jak Microsoft Windows czy Linux. Pozwala to na wykonywanie nietrywialnych obliczeń na miejscu, tj. bez udostępniania na zewnątrz danych podlegających specjalnej ochronie.

Moc obliczeniowa, którą mamy do dyspozycji na CPU platformy, wynosi 1,73 GFLOPS Dla OrangePi PC z procesorem Quad-Core ARM H3. Na pokładzie można zainstalować również układ GPU oferujący dodatkowe 142 GFLOPS mocy obliczeniowej. Ma to miejsce na przykład w przypadku komputera jednopłytkowego Odroid-XU4.

Gdy klasyfikator scenariuszy rozpozna w bieżących danych warunki spełniające założenia określone w dowolnym ze scenariuszy wybranych uprzednio przez odbiorcę jako go interesujący, odbiorca jest informowany o możliwości weryfikacji własnej elastyczności poprzez aktywne dostosowanie odbioru do założeń wskazanych w wybranych w aplikacji odbiorcy odpowiedziach do scenariuszy.

Rozpoczęcie korzystania przez odbiorcę z systemu FSCP.

Odbiorca przystępujący do programu szacowania elastyczności podejmuje kroki przedstawione na Rys. 5.

Proces oceny elastyczności na schemacie blokowym.

Iteracja oceny elastyczności odbiorcy na schemacie blokowym.

Platforma umożliwia również zlecenie bardziej wymagających obliczeń na zewnątrz przy wykorzystaniu architektury mikro-usługowej.

Urządzenie FlexMeter

W ramach weryfikacji implementowalności architektury na istniejącym sprzęcie przeprowadzone zostały przez autora testy z użyciem docelowej platformy sprzętowej. Wykorzystany zestaw roboczy przedstawiony został na Rys. 5.

Sprzętowa część zestawu roboczego programisty urządzenia FlexMeter.

System opiera się na zdarzeniach. Po otrzymaniu sygnału o potrzebie redukcji odbiorca ma szansę zredukować swoje zapotrzebowanie na energię w nawiązaniu do swych wcześniejszych deklaracji. System weryfikuje, czy zmiana zapotrzebowania odbiorcy rzeczywiście osiągnęła oczekiwany poziom, a także czy jej charakterystyka była zgodna z deklaracją odbiorcy.

Profil poboru odbiorcy o sztywnej charakterystyce zużycia energii przedstawiono na Rys. 5. Czerwona linia na wykresie oznacza stan linii sygnalizującej zapotrzebowanie na odpowiedź odbiorcy. Odpowiedź odbiorcy może przyjmować postać jednej z charakterystyk zaprezentowanych na Rys. 3, w zależności od wcześniej zgłoszonej przez niego deklaracji.

Odbiorca o sztywnym profilu zużycia.

Dla porównania, na Rys. 5 widzimy odpowiedź użytkownika elastycznego na otrzymany monit o zapotrzebowaniu na redukcję.

Odbiorca o elastycznym profilu zużycia.

Jak pokazano na Rys. 5, odbiorca elastyczny odpowiedział aktywnie na otrzymaną informację o zapotrzebowaniu na redukcję i ograniczył swoje zapotrzebowanie do 45% na czas kilku godzin. Tę informację, oraz jej uszczegółowienie w poszczególnych godzinach, dostarcza urządzenie FlexMeter. Pozwala ono także na pomiar elastyczności z uwzględnieniem osobno zdefiniowanego scenariusza.

Urządzenie FlexMeter programowane jest w środowisku konsoli systemu operacyjnego typu GNU/Linux, przy pomocy portu szeregowego UART. Konsolę portu szeregowego do programowania urządzenia ilustruje Rys. 5.

Konsola programisty urządzenia docelowego.

W pracy tej założono, że strona implementująca zadba o dostarczenie danych z licznika do urządzenia FlexMeter przez protokół Bluetooth albo przy pomocy bezprzewodowej sieci lokalnej.

Przykładową implementacją takiej komunikacji jest nakładka na licznik OneMeter, której producent urządzenia oferuje klientom biznesowym dostęp do API urządzenia ???.

Od strony technicznej, nakładka OneMeter pobiera dane z portu optycznego licznika i udostępnia je poprzez interfejs Bluetooth aplikacji mobilnej. Urządzenie oferuje również wysłanie danych do chmury w celu przechowywania i analizy. Maksymalna rozdzielczość czasowa pomiaru przez OneMeter wynosi 15 minut.

Nakładka na licznik OneMeter jako potencjalne źródło danych dla systemu Self Flex-Check Plus, źródło: MamStartup/OneMeter ???.

Architektura FMA przewiduje opcję przetwarzania danych dotyczących zużycia w siedzibie odbiorcy.

Urządzenie przewidziane do tego celu oparte jest na procesorze w architekturze ARM. Ma ono wymiary pozwalające na umieszczenie go w obudowie licznika energii bez znacznego zwiększania wymiarów licznika, lub też na obudowie w postaci oddzielnego modułu.

Do tego celu wybrano urządzenie NanoPi NEO przedstawione na ilustracjach w ramach Rys. 5.1.

(a) Strona slotu kart pamięci, źródło: Elty.pl.

(b) Zestawienie z dłonią, źródło: Forbot.pl.

Rys. 5.1: Jednopłytkowy komputer NanoPi NEO: strona slotu kart pamięci (a) i orientacyjne porównanie wymiarów z dłonią (b).

Schemat komputera NanoPi NEO z zaznaczeniem wymiarów, źródło: FriendlyElec.

Opisywane urządzenie przedstawiono jako przykład platformy zdolnej do wypełnienia przyjętych założeń dotyczących funkcjonalności systemu. Na etapie wdrożenia możliwy jest wybór innego urządzenia opartego o architekturę ARM wyspecjalizowanego w pod względem wymiarów, mocy obliczeniowej czy zapotrzebowania na energię.

Moduł scenariuszy

Dla agregatora istotne są przejawy elastyczności popytu jedynie w niektórych okolicznościach, których pojedynczy zbiór w pracy tej określany jest mianem *scenariusza* i oznaczany symbolem S , stąd też elastyczność odbiorcy rozpatrywana jest wyłącznie w kontekście konkretnego scenariusza określonego przez agregatora. Rolą modułu scenariuszy jest zapewnienie, by odbiorca mógł szacować swą elastyczność wyłącznie w ściśle określonych warunkach środowiska.

Określone przez zamawiającego scenariusze związane są z powtarzalnymi wzorcami meteorologicznymi, które mogą wpływać na ciągłość dostaw energii elektrycznej. Identyfikacja tych wzorców ma na celu wyznaczenie prawdopodobieństwa wystąpienia niedoboru energii w ramce czasu, której dotyczyć ma próbne żądanie redukcji zapotrzebowania, przez co potrzebne w niej są informacje takie, jak: średnia temperatura powietrza, historia opadów czy względna wilgotność [43]. W projekcie pod uwagę wzięta została również możliwość zastosowania gotowego mechanizmu synoptycznego dostarczonego z zewnątrz.

Ilustrację strumienia danych jako sieci Petriego ??? przedstawiono na Rys. 5. Dane pogodowe pobierane są z API ??? udostępnianego przez serwis AccuWeather.com.

Przepływ danych pomiędzy elementami podsystemu odpowiedzialnego za monitorowanie warunków otoczenia.

Opis algorytmu rozpoznawania scenariuszy do celów szacowania elastyczności przedstawiono na Rys. 5.

Rozpoznawanie scenariuszy w oparciu o dane od agregatora.

Strukturę danych odpowiadającą za obsługę scenariuszy przez moduł programowy ilustruje Listing 5.1.

```
1 class ScenarioOn {
2 public:
3     ScenarioOn(Weather_log& h, Scenario& s);
4     ~ScenarioOn();
5     add_scenario(Scenario& s);
6     remove_scenario( String & name);
7     const Scenario  get_likeliest_scenario (Weather_log& h);
8     const bool scenario_on(Weather_log& h);
9 }
```

Listing 5.1: Główne funkcje klasy scenariusza.

Wraz z prawdopodobieństwem wystąpienia zaburzeń dostaw wyznacza się tu wartość scenariusza - *w*, która sygnalizuje odbiorcom strategiczną istotność wypełnienia przez nich założeń owej redukcji, a w wypadku żądań w trybie operacyjnym określa wymiar wynagrodzenia dla podmiotów, które z sukcesem wypełnią swoje zobowiązania dotyczące wymiaru i czasu trwania redukcji.

Klasyfikacja na podstawie scenariuszy zakłada dostarczenie do modułu ewaluacji danych o pogodzie obejmujących dni wcześniejsze, stan bieżący oraz prognozę na najbliższe godziny.

Rosnąca dokładność przewidywań pogody ma znaczenie dla pracy, ponieważ wraz ze wzrostem dokładności danych o pogodzie oraz jej przewidywania agregator może w sposób bardziej precyzyjny, czyli przy pomocy zmiennych opisujących warunki meteorologiczne, definiować interesujące go scenariusze, w których bieżąca sytuacja pod uwagę skorzystanie z usługi DSR.

Dokładniejsze dane oznaczają również łatwiejsze rozpoznawanie podobieństw i różnic pomiędzy scenariuszami, w których użytkownik był certyfikowany, a scenariuszami, jakich agregator spodziewać się powinien na przestrzeni najbliższych godzin.

Moduł FlexCheck

Główne moduły programowe systemu umieszczone są na urządzeniu zainstalowanym w siedzibie użytkownika. Pobierają one od użytkownika informacje o jego poborze energii. Informacje te są porównywane z danymi zawartymi w scenariuszach określanych przez zamawiającego, a na podstawie tych dwóch zestawów danych moduł dostarcza obydwu interesariuszom potrzebne im informacje w postaci obliczonego wskaźnika *FI*. Każdy z programowych modułów obliczeniowych przedstawiony został jako klasa przy pomocy języka programowania obiektowego C++.

Zadaniem modułu jest szacowanie elastyczności odbiorcy. Przykładem takiego oszacowania jest algorytm przedstawiony na Rys. 5.

Weryfikacja deklaracji elastyczności odbiorcy w danym scenariuszu.

Dopasowanie profilu odpowiedzi odbiorcy do deklarowanej dla danego scenariusza poglądowo ilustruje blokowy schemat Rys. 5. Realizowany jest w ten sposób cel szczegółowy polegający na wsparciu dopasowania reakcji odbiorcy do reprezentatywnego profilu elastycznego zużycia.

Dopasowanie elastycznej reakcji odbiorcy do najbliższego profilu wzorcowego.

Struktury danych

Struktury danych systemu zostały przedstawione na Rys. 5.

Diagram klas architektury FMA.

Na diagramie klas na rysunku Rys. 5 widzimy klasy przeznaczone do przechowywania danych o głównych interesariuszach. W zainteresowaniu odbiorcy leży scenariusz zdobycia certyfikatu na swoją usługę, a następnie dostarczenie tego certyfikatu dostawcy celem potwierdzenia wiarygodności swojej oferty.

Wzór implementacji

Należy zauważyć, że ze względu na trwające prace nad rozwojem i udoskonaleniem poszczególnych aspektów pomiaru elastyczności, szczegóły sposobu jej wyznaczania mogą podlegać modyfikacjom stosunkowo często.

W odpowiedzi na tę prawidłowość projekt przewiduje definiowanie dodatkowych, alternatywnych metod szacowania elastyczności, które następnie mogą być wykorzystane w miejscu tych przewidzianych na początku. Umożliwia to zastosowany model klasy, w której interfejsy publiczne są względnie stałe i precyzyjnie określone, natomiast szczegóły implementacji mogą być zmieniane w zależności od potrzeb. Dzięki zastosowaniu modelu klasy, który w tych okolicznościach może być przedstawiony jako wzorzec tzw. metody szablonowej, - zmiany szczegółów implementacji nie mają wpływu na spójność systemu jako całości.

W opisie zastosowano składnię języka C++ ze względu na jego zaletę w postaci klarowności zapisu typów zmiennych. Docelowo uwzględniono implementację modułu z użyciem innego języka programowania z możliwością programowania obiektowego takiego jak Python lub JavaScript. Możliwość tą wykorzystano w konstruowaniu prototypu modułu.

```
1 class Flexibility {
2 // private :
3 // (...)
4 public:
5     Flexibility ();
6     ~ Flexibility ();
7     float index(UsageHistory& h, Scenario& s, Prediction & p);
8     float index(UsageHistory& h, Scenario& s, Prediction & p);
9     int function (UsageHistory& h, Scenario& s, Prediction & p);
10    int function (UsageHistory& h, Scenario& s, Prediction & p);
11    int submit_index(Scenario& s, float index);
```

```

12     int submit_function(Scenario& s, float * function);
13 }

```

Listing 5.2: Publiczne funkcje programowego modułu pomiaru elastyczności.

Listing 5.2 przedstawia klasę implementującą pomiar elastyczności na podstawie danych pobranych od obydwu interesariuszy.

Zależności pomiędzy głównymi składowymi oprogramowania rozwiązania przedstawia Rys. 5.

Struktura zależności głównych pakietów oprogramowania.

Komunikacja zewnętrzna

Projektując interakcje pomiędzy komponentami systemu spotykamy następujący problem: urządzenie FlexMeter potrzebuje do pracy danych pobranych od usługi udostępniania scenariuszy, a ponadto także informacji o scenariuszach interesujących danego odbiorcę, bieżących warunkach pogodowych i w końcu - w przypadku pomyślnego przejścia danej próby - dostępu do bazy certyfikatów elastycznych odbiorców energii.

Sytuacja ta oznacza potrzebę kontaktu z łącznie czterema komponentami. W przypadku bezpośredniego powiązania urządzenia z tymi komponentami konsekwencje obejmują między innymi trudność w razie potrzeby przyszłego wprowadzenia zmian do interfejsów programistycznych tych urządzeń.

Lepszym wyjściem jest w tej sytuacji zastosowanie tzw. prawa Demeter. Prawo Demeter, inaczej zwane regułą ograniczenia interakcji, jest zaliczane do popularnych wzorców projektowych. Reguła ta służy wytwarzaniu oprogramowania o modułowym charakterze, z elementami, które można łatwo wymieniać i modyfikować według potrzeb.

Reguła ograniczenia interakcji głosi, że skupić się należy na utrzymaniu liczby powiązań pomiędzy modułami blisko niezbędnego minimum **???**. Korzyści, które niesie ze sobą stosowanie reguły ograniczenia interakcji często okazują się nieocenione w świecie często zmieniających się wymagań i technicznych ograniczeń, a skutkami jej stosowania są typowo rozwiązania łatwiejsze w rozwoju i utrzymaniu oraz odporne na błędy.

Zapewnienie odporności systemu na lokalne awarie wymaga, aby w przypadku potrzeby komunikacji z wieloma rozłącznymi jednostkami ograniczyć liczbę jednostek kontaktowanych bezpośrednio do najbliższych, a dalsze połączenia nawiązywać za ich pośrednictwem. Dzięki zastosowaniu w ten sposób “prawa Demeter”, otrzymujemy system, którego komponenty są zależne jedynie od swych sąsiadów, co niesie ze sobą ważne zalety takie, jak: odporność systemu na awarie pojedynczych komponentów, prostotę zastępowania elementów nowymi, a także swobodę w modyfikowaniu wewnętrznych struktur wybranych komponentów na przykład dla poprawy ich wydajności.

Komunikacja między urządzeniami w architekturze FMA została zaprojektowana w oparciu o opisaną regułę. Wykorzystywane są w tym celu sieciowe interfejsy API, które pozwalają na dostosowywanie formatów wymiany danych wraz z ewolucją systemu, przy jednoczesnym zapewnieniu zgodności formatów wymiany danych pomiędzy poszczególnymi interfejsami. Taka elastyczność wynika z możliwości wersjonowania interfejsów oraz udostępnianej przez model REST API swobody dołączania dodatkowych danych do komunikatów bez zaburzania podstawowej funkcjonalności systemu.

Wyjściowy przegląd sekwencji występujących w procesie szacowania elastyczności odbiorcy ilustruje Rys. 5.

Diagram sekwencji.

Urządzenie pomiarowe zbierające dane przesyła informacje do swojego operatora.

Jego działanie jest podobne do sposobów funkcjonowania pozostałych rozwiązań tego typu na rynku. Mimo to, działanie systemu FlexMeter różni się od istniejących w istotnym z punktu widzenia prywatności użytkownika aspekcie.

FlexMeter zwiększa prywatność odbiorcy przetwarzając jego najbardziej wrażliwe dane na miejscu, na zewnątrz wysyłając jedynie wnioski z takiej analizy. Kluczowe dane dostarcza zatem nie odbiorca - dostawcy, lecz rzecz dzieje się na odwrót. Dodatkowo, ryzyko ataku jest niwelowane przez zastosowanie standardów ochrony danych przed zafałszowaniem czy podszywaniem się.

W przypadku zaprojektowanej architektury głównym zagrożeniem jest przejęcie lub nieuprawniona modyfikacja danych transmitowanych przez dostawcę albo odbiorcę. Wartości, którym zagraża taki obrót zdarzeń nazywamy poufnością oraz integralnością danych.

Przedstawiony problem dotyczy komunikacji dostawcy z urządzeniem FlexMeter zamontowanym w strefie pod bezpośrednim nadzorem Użytkownika, a poza jurysdykcją dostawcy. Projekt oparty jest na założeniu, że nie zostanie naruszona wewnętrzna spójność urządzenia FlexMeter. Dzięki zastosowaniu metody Diffiego-Hellmana kanał komunikacji pomiędzy dostawcą a odbiorcą bezpieczeństwo komunikacji zapewnione jest nawet w wypadku gdy sam kanał transmisyjny może być dowolnie podsłuchiwany, zakłócany czy wypełniany danymi przez niepowołaną третią stronę.

Metoda Diffiego-Hellmana przewidziana do zabezpieczenia komunikacji agregator-odbiora pozwala obydwu stronom ustanowić taki kanał przy użyciu publicznego nośnika pomimo dostępu do niego osób trzecich.

Metoda Diffiego-Hellmana jest obecnie powszechnie używana, między innymi jako część fundamentalnych dla bezpiecznej wymiany informacji przez Internet protokołów Secure Shell (SSH) ??? oraz Secure Sockets Layer (SSL) ???.

Interfejs odbiorcy

Podstawowe wymagania funkcjonalne dotyczące interfejsu użytkownika to:

1. Odseparowanie trzech ról dostępu do danych: odbiorcy, dostawcy oraz technika;
2. Umożliwienie odbiorcy wglądu do listy scenariuszy: dostępnych oraz zadeklarowanych;
3. Umożliwienie odbiorcy zmianę deklaracji interesujących go scenariuszy;
4. Wyświetlanie obok wypełnionych scenariuszy odpowiedniej informacji, i w końcu
5. Możliwość eksportu podpisanego cyfrowo certyfikatu przez danego użytkownika.

Interfejs użytkownika opracowany został według wzorca Model-Widok-Kontroler. Do zastosowania w roli widoku na etapie implementacji rozważono komponenty: React, Angular oraz Vue - z preferencją dla tego ostatniego ze względu na najmniejszy rozmiar wersji dystrybucyjnej przy zachowaniu kluczowej funkcjonalności obydwu konkurentów oraz budowy zorientowanej na rozbudowę przy pomocy elementów dostarczanych z zewnątrz.

Scenariuszami, w których bierze udział użytkownik, może on zarządzać poprzez interfejs WebUI pokazany na Rys. 5.

Strona logowania do przeglądarkowego interfejsu użytkownika - układ strony mobilnej.

W przypadku zgłoszenia zapotrzebowania przez Operatora, Użytkownik otrzymuje informację poprzez aplikację. Wygląd informacji przedstawiony został na Rys. 5.

Przychodząca informacja o zapotrzebowaniu na redukcję - widok ekranu na telefonie.

Za przedstawionym na Rys. 3 zestawieniem charakterystyk, użytkownik ma możliwość wybrania jednej z nich jako swoją odpowiedź na przedstawiony mu scenariusz.

Wybór interesujących użytkownika scenariuszy, w których gotów jest reagować.

Wybór przez użytkownika odpowiedzi dla wskazanych w poprzednim kroku scenariuszy.

Poziomy prywatności

W świetle wniosków z przeglądu wypowiedzi odbiorców przytoczonych w rozdziale dotyczącym elastycznego odbioru, opracowana architektura uwzględnia indywidualne potrzeby odbiorców, oferując im możliwość wyboru miejsca przetwarzania ich danych.

Zaleca się podzielenie użytkowników przez agregatora na dwie grupy o następujących cechach charakterystycznych:

1. Grupa pierwsza udostępnia agregatorowi pełne informacje o własnym zużyciu energii z godzinową rozdzielczością,
2. Grupa druga udostępnia agregatorowi jedynie dane o własnej elastyczności.

W przypadku grupy pierwszej przewidziane jest użycie modułu oprogramowania Flex na infrastrukturze agregatora. Agregator udostępnia wtedy odbiorcy aplikację internetową umożliwiającą przeglądanie danych o zużyciu i elastyczności.

W grupie drugiej przetwarzanie danych o zużyciu energii odbywa się lokalnie w domu, biurze lub zakładzie produkcyjnym. Aplikacja mobilna komunikuje się w tym przypadku z urządzeniem FlexMeter przy pomocy protokołu komunikacji bezpośredniej typu Bluetooth, albo z wykorzystaniem lokalnej sieci bezprzewodowej.

Urządzenie zaproponowane w podrozdziale ?? jest przykładem wypełnienia podstawowych wymagań architektury FMA dla celów przetwarzania danych u odbiorcy.

W praktyce rynku energii elektrycznej optymalne może się okazać użycie osobnego wyspecjalizowanego typu urządzenia do obsługi każdej z grup konsumentów. W niniejszej pracy skupiono się na funkcjonalności, która następnie może zostać przeniesiona na urządzenie o własnościach odpowiadających wymaganiom docelowej grupy takim jak moc obliczeniowa, cena, wymiary, czy metoda zasilania.

Podsumowując, uwzględnienie podziału odbiorców na grupy o opisanych charakterystykach pozwala na połączenie dotychczasowych metod przetwarzania danych przez podmioty związane z energetyką z umożliwieniem przetwarzania danych najbardziej wrażliwych bezpośrednio u odbiorcy. Konsekwencją powstania drugiej ze wspomnianych możliwości jest z kolei szansa na aktywację w usługach DSR obiektów o ograniczonej lub niestabilnej łączności sieciowej.

Charakterystyczną cechą architektury FMA jest przechowywanie i przetwarzanie wrażliwych danych odbiorcy przy użyciu urządzeń zainstalowanych u niego samego. Do realizacji tego zadania przewidziano opisywaną dalej kombinację elementów.

Ograniczenia binarnej arytmetyki zmiennoprzecinkowej

W obliczeniach przykładowego wskaźnika elastyczności otrzymaliśmy wartość o szesnastocyfrowym rozwinięciu dziesiętnym. Dla zobrazowania danych dla celów podejmowania decyzji na ich podstawie

lepszą byłaby reprezentacja procentowa z dokładnością do pierwszej cyfry po przecinku. To bowiem, co widzimy w wyniku, to jedynie reprezentacja wartości przechowywanej w komórce pamięci zastosowanej maszyny, uwzględniająca błędy binarno-dziesiętnej konwersji reprezentacji.

Chcąc zaokrąglić wynik do wartości procentowej napotykamy w dokumentacji komunikat ostrzegający o powszechnej nieprzewidywalności zaokrąglania wartości zmiennoprzecinkowych na maszynach operujących na danych zapisanych binarnie, przedstawiony na Rys. 5.

Uwaga na zaokrąglanie binarnie przechowywanych liczb zmiennoprzecinkowych.

Ten fakt może wymagać od nas zastosowania mechanizmu operacji na liczbach, który będzie działał stale zgodnie z naszymi oczekiwaniami.

Wyobraźmy sobie sytuację, w której dwóch użytkowników złożyło konkurencyjne oferty na sprzedaż znaczących wolumetrycznie produktów. Ocena jednego z produktów jest prawdziwym zaokrągleniem wyniku, a ocena drugiego wynika z błędnego zachowania opisanego w przytoczonym fragmencie dokumentacji. Możemy wybrać tylko jeden produkt. Który powinniśmy wybrać?

Motywuje to do rozważenia zastosowania w obliczeniach arytmetyki, której zachowanie będzie zachowywało spójność i przewidywalność wyników działań. Jakie narzędzia mogą posłużyć do tego? Wiemy, że takie metody stosowane są w instytucjach finansowych.

Dlaczego jest to ważne? Obejrzyjmy trzy przykłady:

```
1 >>> 0.1 + 0.2
2 0.30000000000000004
```

```
1 >>> round(2.675, 2)
2 2.67
```

```
1 sum = 0.0
2 for i in range(10):
3     sum += 0.1
4
5 sum
```

```
1 Out[50]: 0.9999999999999999
```

Implementacja programu stosuje typ danych `Decimal`, który pozwala ograniczyć wpływ opisanych trudności na działanie systemu FMA. Typ danych `Decimal` oferuje następujące korzyści: możemy w nim z góry dowolnie zdefiniować precyzję dla danego zadania; możemy także zainstalować pułapki wychwytyjące operacje zaokrąglania i zadbać o blokowanie operacji grożących utratą precyzji.

Na przykład, w dziesiętnym zapisie zmiennoprzecinkowym typu `Decimal` następujące równanie jest prawdziwe w odróżnieniu od zwykłego typu `double precision float`:

$$0.1 + 0.1 + 0.1 - 0.3 = 0$$

Większa dbałość o precyzję zapisu sprawia, że typ `Decimal` jest preferowanym typem do zastosowań finansowych gdzie pośród wymagań spotykamy ściśle niezmienniki co do równań.

Rekomendacje dotyczące typu danych `Decimal`

Pomimo zalet typ `Decimal` niesie ze sobą istotne wymagania i ograniczenia.

W typie `Decimal` to wciąż człowiek (programista) musi zdecydować, jaka precyzja będzie wymagana dla danej operacji. Dlaczego tak jest? Spójrzmy na przykłady podane w książce Knutha *???*, które ilustrują załamanie się własności: asocjatywności oraz dystrybutywności dodawania wobec niedostatecznej precyzji obliczeń:

```

1 >>> from decimal import Decimal, getcontext
2 >>> getcontext().prec = 8
3
4 >>> u, v, w = Decimal(11111113), Decimal(-11111111), Decimal('7.51111111')
5 >>> (u + v) + w
6 Decimal('9.5111111')
7 >>> u + (v + w)
8 Decimal('10')
```

Po uwzględnieniu ograniczeń typu danych `Decimal`, które wymagają za każdym razem ręcznego doboru precyzji obliczeń okazują się w praktyce bliskie ograniczeniom standardowej arytmetyki liczb zmiennoprzecinkowych, w prototypowej implementacji podmodułu certyfikacji do przechowywania liczb zmiennoprzecinkowych zastosowano standardowy typ `float`.

Rozdział 6

Wyniki testów

Celem implementacji modułu FlexClassifier było zaoferowanie odbiorcy funkcjonalności oszacowania własnej elastyczności. Obraną metodą realizacji było wykazanie najbliższego profilu elastyczności wypełnianego przez odbiorcę spośród profili interesujących Agregatora.

Funkcjonalność prototypu

Celem testów funkcjonalnych była weryfikacja możliwości oferowanych przez opracowany prototyp modułu programowego FlexClassifier w kontekście operacji na profilach zużycia i profilach elastyczności pod kątem analizy danych na urządzeniu FlexMeter.

Implementacja prototypu zakładała zapewnienie funkcjonalności szacowania elastyczności przez moduł FlexClassifier, oraz możliwości zdalnego dostępu do tej funkcjonalności.

Pierwszą z funkcjonalności zrealizowano przy użyciu języka programowania Python. Drugą zapewniono poprzez wykorzystanie zdalnego dostępu konsolowego przez protokół SSH oraz sieciowego interfejsu API środowiska Jupyter.

Cztery notatniki z badań zawierające szczegółową prezentację funkcjonalności zaimplementowanego modułu FlexClassifier, wraz z samym modulem, znajdują się na dołączonej do tej pracy płycie CD.

Metodę szacowania elastyczności dopasowano pod kątem czytelności wyników, zgodnie z zaprojektowanymi cechami architektury. Osiągnięto to przyjmując całkowitoliczbowy typ wektorów wejściowych oraz wykorzystując miarę elastyczności opartą na różnicach wektorów.

W wymaganiach funkcjonalnych dla opracowanego prototypu przewidziano następujące przypadki użycia:

1. Graficzne porównanie dwóch profili zużycia,
2. Obliczenie funkcji różnicy pomiędzy dwoma profilami zużycia,
3. Graficzna inspekcja funkcji różnicy między profilami,
4. Graficzne zestawienie funkcji różnicy profilu z profilem wzorcowym,
5. Uzyskanie procentowej różnicy ilości zużytej energii w oparciu o dwa profile,
6. Obliczenie różnicy pomiędzy uzyskaną a oczekiwaną modyfikacją profilu zużycia, i w końcu
7. Wyznaczenie odległości podanego profilu elastyczności od dwóch profili wzorcowych według miary odległości wyznaczanej przez całkowitą ilość energii w danym profilu,
8. Wskazanie profilu bliższego do podanego spośród dwóch profili testowych, przy wykorzystaniu miary odległości z punktu 7.

Dla celów pierwszego testu wygenerowano dwa przykładowe profile zużycia. Ich zestawienie graficzne wygenerowane przy pomocy testowanego modułu ilustruje Rys. 6.

Zestawienie dwóch profili zużycia: referencyjnego oraz próbnego.

Drugi test zakładał obliczenie funkcji różnicy pomiędzy dwoma profilami zużycia. Trzeci obejmował graficzne zestawienie funkcji różnicy z porównywanymi przebiegami.

Wygenerowaną funkcję różnicy wraz z porównanymi przebiegami wypełniającą założenia przypadków użycia 2. i 3. zademonstrowano na Rys. 6.

Obliczona funkcja różnicy profili zużycia.

Czwarty przypadek użycia związany był z graficznym zestawieniem profilu wzorcowego z różnicą profili. Rezultat przedstawiono na Rys. 6.

Zestawienie profilu wzorcowego z różnicą zużycia.

W piątym z przypadków użycia obliczono całkowite zmniejszenie zużycia uzyskując wynik o oczekiwanej postaci i wartości. Test uznano w ten sposób za zaliczony.

Przypadek szósty zakładał porównanie profili elastyczności.

Rezultat testu przedstawiający zestawienie dwóch profili elastyczności - oczekiwanego i rzeczywistego zawarty został na Rys. 6.

Porównanie profili zużycia.

Zawartość danych wyjściowych była zgodna z oczekiwaną, wobec czego test uznano za wypełniony.

Siódmy przypadek testowy związany był z wyznaczeniem odległości podanego profilu elastyczności od dwóch profili porównawczych. Zadana miarą odległości była całkowita energia związana z profilem.

Odległość profili zużycia

W ramach przypadku testowego otrzymano odległości profilu testowanego oznaczonego kolorem niebieskim od profili wzorcowych odpowiadających kolejno: obciążeniu szczytu oraz strategicznemu oszczędzaniu wynoszącą odpowiednio 2 oraz -70.

Rezultaty te nawiązują do sytuacji gdy wykazana przez odbiorcę elastyczność przekroczyła próg profilu obciążenia szczytu, lecz nie sięgnęła ilości pozwalającej na zakwalifikowanie odpowiedzi do kategorii strategicznego oszczędzania energii.

Stwierdzono zgodność wyniku z wizualną interpretacją danych, uznając przypadek testowy siódmy uznano za zaliczony.

Przypadek ósmy zakładał programowe wskazanie profilu elastyczności najbliższego podanemu z dwóch dostarczonych profili wzorcowych.

Wykorzystano zestaw profili elastyczności tożsamy z tym w przypadku siódmym. Otrzymano wynik wskazujący na pierwszy z profili porównawczych. Profil ten najmniej różni się od profilu testowego. Co więcej, dodatnia różnica między nim a profilem testowym oznacza, że profil testowy sygnalizował większą zrealizowaną oszczędność niż wymagana przez profil porównawczy. Był to wynik zgodny z oczekiwaniami, wobec czego test uznano za zaliczony.

Podsumowanie testów funkcjonalności

Założeniem modułu FlexClassifier było zaoferowanie odbiorcy możliwości wyznaczenia najbliższego profilu elastyczności przez niego wypełnianego spośród profili interesujących Agregatora.

Celem wykonanych testów była weryfikacja możliwości oferowanych przez opracowany prototyp moduł programowego FlexClassifier w kontekście operacji na profilach zużycia i profilach elastyczności pod kątem analizy danych na urządzeniu FlexMeter bądź na klastrze obliczeniowym.

Wszystkie przypadki testowe zostały pomyślnie wypełnione, co potwierdziło gotowość modułu do realizacji automatycznych obliczeń z użyciem profili zużycia oraz profili elastyczności z umożliwieniem użytkownikowi nieskomplikowanej inspekcji danych oraz wyników ich przetwarzania.

Wydajność prototypu

Celem testów wydajności prototypu było eksperymentalne zweryfikowanie zaproponowanej architektury pod kątem wydajności w przewidzianych zadaniach.

W testach założono maksymalną roczną liczbę oszacowań elastyczności przez pojedynczego odbiorcę wynoszącą 400.

Wydajność prototypu sprawdzono na trzech platformach obliczeniowych. W ramach pojedynczego testu wydajności uruchomiono 100 kompletnych przebiegów testowych mających za zadanie odwzorowanie charakteru najważniejszych obliczeń wykonywanych w ramach szacowania elastyczności w okresie kwartału.

Zestawienie czasu wykonania pojedynczego testu wydajności na poszczególnych platformach ujęte zostały w tabeli 6.1.

Tabela 6.1: Wydajność prototypu na poszczególnych platformach.

Jednostka obliczeniowa	ARM v71 (4 rdzenie)	Intel i7 CPU (8 rdzeni)	Intel i9 CPU (32 rdzenie)
Czas przetwarzania szeregowego	7 min. 32 sek.	1 min. 39 sek.	1 min. 47 sek.
Czas przetwarzania równoległego	5,549 sek.	1,188 sek.	1,210 sek.

Platformą testową dla procesora ARM v71 było urządzenie NanoPi NEO zawierające 512 MB pamięci RAM w technologii DDR3. Maksymalne znamionowe taktowanie rdzeni procesora wynosiło 1,2 GHz. Urządzenie wykonywało obliczenia lokalnie i było obsługiwano poprzez protokół SSH z odległości wynoszącej 7,5 km obliczonej wzdłuż głównych dróg.

Platformą dla CPU Intel i7 był komputer osobisty obsługiwany lokalnie. Procesor oznaczony był numerem modelu i7-8550U, a przedział znamionowego taktowania jego rdzeni wynosiło od 1,8 GHz w trakcie normalnej pracy do 4.0 GHz przy sterowaniu automatycznym z użyciem technologii Turbo Boost.

Platforma procesora Intel i9 obsługiwana była zdalnie z odległości równej tej w przypadku pierwszego urządzenia w architekturze ARM. Do sterowania urządzeniem wykorzystano protokół SSH. Same obliczenia prowadzone były lokalnie na urządzeniu.

Testy wykorzystujących zrównoleglenie obliczeń zostały wykonane przy pomocy narzędzia GNU Parallel.

W celu zniwelowania wpływu na wyniki testów kosztownych czasowo operacji związanych z drukowaniem komunikatów na ekranie konsoli, wszystkie testy uruchomiano przy ograniczeniu informacji raportowanych do konsoli do niezbędnego minimum.

Dla ograniczenia wpływu opóźnień zdalnego dostępu do stacji przetwarzania danych obliczenia uruchamiane były w emulowanej konsoli lokalnej dostępnej z użyciem narzędzia *Screen*.

Podsumowanie

Przeprowadzone testy wydajności wykazały, że zarówno na platformach stacjonarnych jak i na proponowanej platformie wbudowanej spodziewany maksymalny kwartalny nakład obliczeń wykonywany jest w czasie poniżej 8 minut. Oznacza to, że każda z przetestowanych platform pozwala na płynną obsługę użytkownika w zakresie szacowania elastyczności odbioru energii.

Rozdział 7

Podsumowanie

Celem tej pracy było przedstawienie architektury umożliwiającej zautomatyzowaną ocenę elastyczności odbiorcy energii elektrycznej w kontekście udziału w programie DSR.

Cel pracy rozpatrywany był w dwóch aspektach: w aspekcie identyfikacji potrzeb związanych z szacowaniem elastyczności odbiorców energii elektrycznej, oraz z z perspektywy zaprojektowania architektury systemu wspierającego odbiorcę w podejmowaniu decyzji związanych z przystępowaniem do programu DSR umożliwionym odbiorcy przez agregatora. Te cele ogólne osiągnięto poprzez:

1. Zidentyfikowanie potrzeb, w tym:

- Rozpoznanie docelowego środowiska dla systemu poprzez omówienie znaczenia programów typu DSR w kontekście bezpieczeństwa i stabilności sieci elektroenergetycznej (rozdział 1), oraz
- Wypunktowanie kluczowych potrzeb i korzyści wynikających z zastosowania procesu szacowania elastyczności odbiorców w zarządzaniu pulą aktywnych odbiorców z perspektywy operatora oraz odbiorcy (rozdział 3),

2. Zaprojektowanie architektury systemu, w tym:

- Zaproponowanie architektury (rozdział 3) dla systemu odpowiadającego na zidentyfikowane potrzeby w kontekście docelowego środowiska, oraz
- Weryfikację możliwości implementacji podstawowych elementów systemu na docelowej platformie poprzez wykonanie i przetestowanie (rozdział 6 oraz zawartość dołączonej płyty) prototypowej implementacji na zaproponowanej docelowej platformie sprzętowej.

Kluczowymi rezultatami pracy osiągniętymi w odniesieniu do celów szczegółowych w zakresie projektowania architektury dla systemu automatycznego szacowania elastyczności odbiorców było:

- Identyfikacja i wybór parametrów pozwalających jak najlepiej określić i mierzyć elastyczność odbiorcy energii (rozdział 3),
- Określenie charakterystycznych wzorcowych profili zmian zużycia energii i opisanie ich w odniesieniu do zdefiniowanej miary elastyczności (rozdział 3),
- Opracowanie koncepcji szacowania elastyczności poprzez odniesienie do wzorcowych profili elastyczności (rozdział 3),
- Zaprojektowanie architektury systemu umożliwiającej realizację opracowanej koncepcji (rozdział 4),
- Wykonanie i przedstawienie wzorcowej, prototypowej implementacji kluczowego modułu oprogramowania systemu oraz opisanie jego funkcjonowania (rozdział 5), oraz

- Dostarczenie odbiorcy przy pomocy systemu informacji istotnych dla zawarcia obopólnie korzystnej umowy współpracy z agregatorem (rozdział 5).

Przedstawiona w tej pracy architektura spełnia postawione przez nią zadanie, implementując ocenę elastyczności ze szczególnym uwzględnieniem bezpieczeństwa wrażliwych danych odbiorcy, wygody dostępu do wyników pomiaru oraz łatwości dalszego rozwoju systemu.

Charakterystyka architektury FMA sprawia, że odbiorca może w bardziej zautomatyzowany sposób zrównoważyć koszty odbieranej energii zyskami z tytułu modyfikowania profilu swego zapotrzebowania [44].

Przy użyciu opracowanej w niniejszej pracy koncepcji szacowania elastyczności poprzez odniesienie do wzorcowych profili elastyczności, szacowanie może być realizowane zarówno indywidualnie, jak też w sposób automatyczny może być realizowane przez agregatora w stosunku do wszystkich podmiotów, które reprezentuje.

Proponowane kierunki dalszego rozwoju

Projektanci budynków i systemów ich obsługi zwracają szczególną uwagę na ich przystosowanie do dołączenia ich do systemów inteligentnego zarządzania poborem energii. Wskazuje na to praca magisterska w dziedzinie Architektury przedstawiona na Instytucie Technologicznym Massachusetts (MIT) ????. Ten sam autor kontynuował prace nad rozwojem metod integracji budynków z systemami DSR, publikując w trzy lata później pracę doktorską w tej właśnie dziedzinie ???.

Jedną z perspektyw zastosowania zaprojektowanej architektury jest włączenie jej do jednego z systemów projektowanych do automatycznej obsługi usług DSR w jednym z nowo projektowanych budynków.

Inną drogą jest istniejącego systemu związanego z automatycznym sterowaniem odbiorem takiego jak OpenADR, albo wykorzystanie opracowanego modułu w roli rozszerzenia wspomnianego wcześniej systemu OneMeter.

Możliwe jest także wykorzystanie modułu szacowania elastyczności jako komponentu w budowie rozproszonego systemu świadczącego usługi DSR ???.

Wnioski

Wyniki przeprowadzonych badań pokazały, że przedstawiona architektura spełnia postawione początkowo zadanie. Ze stanu takiego wynika, iż architektura FMA oferuje realną perspektywę ograniczenia kosztów świadczenia usługi DSR przez jej operatora, a odbiorcy przynosi komfort związany z automatyzacją procesu jego przyjęcia.

Podsumowując zatem, zaprojektowanie architektury odpowiadającej postawionym kryteriom zwerifikowano jako możliwe, a przygotowany prototyp potwierdził działanie kluczowych elementów architektury w zamierzony sposób.

Literatura

- [1] M. Leks, “Parę słów o architekturze korporacyjnej | ArchiReq | Architektura korporacyjna w praktyce,” 23-May-2010. [Online]. Available: <http://archireq.pl/pl/pare-slow-o-architekturze-korporacyjnej/>. [Accessed: 14-Jan-2019].
- [2] C. Finck, R. Li, R. Kramer, and W. Zeiler, “Quantifying demand flexibility of power-to-heat and thermal energy storage in the control of building heating systems,” *Applied Energy*, vol. 209, pp. 409–425, Jan. 2018.
- [3] R. G. Junker *et al.*, “Characterizing the energy flexibility of buildings and districts,” *Applied Energy*, vol. 225, pp. 175–182, Sep. 2018.
- [4] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education, 1994.
- [5] W. Dołęga, “National grid electrical power infrastructure – threats and challenges,” *Polityka Energetyczna*, vols. T. 21, z. 2, 2018.
- [6] PSE, “Komunikat dotyczący wprowadzenia stopni zasilania z dnia 10 sierpnia 2015 r. Z godz. 7:55 - Komunikaty OSP - PSE,” 10-Aug-2015. [Online]. Available: <https://www.pse.pl/-/komunikat-dotyczacy-wprowadzenia-stopni-zasilania-z-dnia-10-sierpnia-2015-r-z-godz-7-55>. [Accessed: 24-Dec-2018].
- [7] J. Kwarciński, “Komunikat dotyczący wprowadzenia stopni zasilania z dnia 10 sierpnia 2015 r.” BIP, 10-Aug-2015.
- [8] B. Oksińska, “URE nie odpuści firmom za 20. stopień zasilania,” *energianews*, 20-Jun-2017. [Online]. Available: <https://energia.rp.pl/energetyka-zawodowa/elektroenergetyka/5200-ure-nie-odpusci-firmom-za-20-stopien-zasilania>. [Accessed: 23-Dec-2018].
- [9] A. Wieczerzak-Krusińska, “Ograniczenia prądu: 600 firm ukaranych za brak redukcji podczas 20 stopnia zasilania,” 03-Aug-2017. [Online]. Available: <https://www.parkiet.com/Energetyka/308039932-Ograniczenia-pradu-600-firm-ukaranych-za-brak-redukcji-podczas-20-stopnia-zasilania.html>. [Accessed: 23-Dec-2018].
- [10] A. Wieczerzak-Krusińska, “Upały przywołują widmo stopni zasilania,” *energianews*, 31-Jul-2018. [Online]. Available: <https://energia.rp.pl/energetyka-zawodowa/elektroenergetyka/13501-upaly-przywoluja-widmo-stopni-zasilania>. [Accessed: 24-Dec-2018].
- [11] A. Wieczerzak-Krusińska, “PSE: Kolejny dzień, kolejny rekord poboru mocy,” *energianews*, 02-Aug-2018. [Online]. Available: <https://energia.rp.pl/energetyka-zawodowa/13526-pse-kolejny-dzien-kolejny-rekord-poboru-mocy>. [Accessed: 24-Dec-2018].

- [12] PAP, “PSE: wprowadzenie stopni zasilania w sierpniu zapobiegło blackoutowi,” *energianews*, 03-Nov-2015. [Online]. Available: <https://energia.rp.pl/energetyka-zawodowa/elektroenergetyka/10388-pse-wprowadzenie-stopni-zasilania-w-sierpniu-zapobieglo-blackoutowi>. [Accessed: 24-Dec-2018].
- [13] M. Klos, M. Jakubek, and M. Krupa, “On a Market Design of Emergency DSR in Poland - Valuation and Optimal Acquisition of Services,” in *2018 15th International Conference on the European Energy Market (EEM)*, 2018, pp. 1–5.
- [14] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings, “Putting the ‘smarts’ into the smart grid: A grand challenge for artificial intelligence,” *Communications of the ACM*, vol. 55, no. 4, pp. 86–97, Jan. 2012.
- [15] P. D. Lund, J. Lindgren, J. Mikkola, and J. Salpakari, “Review of energy system flexibility measures to enable high levels of variable renewable electricity,” *Renewable and Sustainable Energy Reviews*, vol. 45, pp. 785–807, May 2015.
- [16] H. Mangesius, S. Hirche, M. Huber, and T. Hamacher, “A framework to quantify technical flexibility in power systems based on reliability certificates,” in *IEEE PES ISGT Europe 2013*, 2013, pp. 1–5.
- [17] A. Wang, R. Li, and S. You, “Development of a data driven approach to explore the energy flexibility potential of building clusters,” *Applied Energy*, vol. 232, pp. 89–100, Dec. 2018.
- [18] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, “Private memoirs of a smart meter,” in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building - BuildSys '10*, 2010, p. 61.
- [19] D. Alahakoon and X. Yu, “Smart Electricity Meter Data Intelligence for Future Energy Systems: A Survey,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 1, pp. 425–436, Feb. 2016.
- [20] P. Carroll, T. Murphy, M. Hanley, D. Dempsey, and J. Dunne, “Household Classification Using Smart Meter Data,” *Journal of Official Statistics*, vol. 34, no. 1, pp. 1–25, Mar. 2018.
- [21] Q. Wang and B. Hodge, “Enhancing Power System Operational Flexibility With Flexible Ramping Products: A Review,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1652–1664, Aug. 2017.
- [22] J. Socha, “DSR Program Bieżący Uproszczony,” *Polskie Sieci Elektroenergetyczne S.A.*, p. 26, Jul. 2018.
- [23] J. N. Oliveira, “Formal Methods: The CAMILA Perspective,” 21-Mar-1997. [Online]. Available: <http://www4.di.uminho.pt/~jno/html/cam-tcp.html>. [Accessed: 14-Jan-2019].
- [24] A. Josey, M. Lankhorst, I. Band, H. Jonkers, and D. Quartel, “An Introduction to the ArchiMate® 3.0 Specification,” p. 20, Jun. 2016.
- [25] Visual Paradigm, “ArchiMate Diagram Tutorial.” [Online]. Available: <https://online.visual-paradigm.com/tutorials/archimate-tutorial/#full-togaf-adm-in-archimate-3>. [Accessed: 18-Jan-2019].
- [26] The Open Group, “ArchiMate® 3.0.1 Specification,” *The Open Group*, 2017. [Online]. Available: http://pubs.opengroup.org/architecture/archimate3-doc/chap01.html#_Toc489945950. [Accessed: 14-Jan-2019].
- [27] R. Owens, “Legion: Distributed, stateful, homogeneous microservice framework.” 2017. [Online]. Available: <http://hackage.haskell.org/package/legion>. [Accessed: 17-Jan-2019].
- [28] *Take the World from Another Point of View*. 1973.

- [29] B. Milewski, *Category Theory for Programmers*. University of Texas, 2018.
- [30] E. S. Raymond, *The Art of UNIX Programming*. Addison-Wesley Professional, 2003.
- [31] A. Church, “A formulation of the simple theory of types,” *The Journal of Symbolic Logic*, vol. 5, no. 2, pp. 56–68, Jun. 1940.
- [32] P. Andrews, “Church’s Type Theory,” in *The Stanford Encyclopedia of Philosophy*, Summer 2018., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2018.
- [33] P. Hudak *et al.*, “Report on the Programming Language Haskell: A Non-strict, Purely Functional Language Version 1.2,” *SIGPLAN Not.*, vol. 27, no. 5, pp. 1–164, May 1992.
- [34] L. Courtès, “Functional Package Management with Guix,” May 2013.
- [35] GNU, “Download — GuixSD,” 2019. [Online]. Available: <https://www.gnu.org/software/guix/download/>. [Accessed: 15-Jan-2019].
- [36] Sterlind, “Guix: An advanced operating system | Hacker News,” 2019. [Online]. Available: <https://news.ycombinator.com/item?id=18902823>. [Accessed: 15-Jan-2019].
- [37] P. Neidhardt, “Guix: A most advanced operating system,” 2019. [Online]. Available: <https://ambrevar.xyz/guix-advance/index.html>. [Accessed: 15-Jan-2019].
- [38] E. Dolstra and A. Löb, “NixOS: A Purely Functional Linux Distribution,” in *Proceedings of the 13th ACM SIGPLAN International Conference on Functional Programming*, 2008, pp. 367–378.
- [39] P. Prins, “Creating a reproducible workflow with CWL,” *GitLab*, 2019. [Online]. Available: <https://gitlab.com/pjotrp/guix-notes/blob/master/WORKFLOW.org>. [Accessed: 15-Jan-2019].
- [40] P. Wadler, “The essence of functional programming,” in *Proceedings of the 19th ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL ’92*, 1992, pp. 1–14.
- [41] Wikibooks, “Haskell - Wikibooks, biblioteka wolnych podręczników,” 2017. [Online]. Available: [https://pl.wikibooks.org/wiki/Haskell#Haskell_vs_OOP_\(Programowanie_obiektowe\)](https://pl.wikibooks.org/wiki/Haskell#Haskell_vs_OOP_(Programowanie_obiektowe)). [Accessed: 16-Jan-2019].
- [42] J. Backfield, *Becoming Functional: Steps for Transforming Into a Functional Programmer*. ”O’Reilly Media, Inc.”, 2014.
- [43] L. Hernandez *et al.*, “A multi-agent system architecture for smart grid management and forecasting of energy demand in virtual power plants,” *IEEE Communications Magazine*, vol. 51, no. 1, pp. 106–113, Jan. 2013.
- [44] C. Edmunds, S. Galloway, and S. Gill, “Distributed electricity markets and distribution locational marginal prices: A review,” in *2017 52nd International Universities Power Engineering Conference (UPEC)*, 2017, pp. 1–6.