

## Autoreferat

### 1. Imię i nazwisko

Wiktor Bohdan Daszczuk

### 2. Posiadane dyplomy, stopnie naukowe

- 2003 stopień doktora nauk technicznych w dyscyplinie: Informatyka;  
miejsce: Wydział Elektrotechniki i Technik Informatycznych, Politechnika Warszawska  
tytuł rozprawy doktorskiej:  
**„Verification of temporal properties in concurrent systems”**
- 1982 tytuł magistra inżyniera informatyki, specjalność: budowa i oprogramowanie komputerów  
miejsce: Wydział Elektrotechniki i Technik Informatycznych, Politechnika Warszawska  
tytuł pracy dyplomowej: „System biblioteczny dla języka Pascal 1900”,  
praca dyplomowa została wyróżniona

#### 2.1. Nagrody i wyróżnienia

- 1981 Praca dyplomowa magisterska “System biblioteczny dla języka Pascal 1900”, została wyróżniona.
- 1990 Medal Mikrolaur na targach Komputer '90 za zabezpieczenia przeciwwawaryjne i rekonstrukcję poawaryjną w systemie obsługi magazynów wysokiego składowania STER
- 1990 Nagroda Ministra Rynku Wewnętrznego za opracowanie i wdrożenie systemu obsługi magazynów wysokiego składowania STER
- 1996 Nagroda Siemens za pracę badawczo-wdrożeniową (metodologia monitorowania obiektów przemysłowych poprzez sieć zarządzania i opracowanie takiego systemu dla bloków energetycznych)
- 1997 Nagroda zespołowa JM Rektora PW za prace badawcze nad automatami CSM służącymi do modelowania i weryfikacji systemów współbieżnych
- 2014 Nagroda zespołowa JM Rektora PW za prace badawcze w projekcie Eco-Mobilność – wykonanie algorytmów zarządzania flotą autonomicznego transportu miejskiego i symulatora takich sieci
- 2016 Nagroda zespołowa II stopnia Prezesa Rady Ministrów RP za prace w projekcie Eco-Mobilność: transport przyjazny środowisku - wykonanie algorytmów zarządzania flotą autonomicznego transportu miejskiego i symulatora, badanie przepustowości sieci transportu
- 2018 Medal Komisji Edukacji Narodowej

### 3. Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych

|                   |   |
|-------------------|---|
| 10.1990 – 09.1998 | asystent, Instytut Informatyki Politechniki Warszawskiej            |
| 10.1998 – 09.2009 | starszy wykładowca, Instytut Informatyki Politechniki Warszawskiej  |
| 10.2002 – 03.2003 | starszy specjalista, Instytut Informatyki Politechniki Warszawskiej |
| 04.2003 – obecnie | adiunkt, Instytut Informatyki Politechniki Warszawskiej             |

#### 3.1. Życiorys

Urodziłem się 13 kwietnia 1958 w Olsztynie. Ukończyłem Szkołę Podstawową nr 14 w Olsztynie i Liceum Ogólnokształcące nr 2 im. K.I. Gałczyńskiego w Olsztynie.

W roku 1977 rozpocząłem studia na Wydziale Elektroniki Politechniki Warszawskiej. Studia ukończyłem w roku **1982**, otrzymując tytuł **magistra inżyniera informatyka**. W czasie studiów prowadziłem w ramach Warszawskiego Centrum Studenckiego Ruchu Naukowego prace związane z wizualizacją procesu produkcji w przedsiębiorstwie Bumar.

W latach **1982-1990** pracowałem w Przedsiębiorstwie Systemów Komputerowych MERA-SYSTEM. Moje projekty były związane z systemem operacyjnym SOM-5 dla minikomputerów MERA-400 (głównie przy jądrze systemu), konstrukcji sieci pierścieniowej dla zespołu kas, oraz budowie sprzętu i oprogramowania systemowego dla komputera MS-86 w architekturze rozproszonej. W ramach tych prac współpracowałem z drem Markiem Gondzio z Instytutu Informatyki Politechniki Warszawskiej nad debuggerem do uruchamiania sprzętu i systemu operacyjnego komputera MS-86, co zaowocowało pracą dyplomową studenta IIPW. Jednocześnie prowadziłem szereg innych działalności, w tym nad oprogramowałem interferometru cyfrowego na Wydziale Mechaniki Precyzyjnej Politechniki Warszawskiej (dziś Wydział Mechatroniki). Okazjonalnie prowadziłem wykład z modelowania mechanizmów systemów operacyjnych na seminarium dyplomowym studentów informatyki Uniwersytetu Warszawskiego.

Od roku **1990** do dziś pracuję w Instytucie Informatyki Politechniki Warszawskiej, początkowo na stanowisku asystenta, później starszego wykładowcy, a od roku **2003** po uzyskaniu stopnia **doktora nauk technicznych** na stanowisku adiunkta. Przez cały czas zatrudnienia na Politechnice Warszawskiej zajmuję się modelowaniem, specyfikacją, weryfikacją i symulacją systemów rozproszonych. Drugim, ząębającym się z podstawowym tematem, istotnym obszarem badań jest modelowanie autonomicznego transportu miejskiego.

Przed i w czasie pracy w IIPW, jednocześnie prowadziłem własne przedsiębiorstwa (w latach 1986-2013), w którym kierowałem wykonaniem systemu automatyzującego gospodarkę magazynową, w szczególności w magazynach wysokiego składowania. Systemy te zostały wdrożone w ok. 30 przedsiębiorstwach. Szczególną cechą wykonanych systemów była automatyczna rekonstrukcja poawaryjna, co było szczególnie istotne w obliczu bardzo

awaryjnego w tym okresie sprzętu komputerowego, szczególnie sieciowego. Za prace te otrzymałem w roku 1990 Medal Mikrolaur na targach Komputer '90.

W latach 2011-2012 ukończyłem studia podyplomowe na Wydziale Prawa Uniwersytetu Warszawskiego w zakresie ochrony własności intelektualnej.

#### **4. Wskazanie osiągnięcia wynikającego z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. 2017 r. poz. 1789)**

##### **4.1. Tytuł osiągnięcia naukowego**

***Modelowanie, specyfikacja, weryfikacja i symulacja systemów rozproszonych***

##### **4.2. Cykl publikacji powiązanych tematycznie**

Przedkładany cykl publikacji powiązanych tematycznie zawiera 9 pozycji ([H1-H9], w kolejności publikacji) opublikowanych w latach 2015-2020. Prace te są opublikowane w czasopiśmie posiadających Impact Factor (JCR, pozycje [H1-H3]), w czasopiśmie indeksowanym jako Web of Science Emerging Sources [H4], oraz w recenzowanych materiałach konferencyjnych indeksowanych przez Web of Science (pozycje [H5-H7]). Ostatnia pozycja [H9] to monografia wydana w serii Studies in Computational Intelligence, indeksowanej w WoS. Spis publikacji wchodzących w skład cyklu zawiera informacje bibliograficzne, a także Impact Factor (IF) z roku wydania (w przypadku czasopism), liczbę punktów MNiSW, liczbę cytowań według Web of Science (WoS), według Scopus (Sc) oraz według Google Scholar (GS).

Suma IF, zgodnie z rokiem opublikowania, publikacji wchodzących w skład cyklu publikacji powiązanych tematycznie wynosi: **3,656** (ważony udziałem **2,551**).

Sumaryczny IF razem z publikacjami przed doktoratem: **5,308** (ważony udziałem **4,203**)

Sumaryczny indeks cytowań, według Web of Science wynosi **WoS: 31**, według Scopus **Sc: 53**, a według Google Scholar **GS: 270** (ostatni indeks wykazuje większą liczbę, ale niektóre cytowania są powielone).

[H1] **Daszchuk W B**, Choromański W, Mieścicki J, Grabski W (2015) Empty vehicles management as a method for reducing passenger waiting time in Personal Rapid Transit networks. *IET Intelligent Transport Systems*, 9(3), 231–239. Doi: 10.1049/iet-its.2013.0084, **JCR, IF: 0.843, MNiSW:25, WoS:15, Sc:16, GS:23**  
**udział habilitanta 55%**, polegał na opracowaniu i wykonaniu (wspólnie z innymi autorami) środowiska symulacyjnego sieci PRT (Personal Rapid Transit – szybki transport osobisty) oraz rozproszonych algorytmów zarządzania i koordynacji pojazdów PRT. Habilitant wykonał modele symulacyjne oraz przeprowadził eksperymenty

symulacyjne związane z zarządzaniem pustymi pojazdami w sieci PRT (wspólnie z innymi autorami). W publikacji opisano wieloparametryczny, rozproszony algorytm zarządzania pustymi pojazdami, wykazując że można uzyskać bardzo dobre wyniki rozmieszczenia pustych pojazdów w sieci, bez stosowania algorytmów predykcji. Tekst artykułu został opracowany w 90% przez habilitanta.

- udział Włodzimierza Choromańskiego 15%
- udział Jerzego Mieścickiego 15% (współautor nie żyje)
- udział Waldemara Grabskiego 15% (oświadczenie)

- [H2] **Daszczuk W B**, Mieścicki J, and Grabski W (2016) Distributed algorithm for empty vehicles management in personal rapid transit (PRT) network. *Journal of Advanced Transportation*, vol. 50, Issue 4, pp. 608-629. Doi: 10.1002/atr.1365, **JCR, IF: 1.813, MNiSW:30, WoS:6, Sc:7, GS:11**

**udział habilitanta 60%**, polegał na opracowaniu i wykonaniu (wspólnie z innymi autorami) środowiska symulacyjnego sieci PRT (Personal Rapid Transit) oraz rozproszonego algorytmu koordynacji pojazdów w sieci PRT (Personal Rapid Transit – szybki transport osobisty). Habilitant wykonał modele symulacyjne oraz przeprowadził eksperymenty symulacyjne związane z ewaluacją algorytmu (wspólnie z innymi autorami) i porównaniem z wydajnością innych, scentralizowanych algorytmów. Habilitant zbadał wpływ różnych parametrów na jakość działania algorytmu, w tym przede wszystkim wpływ horyzontu ograniczającego maksymalną odległość komunikujących się sterowników. Tekst artykułu został opracowany w 90% przez habilitanta.

- udział Jerzego Mieścickiego 20% (współautor nie żyje)
- udział Waldemara Grabskiego 20% (oświadczenie)

- [H3] **Daszczuk W B** (2017) Communication and Resource Deadlock Analysis using IMDS Formalism and Model Checking. *The Computer Journal*, vol.60, No.5, pp. 729-750, DOI: 10.1093/comjnl/bxw099, **JCR, IF: 1.000, MNiSW:25, WoS:3, Sc:4, GS:10**

Wkład habilitanta polegał na zdefiniowaniu poprzedniej wersji formalizmu IMDS (Integrated Model of Distributed Systems), opartej na zbiorach serwerów, agentów, wartości stanów i serwisów, oraz jego zastosowaniu wraz z weryfikacją modelową opartą na ogólnych (niezależnych od struktury modelu) formułach temporalnych do wykrywania zakleszczeń i rozproszonej terminacji procesów. Artykuł ten posłużył jako baza do opracowania rozdziałów 2 i 4 monografii [H9].

- [H4] **Daszczuk W B** (2018) Specification and Verification in Integrated Model of Distributed Systems (IMDS). *MDPI Computers*, vol.7, No.4, paper 65, pp. 1-26, DOI: 10.3390/computers7040065. **WoS Emerging Sources**

Wkład habilitanta polegał na zdefiniowaniu aktualnej wersji formalizmu IMDS (Integrated Model of Distributed Systems), opartej na stanach serwerów i komunikatach agentów, oraz jego zastosowaniu wraz z weryfikacją modelową opartą na ogólnych (niezależnych od struktury modelu) formułach temporalnych do wykrywania zakleszczeń i rozproszonej terminacji procesów. Artykuł ten posłużył jako baza do opracowania rozdziałów 1, 3 i 4 monografii [H9].

- [H5] **Daszczuk W B** (2018) Siphon-based deadlock detection in Integrated Model of Distributed Systems (IMDS). *3<sup>rd</sup> Workshop on Constraint Programming and Operation Research Applications (CPORA'18), Federated Conference on Computer Science and Information Systems, FedCSIS'18*, Poznań, Poland, 9-12 Sept 2018, pp.421-431, IEEE Xplore, DOI: 10.15439/2018F114, konferencja indeksowana w **Web of Science**, **MNiSW:15, WoS:1, GS:2**
- Wkład habilitanta polegał na rozszerzeniu opracowanej uprzednio metodologii znajdowania zakleszczeń w formalizmie IMDS, poprzez analizę syfonów odpowiadającej sieci Petriego. Rozszerzenie polega na analizie syfonów przy pomocy weryfikacji modelowej, co pozwala sklasyfikować syfony na te które odpowiadają i nie odpowiadają zakleszczeniom, oraz na identyfikację procesów uczestniczących w zakleszczeniu. Metoda pozwala na znajdowanie wielu zakleszczeń jednocześnie, w systemach o dowolnym kształcie: czysto cykliczne, terminujące i hybrydowe. Artykuł ten posłużył jako baza do opracowania rozdziału 7 monografii [H9].
- [H6] Czejdo B, **Daszczuk W B**, Grabski W, Bhattacharya S (2018) Cooperation of Multiple Autonomous Robots and Analysis of Their Swarm Behavior, *XXII konferencja TransComp*, 3-6.12.2018, Zakopane, *Autobusy-TEST* vol.19(12), pp. 872-879. DOI: 10.24136/atest.2018.516, **MNiSW:7**
- Wkład habilitanta 50%**, polegał na zastosowaniu automatycznej weryfikacji nieuchronności terminacji przy pomocy formalizmu IMDS i programu Dedan do wykazania poprawności zaplanowanych marszrut zbioru autonomicznych robotów. Artykuł przedstawia „szybkie prototypowanie” rozwiązań wraz z oceną ich poprawności przy pomocy formalizmu IMDS. Tekst artykułu został opracowany w 50% przez habilitanta.
- udział Bogdana Czejdo 30% (oświadczenie)
  - udział Waldemara Grabskiego 10% (oświadczenie)
  - udział Sambita Bhattacharya 10% (oświadczenie)
- [H7] **Daszczuk W B** (2019) Asynchronous Specification of Production Cell Benchmark in Integrated Model of Distributed Systems, *23<sup>rd</sup> International Symposium on Methodologies for Intelligent Systems*, Warsaw, Poland, 26-29 June 2017, pp. 115-129, *Studies in Big Data*, vol. 40, Springer, Cham, DOI: 10.1007/978-3-319-77604-0\_9, konferencja indeksowana w **Web of Science**, **MNiSW:15, GS:5**
- Wkład habilitanta polegał na zastosowaniu formalizmu IMDS (Integrated Model of Distributed Systems) do modelowania dużego systemu sterowania, znanego jako Karlsruhe Production Cell. Zweryfikowano kilka wersji systemu, wykorzystując asynchronizm formalizmu IMDS. Przeprowadzono badanie modelu z ograniczeniami czasu rzeczywistego, który jednak okazał się zbyt duży, nawet dla zewnętrznego weryfikatora z reprezentacją symboliczną przestrzeni osiągalności.
- [H8] **Daszczuk W B** (2019) Fairness in Temporal Verification of Distributed Systems, *13<sup>th</sup> International Conference on Dependability and Complex Systems DepCoS-RELCOMEX*, 2-6 July 2018, Brunów, Poland, pp. 135-150, *Advances in Intelligent Systems and Computing* vol.761, Springer, Cham, DOI: 10.1007/978-3-319-91446-6\_14, konferencja

indeksowana w **Web of Science**, **MNiSW:15**, **Sc:1**, **GS:2**

Wkład habilitanta polegał na opracowaniu i wdrożeniu algorytmu weryfikacji formuł temporalnych wykrywania zakleszczeń i weryfikacji rozproszonej terminacji. Algorytm wspiera silną sprawiedliwość (ang. compassion) w weryfikacji systemów rozproszonych.

[H9] **Daszczuk W B** (2020) Integrated Model of Distributed Systems, monografia w serii „Studies in Computational Intelligence” Springer, Cham, ISBN 978-3-030-12834-0, DOI: 10.1007/978-3-030-12835-7, **MNiSW** (według poprzedniego wykazu:**25**, według nowego wykazu:**80**)

Monografia stanowi podsumowanie prac habilitanta nad modelowaniem, specyfikacją, weryfikacją i symulacją systemów rozproszonych, w związku z czym w monografii powtórzono i rozwinięto opis i wnioski z wielu badań opisanych w innych pracach, głównie [H3-H8]. Są też opisane prace jeszcze nieopublikowane w oddzielnych artykułach, jak translacja IMDS do automatów czasowych i weryfikacja z ograniczeniami czasu rzeczywistego czy niewyczerpujący algorytm weryfikacji „2 wagabundów”.

## 5. Omówienie celu naukowego prac, osiągniętych wyników oraz ich ewentualnego wykorzystania

### 5.1. Motywacja i cele naukowe

Przedstawione prace dotyczą obliczeń rozproszonych, to znaczy takich, w których nie istnieje globalny stan, obliczenia prowadzone są w sposób równoległy, a postęp poszczególnych równoległych procesów dokonuje się na bazie niepełnej informacji o innych procesach. Mimo znacznych osiągnięć w projektowaniu i analizie systemów rozproszonych, w dziedzinie tej pozostaje wiele do zrobienia. Największą bolączką jest fakt, że większość metod opiera się na synchronicznej komunikacji, a akcje w systemie są podejmowane na podstawie stanu globalnego (lub nielokalnego, zakładając znajomość stanu co najmniej dwóch rozproszonych elementów). Jest to wynikiem przeniesienia metod odnoszących się do skupionych systemów współbieżnych do środowiska rozproszonego. Prace habilitanta zdążają do opracowania takiej metodologii projektowania i specyfikacji systemów rozproszonych, które uwzględnią ich naturalne cechy.

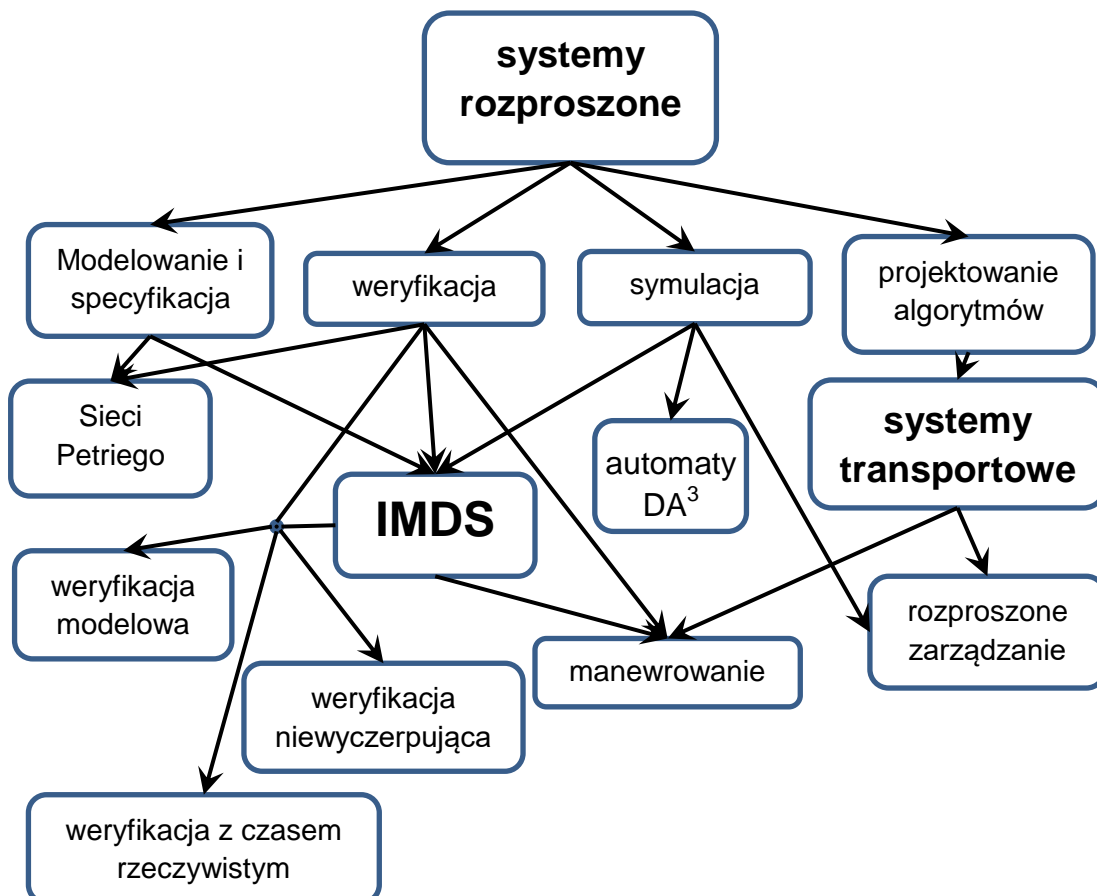
Przy projektowaniu, jak również eksploatacji systemów rozproszonych, istotna jest weryfikacja ich poprawności. Jedną z najistotniejszych cech jest dowodzenie wolności od zakleszczeń, czyli niepożądanego dyskontynuacji systemu. Jest wiele metod znajdowania zakleszczeń, jednak w większości dotyczą one zakleszczeń totalnych, to znaczy dotyczących wszystkich procesów w systemie. W systemach rozproszonych częste są zakleszczenia częściowe, dotyczące podzbioru procesów<sup>1</sup>. Nieliczne metody wykrywania zakleszczeń częściowych dotyczą systemów o bardzo ograniczonej strukturze, na przykład tylko cyklicznych. Ponadto, większość metod nie odróżnia zakleszczenia od rozproszonej terminacji, czyli celowego zakończenia pracy. Celem

---

<sup>1</sup> W niektórych pozycjach literaturowych zakleszczenia są nazywane globalnymi/lokalnymi, lecz habilitant wybrał określenia totalne/częściowe, gdyż lokalność odnosi się w systemach rozproszonych do nieznanego stanu globalnego.

habilitanta było stworzenie metod automatycznego znajdowania zakleszczeń totalnych i częściowych różnych typów, oraz totalnej i częściowej rozproszonej terminacji.

Wieloletnie prace habilitanta nad systemami rozproszonymi pozwoliły na zastosowanie metodologii ich projektowania w praktycznych przypadkach.



## 5.2. Zakres tematyczny badań i główne osiągnięcia

Zakres tematyczny badań, przedstawiony w postaci grafu na rysunku, obejmuje zagadnienia z obszaru modelowania, specyfikacji, weryfikacji i symulacji systemów rozproszonych, oraz projektowania rozproszonych algorytmów.

W tabeli przedstawiono zakres tematyczny poszczególnych publikacji wchodzących w skład osiągnięcia naukowego. Poszczególne zagadnienia badawcze i główne osiągnięcia naukowe można podsumować następująco:

### Modelowanie i specyfikacja

Najważniejszym wkładem badawczym jest sformułowanie formalizmu Zintegrowany Model Systemów Rozproszonych (Integrated Model of Distributed Systems – IMDS). Formalizm ten został opracowany w wyniku analizy różnych metod opisu systemów rozproszonych, które nie

uwzględniały naturalnych cech takich systemów: lokalność stanów, autonomia decyzji, asynchronizm akcji i komunikacji, dualizm komunikacyjny komunikatów/zasobów. Początkowa wersja IMDS została opisana w [H3], obecna wersja w [H4], oraz w podsumowującej monografii [H9].

|                                      | [H1] | [H2] | [H3] | [H4] | [H5] | [H6] | [H7] | [H8] | [H9] |
|--------------------------------------|------|------|------|------|------|------|------|------|------|
| modelowanie i specyfikacja           | +    | +    | +    | +    |      | +    | +    | +    | +    |
| IMDS                                 |      |      | +    | +    | +    | +    | +    | +    | +    |
| weryfikacja                          |      |      | +    | +    | +    | +    | +    | +    | +    |
| • weryfikacja modelowa               |      |      | +    | +    |      |      | +    | +    | +    |
| • weryfikacja z czasem rzeczywistym  |      |      |      |      |      |      | +    |      | +    |
| • weryfikacja niewyczerpująca        |      |      |      |      |      |      |      |      | +    |
| sieci Petriego                       |      |      |      |      | +    |      |      |      | +    |
| automaty rozproszone DA <sup>3</sup> |      |      |      |      |      |      |      |      | +    |
| systemy transportowe                 | +    | +    |      |      |      | +    |      |      | +    |
| symulacja                            | +    | +    |      |      |      |      |      |      | +    |
| projektowanie algorytmów             | +    | +    |      |      |      |      |      |      |      |
| manewrowanie                         |      |      |      |      |      | +    |      |      | +    |
| rozproszone zarządzanie              | +    | +    |      |      |      |      | +    |      |      |

Drugim osiągnięciem jest symulator sieci autonomicznego transportu miejskiego (opracowany w 50% przez habilitanta), w którym są badane oryginalne rozproszone algorytmy zarządzania flotą. Badania te opisano w pracach [H1], [H2], a samą konstrukcją symulatora w niewłaśczonej do cyklu publikacji pracy [Daszczuk, 2016].

### Zintegrowany Model Systemów Rozproszonych – IMDS

Formalizm IMDS, który jest oryginalnym wkładem habilitanta, pozwala na specyfikację systemów rozproszonych zachowując naturalne cechy systemów rozproszonych: lokalność serwerów (rozproszonych węzłów), autonomię ich akcji, asynchronizm wykonywania akcji i przesyłania komunikatów, oraz dualizm komunikacyjny [H3], [H8], [H9]. Formalizm IMDS jest unikatowy w modelowaniu połączenia wspomnianych cech systemów rozproszonych. Jednocześnie formalizm jest ramą dla kilku technik weryfikacji, opisanych indywidualnie. Z nielicznymi wyjątkami, prace nad modelem IMDS i weryfikacją modelową były i są prowadzone przez habilitanta samodzielnie.

### Weryfikacja

Formalizm IMDS pozwala na wyodrębnienie w systemie rozproszonym rezydentnych procesów serwerów i mobilnych procesów agentów. W kategoriach typów tych procesów można opisać określone cechy zachowania pojedynczych procesów w kontekście systemu rozproszonego. Pozwala to na znajdowanie częściowych zakleszczeń (w których uczestniczą podzbiór procesów) i częściowej terminacji. Znajdowanie częściowych zakleszczeń i terminacji opisano w [H3], [H4], [H5], [H6], [H7], [H8], [H9]. Opracowanie formuł temporalnych do wykrywania częściowych zakleszczeń i terminacji jest oryginalnym wkładem habilitanta, jak



również wykonanie weryfikatora Dedan znajdującego zakleszczenia częściowe i totalne, opisany w [H9]. Unikatowe cechy zaproponowanej metodologii pozwalają na automatyczną weryfikację, nie wymagającą od użytkownika znajomości logiki temporalnej ani samodzielnego specyfikowania częściowych zakleszczeń/terminacji w kategoriach cech weryfikowanego systemu.

### **Weryfikacja modelowa**

Zaprojektowano formuły temporalne w logikach czasu liniowego i czasu rozgałęzionego, które wyrażają najważniejsze (zdaniem habilitanta) cechy systemu rozproszonego: wolność od zakleszczeń i nieuchronność rozproszonej terminacji (jeżeli od procesów w danych systemie jest wymagana terminacja). Wkładem habilitanta jest opracowanie formuł niezależnych od struktury weryfikowanego systemu, pozwalających na automatyczną weryfikację. Formuły opisano w pracach [H3], [H4] i [H9], a własny algorytm ewaluacji formuł temporalnych w [H8] i [H9]. Powyższe cechy habilitant zaimplementował w środowisku Dedan, intensywnie wykorzystywanym przez studentów którzy weryfikują swoje rozwiązania problemów synchronizacyjnych.

### **Weryfikacja z czasem rzeczywistym**

Ograniczenia czasu rzeczywistego mogą powodować zniknięcie pewnych zakleszczeń, lub pojawienie się nowych. Habilitant opracował formalną translację IMDS do automatów czasowych TA [Baier and Katoen, 2008], oraz wyjście z programu Dedan do weryfikatora z czasem rzeczywistym Uppaal [H9]. Przedstawiono test funkcjonalny na przykładzie akademickim w [H9], oraz konkretny przykład weryfikacji, niestety zakończony niepowodzeniem z powodu zbyt dużej przestrzeni osiągalności [H7]. Pobocznym wynikiem jest formalna definicja automatów czasowych ze zmiennymi w [H9], które w literaturze są wprowadzane poza formalnym modelem.

Niepowodzenie weryfikacji z czasem rzeczywistym dużego systemu było impulsem do opracowania niewyczerpującego algorytmu weryfikacji z ograniczeniami czasowymi, który jest w trakcie implementacji.

### **Weryfikacja niewyczerpująca**

Bardzo duże systemy nie pozwalają na zbudowanie ich przestrzeni osiągalności, nawet przy zastosowaniu metod redukcji przestrzeni i weryfikacji symbolicznej [Baier and Katoen, 2008]. W takich przypadkach można zastosować niewyczerpujące przeszukiwanie przestrzeni. W pracy [H9] przedstawiono opracowany przez habilitanta innowacyjny algorytm „2 wagabundów”, pozwalający na znajdowanie częściowych zakleszczeń i sprawdzanie częściowej terminacji. Cecha ta jest unikatowa. Istotą algorytmu jest współpraca dwóch wątków, z których pierwszy stawia hipotezy co do zakleszczenia/terminacji, a drugi weryfikuje te hipotezy. Algorytm ten habilitant zaimplementował w środowisku Dedan.

## Sieci Petriego równoważne IMDS

W pracach [H5], [H9] przedstawiono translację IMDS do sieci Petriego (o ograniczonej strukturze). Analiza syfonów w sieci Petriego, połączona z weryfikacją modelową, pozwala na znajdowanie wielu zakleszczeń jednocześnie, w systemach o dowolnej strukturze (a nie tylko w czysto cyklicznych, jak przedstawiono w literaturze dotyczącej znajdowania zakleszczeń poprzez syfony). W systemach nie czysto cyklicznych syfony nie muszą oznaczać zakleszczenia, dlatego w literaturze opisano procedurę tylko dla systemów czysto cyklicznych. Połączenie analizy opróżnialności syfonów z weryfikacją modelową specyfikacji IMDS pozwala weryfikować systemy o dowolnej strukturze.

Połączenie analizy syfonów z weryfikacją modelową celem znajdowania wielu zakleszczeń w jednym przebiegu, w systemach o dowolnej strukturze, jest oryginalnym wkładem habilitanta. Translacja IMDS do sieci Petriego została zaimplementowana przez habilitanta w środowisku Dedan, zaś do wykrywania syfonów jest użyty program Charlie opracowany w Cottbus University [Heiner et al., 2015].

## Automaty DA<sup>3</sup>

Opracowano model automatów rozproszonych DA<sup>3</sup>, równoważnych IMDS, które są oryginalnym wkładem habilitanta. Automaty te pozwalają na graficzne projektowanie systemów rozproszonych oraz na graficzną symulację w kategoriach komponentów rozproszonych [H9]. Opracowane przez habilitanta metody specyfikacji systemów rozproszonych: formalizm IMDS, podzbiór sieci Petriego równoważny formalizmowi IMDS i automaty rozproszone DA<sup>3</sup>, stanowią triadę która służy do wszechstronnej analizy zachowań systemów: IMDS do weryfikacji modelowej i symulacji w kategoriach przestrzeni osiągalności, sieci Petriego do wykrywania wielu zakleszczeń jednocześnie i do innych aspektów analizy strukturalnej (np. wykrywanie martwego kodu), a automaty rozproszone DA<sup>3</sup> do graficznej specyfikacji i do symulacji w kategoriach komponentów systemu rozproszonego.

Automaty rozproszone DA<sup>3</sup> są użyte w sposób nieformalny w wielu pracach jako ilustracja graficzna przykładów systemów IMDS: [H3] [H4] [H5] [H8].

## Systemy transportowe

W pracach [H1], [H2], [H6], [H9] poruszono wiele aspektów projektowania, symulacji i weryfikacji systemów transportowych. W [H1] i [H2] zaproponowano rozproszone algorytmy zarządzania flotą pojazdów sieci PRT/ATN (Personal Rapid Transit/Autonomous Transit Networks). Rozproszone zarządzanie nie wymaga utrzymywania centralnej bazy danych o pojazdach i zapotrzebowaniu, oraz wspiera skalowalność autonomicznego transportu. W literaturze są opisane wyłącznie scentralizowane algorytmy zarządzania.

W [H6] i [H9], oraz w pracy przyjętej na konferencję Depcos [Czejdo and Daszczuk, 2019] przedstawiono przykłady specyfikacji i weryfikacji systemów rozproszonych na przykładzie sterowania ruchem i manewrowania pojazdami. Automatyczna weryfikacja wspomaga projektowanie na zasadzie szybkiego prototypowania, sprawdzając kolejne wersje zasad ruchu i zarządzania pojazdami pod względem bezpieczeństwa i żywotności.

## Symulacja

W praktyce projektowania i weryfikacji systemów rozproszonych istotna jest możliwość ich symulacji. Symulacja systemów wyspecyfikowanych w formalizmie IMDS pozwala na obserwację określonych scenariuszy w trakcie wykonywania weryfikowanego systemu. Istotna jest także możliwość symulacji krok po kroku kontrprzykładów otrzymanych w wyniku poszukiwania zakleszczeń lub sprawdzania nieuchronności terminacji. Symulator taki został wbudowany przez habilitanta do weryfikatora Dedan [H9].

Drugi symulator Feniks, wykonany przez habilitanta w 50% (drugą część wykonał Waldemar Grabski), służy do symulacji algorytmów zarządzania flotą pojazdów w systemach autonomicznego transportu miejskiego (ATN – Autonomous Transport Networks, zwane również PRT – Personal Rapid Transit). Symulator ten (opisany w [H1], [H2], oraz w [Daszczuk, 2016]) służy do badań opracowanych algorytmów zarządzania flotą ATN. Symulator Feniks był i jest nadal wykorzystywany jako baza dla wielu prac dyplomowych, zarówno z zakresu symulacji, jak i samej budowy rozproszonego symulatora.

## Projektowanie algorytmów

Projektowanie algorytmów dla systemów rozproszonych zostało przedstawione na przykładzie algorytmów zarządzania flotą pojazdów systemu autonomicznego transportu miejskiego. Zaprojektowano rozproszony algorytm zarządzania pustymi pojazdami, poprawiający przepustowość sieci. Algorytm nie używa centralnej bazy danych o żądaniach przewozu, tylko opiera się na komunikacji sterowników przystanków i pojazdów w ograniczonym promieniu (zwanym horyzontem). Algorytm w wersji bez predykcji zapotrzebowania przedstawiono w [H1] a z predykcją w [H2]. Algorytm używa wieloparametrycznej sumy ważonej wielu zmiennych opisujących stan statyczny i dynamiczny sieci transportowej. W różnych sytuacjach różne składniki zyskują przewagę, dostosowując działanie do zmieniających się warunków ruchu i zapotrzebowania. W niezaliczonej do cyklu pracy [Daszczuk and Mieścicki, 2015] pokazano dostosowywanie się algorytmu do skrajnie nie zrównoważonych warunków.

Aspekt projektowania został też poruszony w [H6], w połączeniu z weryfikacją poprawności. W pracy pokazano, jak można projektować zachowania zbiorowe w przypadku zestawu mobilnych robotów patrolowych, jednocześnie weryfikując kolejne wersje.

## Manewrowanie

W kilku przykładach pokazano wykorzystanie formalizmu IMDS do modelowania i weryfikacji manewrowania pojazdami w różnych środowiskach rozproszonych, na przykład zespół autonomicznych robotów patrolowych [H6], przystanek systemu ATN/PRT [H9], skrzyżowanie [H9], rondo [H9]. W przykładach tych serwery pełnią rolę sterowników odcinków toru, a agenty – pojazdów. Korzystając z weryfikacji modelowej, można użyć zasady „szybkiego prototypowania”, w którym kolejne wersje protokołu współpracy są sprawdzane pod kątem wykrywania zakleszczeń oraz nieuchronności rozproszonej terminacji (czyli osiągnięcia zamierzonego celu pojazdu).

## Rozproszone zarządzanie

Zaprojektowano i zbadano przy pomocy metod symulacyjnych rozproszony algorytm zarządzania pustymi pojazdami w sieci ATN/PRT. Przedstawiono działanie algorytmu bez predykcji [H1] i z predykcją żądań przewozu [H2]. Habilitant zaproponował średniokwadratową miarę jakości pracy sieci. Badania wskazują na dynamiczne dostosowywanie się floty do zmiennych warunków zapotrzebowania, na przykład w pracy spoza cyklu [Daszczuk and Mieścicki, 2015].

### 5.3. Szczegółowy opis uzyskanych wyników

#### Zintegrowany Model Systemów Rozproszonych (IMDS)

Istnieje kilka modeli równoległego wykonywania w systemach komputerowych [Kessler and Keller, 2007]. Można je ogólnie podzielić na modele synchroniczne i asynchroniczne [Milner, 1983] [Savoiu et al., 2002] [R. van Glabbeek et al., 2008]. W modelach synchronicznych, współpracujące procesy (lub wątki, zadania lub inne działania równoległe) muszą komunikować się będąc jednocześnie w określonych stanach. Dlatego przynajmniej komunikujące się procesy muszą uzgodnić swoje stany lub, alternatywnie, uczestniczyć w określony sposób w globalnym stanie systemu. Przykładami synchronicznych formalizmów są: LOTOS [Rosa and Cunha, 2004], automaty Büchiego [Baier and Katoen, 2008] stosowane do weryfikacji układów wykorzystujących liniową logikę czasową LTL [Baier and Katoen, 2008], automaty Zielonki [Zielonka, 1987] lub automaty czasowe TA [Baier and Katoen, 2008]. W każdym z tych modeli automaty synchronizują się za pomocą wspólnych symboli na przejściach. Z drugiej strony, synchronizacja w CSP [Hoare, 1978] i Automatach Czasowych Uppaal [Behrmann et al., 2011] odbywa się przez wykonanie równoczesnych operacji wysyłania i odbierania. Podobnie synchroniczne operacje na uzupełniających wejściowych i wyjściowych portach w CCS [Milner, 1980] i w Occamie [May, 1983] wymagają jednoczesności określonych działań procesów.

Habilitant twierdzi, że powyższe modele są niewystarczające do modelowania systemów rozproszonych, ponieważ takie systemy nie posiadają globalnego (lub nielokalnego) stanu. W rzeczywistości procesy nie mogą uzgodnić ich stanów. Jedynym sposobem wpływania na zachowanie innego serwera jest wysyłanie do niego wiadomości. Z drugiej strony, otrzymanie odpowiedzi nie gwarantuje, że serwer odpowiadający nadal znajduje się w stanie zgłoszonym w odpowiedzi. Dlatego asynchroniczne formalizmy, w których procesy nie uzgadniają wspólnych stanów, są bardziej realistyczne w modelowaniu systemów rozproszonych [Johnsen et al., 2009].

Formalizmy Message Passing Automata (MPA [Bollig and Leucker, 2004]) i Pushdown Distributed Automata (PDA [Balan, 2009]) są asynchroniczne, ponieważ odebrane komunikaty czekają na akceptację w określonych strukturach: kolejkach lub stosach. Zapewniona jest lokalność działań, ponieważ akceptacja komunikatu zależy tylko od lokalnego stanu odbierającego automatu i lokalnie przechowywanego komunikatu: ten, który czeka najdłużej w kolejce lub, odpowiednio, przez najkrótszy czas na stosie. Jednak w obu formalizmach automaty nie są w pełni autonomiczne, ponieważ nie decydują lokalnie, która wiadomość jest

akceptowana jako pierwsza (decyduje się na podstawowej zasadzie: odpowiednio FIFO lub LIFO).

Wiele środowisk modelowania wykorzystuje jeden z paradygmatów komunikacji [Jia and Zhou, 2005]: przekazywanie wiadomości w paradygmacie klient-serwer lub paradygmat RPC (Remote Procedure Call – zdalne wywołanie procedury) z wiadomościami służącymi jako sposób migracji procesów. Żadne tych środowisk nie zapewnia dualizmu obu form komunikacji w zunifikowanym formalizmie. Zaproponowany przez habilitanta i Stanisława Chrobotą formalizm Zintegrowany Model Systemów rozproszonych IMDS [Chrobot and Daszczuk, 2006] [H3] [H4] [H9] oferuje ten dualizm jako dwie projekcje modelowanego systemu: na serwery lub na agenty. W obecnej formie [H4] [H9] rzutowania polegają po prostu na grupowaniu akcji: na poszczególnych serwerach lub w pojedynczych agentach. Jest to zawsze możliwe, ponieważ każda akcja ma na wejściu i na wyjściu parę: stan serwera i komunikat agenta. Ten dualizm umożliwia wyrażanie różnych własności, które manifestują się w poszczególnych widokach: zakleszczenia komunikacyjne w widoku serwerów, zakleszczenia na zasobach i rozproszona terminacja w widoku agentów. Prace te, rozpoczęte wspólnie z S. Chrobotem, po jego przedwczesnej śmierci są kontynuowane samodzielnie przez habilitanta od 2006 roku.

W znanej pracy [Lauer and Needham, 1979] znajduje się postulat równoważności komunikacji przez wiadomości i przez zasoby (zmienne) [tłumaczenie i skróty habilitanta]:

„Aby scharakteryzować nasze klasyfikacje, skonstruowaliśmy model kanoniczny dla każdej kategorii. Model zorientowany na wiadomości charakteryzuje się [...] zestawem kanałów komunikacyjnych między nimi, [...]. Model zorientowany na procedury charakteryzuje się [...] komunikacją za pomocą bezpośredniego udostępniania i blokowania danych [...]. Te dwa modele definiują dwa różne rodzaje operacji pierwotnych do zarządzania procesami i synchronizacją w systemie operacyjnym [...]:

1. Oba modele są dualne. [...] Prymitywy zdefiniowane przez jeden model mogą być mapowane bezpośrednio do dualnego programu lub podsystemu, który spełnia założenia drugiego modelu.
2. Dualne programy lub podsystemy są logicznie identyczne.
3. Wydajność programu lub podsystemu z jednego modelu, odzwierciedlona przez jego zawartość kolejek, [...] jest identyczna z jego dualnym systemem [...]. Ponadto prymitywy [...] jednego modelu można uczynić tak wydajnymi, jak ich dualne operacje drugiego modelu. ”

W modelu IMDS jeden i ten sam system jest rzutowany na podsystemy (serwery) komunikujące się przez wiadomości lub na inne podsystemy (agenty) komunikujące się przez zasoby (stany serwerów). IMDS wykazuje więc przez konstrukcję poprawność postulatu Needhama-Lauera, a nawet idzie dalej gdyż realizuje oba typy współpracy w jednym modelu z dwoma rzutami, a nie w postaci dwóch równoważnych modeli. Ponadto, rozszerza realizację postulatu na systemy rozproszone, podczas gdy oryginalnie został on sformułowany dla systemu skupionego.

W IMDS akcje są wykonywane na serwerze na podstawie lokalnego stanu i oczekujących lokalnie komunikatach. Umożliwia to modelowanie systemów rozproszonych w sposób

asynchroniczny, co jest realistyczne i naturalne. Habilitant twierdzi, że IMDS nadaje się dobrze do modelowania systemów rozproszonych, ponieważ akcje na serwerach zachowują:

- *Lokalność*. W systemach rozproszonych brak jest globalnego stanu, na podstawie którego mogą być rozpoczęte i kontynuowane rozproszone obliczenia. Oczywiście w celach weryfikacyjnych taki globalny stan musi być w jakiś sposób wyekstrahowany z systemu, ale żadne decyzje co do akcji rozproszonych elementów nie mogą od tego stanu zależeć. Do opisu stanu i zachowania elementu rozproszonego systemu muszą wystarczyć składowe jego lokalnego stanu. Oprócz wartości lokalnych zmiennych opisujących lokalny stan, występują komunikaty otrzymane od innych serwerów systemu, oczekujące na akceptację. Stan globalny systemu, rozumiany jako połączenie stanów lokalnych jego składowych, służy do zbudowania struktury która opisuje zachowania systemu, a akcje w ramach tej struktury są wykonywane przeplotowo. W pracy [Manna and Pnueli, 1992] pokazano, że semantyka przeplotowa jest równoważna równoległemu złożeniu.
- *Autonomia*. Elementy systemu rozproszonego podejmują decyzje z sposób autonomiczny, samodzielnie wybierając które lokalne akcje i kiedy zostaną wykonane. Jedynym sposobem wpływania na zachowanie elementu systemu rozproszonego jest wysłanie do niego komunikatu, który zostanie tam autonomicznie zinterpretowany jak potencjalny wyzwalacz lokalnych akcji. Nie zakłada się żadnego predefiniowanego porządku wśród oczekujących na zaakceptowanie komunikatów.
- *Asynchronizm*. Ponieważ akcje są podejmowane w serwerach systemu rozproszonego w sposób nienależny, bazując wyłącznie na ich lokalnych stanach, nie jest założony żaden porządek w którym akcje te są wykonywane. Żadna akcja nie zależy od stanu innych serwerów systemu ani od wykonywanych na nich akcjach. Asynchronizm wewnątrz serwera jest modelowany w ten sposób, że serwer akceptuje komunikat gdy jest do tego gotowy, w przeciwnym wypadku to komunikat czeka na zaakceptowanie. Również komunikacja jest asynchroniczna: dostarczenie komunikatu jest niezależne od tego, czy adresat jest gotów go zaakceptować. Te komunikaty, które nie są zaakceptowane, po prostu oczekują w serwerze do którego są adresowane.
- *Dualizm komunikacyjny*. Do opisu działania systemów rozproszonych są używane dwa modele: klient-serwer i zdalne wywołanie procedur (RPC) [Jia and Zhou, 2005]. Pierwszy z nich wiąże procesy systemowe z niezależnymi, komunikującymi się węzłami, a drugi zakłada przenoszenie się procesów pomiędzy węzłami wraz ze zdalnymi wywołaniami procedur i powrotami z tym wywołań. Ważne jest uświadomienie sobie, że te dwa paradygmaty nie są zupełnie odrębne od siebie. Można każdy system rozproszony rozpatrywać w jednym lub drugim ze wspomnianych modeli, zależnie od tego jak zostaną wyodrębnione sekwencje akcji. Jeżeli połączymy w sekwencję akcje wykonywane na poszczególnych węzłach – otrzymamy model klient-serwer (przy czym oczywiście poszczególne węzły mogą występować w stosunku do innych w obu rolach – klienta i/lub serwera), a komunikaty staną się medium komunikacyjnym pomiędzy węzłami. W drugiej strony, jeżeli skojarzymy procesy ze zdalnymi wywołaniami procedur – otrzymamy model

rozproszonych obliczeń (które nazywam agentami) przemieszczających się pomiędzy węzłami przy pomocy komunikatów, a stany wewnętrzne węzłów staną się medium komunikacyjnym pomiędzy procesami. Ten model może być nawet uogólniony: agent reprezentujący rozproszone obliczenie nie musi powracać do wywołującego go węzła jak w modelu RPC, lecz może przemieszczać się dalej przez kolejne węzły i kończyć się w węźle innym niż ten, który wystartował obliczenie.

Formalizm IMDS można przedstawić jako relację  $\mathcal{A} \subset (M \times P) \times (M \times P)$ , gdzie  $M$  jest zbiorem komunikatów,  $P$  zbiorem stanów serwerów a  $\mathcal{A}$  jest zbiorem akcji w systemie. Akcja  $\lambda$  jest elementem relacji  $\mathcal{A}: \lambda = ((m, p), (m', p')) \in \mathcal{A}$ , gdzie para  $(m, p)$  jest komunikatem i stanem wejściowym, zaś para  $(m', p')$  komunikatem i stanem wyjściowym. Akcje są atomowe, to znaczy nie są rozróżniane fazy przyjmowania komunikatu, wykonywania obliczeń zmieniających stan serwera i faza generowania nowego komunikatu. Procesy są wyodrębniane jako zbiory akcji: procesy serwerów grupują akcje na poszczególnych serwerach, a procesy agentów grupują akcje poszczególnych agentów. Zbiór procesów serwerów to widok serwerowy systemu, w którym stany serwerów są nośnikami procesów, a procesy współpracują się poprzez przesyłanie komunikatów. Zbiór procesów agentów to widok agentowy systemu, w którym komunikaty są nośnikami procesów, a procesy współpracują poprzez stany serwerów.

W celu odróżnienia terminacji od zakleszczenia (w obu przypadkach procesy nie mają swojej kontynuacji, ale z różnych przyczyn), wprowadzono akcje terminujące procesy agentów. W tym przypadku na wyjściu akcji występuje jedynie singleton  $(p')$ , oznaczający nowy stan serwera, a brak jest wyjściowego komunikatu. Akcja terminująca ma postać  $(m, p) \mathcal{A} (p')$ , tak więc cały zbiór akcji należy do sumy dwóch relacji  $\mathcal{A} \subset (M \times P) \times (M \times P) \cup (M \times P) \times (P)$ .

Formalizm IMDS jest opisany w pierwszej, bardziej skomplikowanej poprzedniej wersji w [H3], a aktualna wersja jest przedstawiona w [H4], [H9].

**Wkład habilitanta:** opracowanie formalizmu IMDS w obecnej postaci (100%), w pierwotnej postaci 50%

**Wkład w rozwój:** opracowanie formalizmu IMDS (100%)

### Częściowe własności (zakleszczenie i terminacja), automatyzacja weryfikacji modelowej

Drugą zasadniczą motywacją do przedstawionych badań była automatyzacja procesu weryfikacji systemów rozproszonych. Opublikowano wiele metod wykazywania poprawności zachowań systemów równoległych, w tym rozproszonych. Metody te poszukują błędów dynamicznie, w czasie wykonywania obliczeń w systemach, lub też statycznie, analizując specyfikacje systemów. Szeroki wykaz metod weryfikacji podano w [H9]. Wśród metod statycznych istotną rolę pełni weryfikacja modelowa (ang. model checking) oparta o wyrażanie cech w logikach temporalnych i ewaluację formuł temporalnych w celu znalezienia błędu lub dowiedzenia poprawności. Weryfikacja modelowa ma wiele odmian, ale wspólną ich bolączką jest wyrażana nawet w prasie naukowej niechęć użytkowników do ich stosowania. Istotną

barierą jest konieczność nauczenia się przez użytkownika logiki temporalnej do wyrażenia określonych cech. W takiej sytuacji bardzo ważna jest automatyzacja weryfikacji temporalnej, tak aby użytkownik po dokonaniu specyfikacji mógł od razu uzyskać odpowiedź na pytania o poprawność rozproszonego systemu. Większość weryfikatorów temporalnych umożliwia takie sprawdzenie dla określonego zestawu cech, typowo dla zakleszczenia (czyli według definicji habilitanta, stanu w którym procesy nie mogą kontynuować swojej pracy oczekując na niemożliwe zdarzenia).

W opinii habilitanta jest to jednak niewystarczające, szczególnie w obliczu powstawania wielu systemów według schematu internetu rzeczy (ang. Internet of Things, IoT), gdzie rozproszone sterowniki uzgadniają ich skoordynowane zachowanie w celu osiągnięcia określonego celu. W takich systemach mogą występować częściowe zakleszczenia, w których część procesów nie może kontynuować swojej pracy, ale inne nadal pracują. Takie zakleszczenie habilitant nazywa częściowym (odróżnieniu od totalnego, w którym występują wszystkie procesy). Częściowe zakleszczenia wykrywają tylko niektóre weryfikatory, i to dla bardzo ograniczonych schematów budowy systemu (na przykład tylko czysto cykliczne lub czysto terminujące). Jeżeli system nie odpowiada założonemu schematowi, częściowe zakleszczenie nie może zostać wykryte automatycznie, i wtedy użytkownik musi sam zapisać odpowiednie formuły temporalne.

W modelu IMDS zaprojektowano formuły w logice temporalnej (zarówno wersji z czasem liniowym LTL jak i z czasem rozgałęzionym CTL), zapewniające automatyczne znajdowanie częściowych zakleszczeń. Zależnie od sposobu wyodrębnienia procesów, są to zakleszczenia komunikacyjne (w widoku serwerów) lub na zasobach (w widoku agentów). Nie każde zakleszczenie na zasobach objawia się jako zakleszczenie komunikacyjne, na przykład gdy wszystkie serwery pracują (brak zakleszczenia komunikacyjnego) ale pewne agenty nie są obsługiwane z braku zasobów (zakleszczenie na zasobach).

W systemie rozproszonym może występować terminacja procesów (jeżeli określone obliczenia kończą się). Jest to inna forma dyskontynuacji procesów. Różni się ona od zakleszczenia tym, że zakończenie jest celowe a nie w wyniku błędu. Nie wszystkie weryfikatory temporalne odróżniają rozproszoną terminację od zakleszczenia. W formalizmie IMDS zaprojektowano formuły temporalne wykrywające terminację, zarówno totalną jak i częściową.

Na marginesie uwaga terminologiczna: zakleszczenia/terminacje które habilitant określa jako totalne/częściowe (ang. total/partial) w literaturze często są nazywane globalnymi/lokalnymi. W kontekście systemów rozproszonych określenie „lokalne” może być mylące, gdyż kojarzy się z sytuacją lokalną topograficznie, czyli w ramach np. jednego serwera, podczas gdy zakleszczenie częściowe może dotyczyć wielu serwerów (choć być może nie wszystkich), lub nawet wszystkich serwerów, ale tylko niektórych agentów.

**Wkład habilitanta:** formuły temporalne wykrywające zakleszczenia totalne i częściowe, zakleszczenia komunikacyjne i na zasobach, rozproszoną terminację totalną i częściową, odróżniające zakleszczenie od terminacji (100%)

**Wkład w rozwój:** jak wyżej (100%)



## Przykłady weryfikacji

W artykułach przedstawiono wiele przykładów specyfikacji i weryfikacji systemów rozproszonych. Są wśród nich przykłady akademickie, jak np., 5 filozofów [Dijkstra, 1971], oraz bardziej praktyczne przykłady jak model referencyjny (ang. benchmark) Karlsruhe Production Cell [Lewerentz and Lindner, 1995] – [H7], [H9], kilka modeli związanych z manewrowaniem pojazdów [H6], [H9]. Model Karlsruhe jest szczególnie ciekawy gdy działa w zamkniętej pętli cyrkulacji detali, ponieważ w jednym z urządzeń następuje wymiana dwóch detali. Prowadzi to sytuacji, w której potrzebujemy co najmniej dwóch detali do ich bezpiecznej cyrkulacji w gnieździe, a pojedynczy detal grzęźnie powodując zakleszczenie. Występowanie zakleszczenia zależy więc od liczby detali. Również praca gniazda w otwartej pętli, to znaczy gdy gniazdo jest zasilane detalami na wejściu, a na wyjściu znikają one z gniazda, powoduje zakleszczenie ostatniego z detali.

Jeden z modeli manewrowania pojazdów pokazuje nieoczywisty przypadek, gdy w systemie występuje zakleszczenie na zasobach, podczas gdy nie ma zakleszczenia komunikacyjnego. Dzieje się to w przypadku, gdy dla jednego z agentów brakuje zasobów (uosiabianych przez serwery), a wszystkie procesy serwerów pracują, obsługując pozostałe agenty.

Omówiono przykłady systemów wygenerowanych ze specyfikacji w języku wyższego poziomu Rybu, opracowanego przez studentów pod kierownictwem habilitanta [H9]; wśród nich system mający 5955 wierszy kodu źródłowego, ponad 5500 akcji. Ponadto, ponad 200 studentów przeprowadziło weryfikację swoich rozwiązań synchronizacyjnych przy pomocy programu Dedan. Program ten był wykorzystany w pracach dyplomowych.

Nie włączony do osiągnięcia naukowego artykuł [Czejdo and Daszczuk, 2019] (artykuł ukaże się w druku w lipcu 2019) przedstawia zasadę „szybkiego prototypowania”, to znaczy opracowywania alternatywnych rozwiązań z natychmiastowym sprawdzaniem ich poprawności.

**Wkład habilitanta:** opracowanie rozproszonej wersji modeli referencyjnych znanych z literatury, opracowanie własnych modeli referencyjnych, badania modeli (100%)

## Inne metody weryfikacji oparte o formalizm IMDS

W monografii [H9] opisano trzy metody weryfikacji systemów IMDS odmienne od „klasycznej” weryfikacji modelowej opartej o ewaluację formuł temporalnych:

- [H5] [H9] Weryfikacja modelowa pozwala znaleźć jedno zakleszczenie, a następnie po poprawieniu systemu i poddaniu ponownej weryfikacji. Translacja IMDS do sieci Petriego w celu znajdowania wielu zakleszczeń w jednej weryfikacji. Analiza syfonów sieci Petriego, znana z literatury [Zuberek, 2009], pozwala na identyfikację zakleszczeń w czysto cyklicznych systemach. Syfon jest podsiecią, która pozbawiona znaczników, nie potrafi ich odtworzyć. Istnieją bardzo efektywne algorytmy wykrywania w jednym przebiegu wszystkich elementarnych syfonów, to znaczy takich, które nie zawierają żadnych innych syfonów. Dodatkowo, należy zbadać czy konkretne znalezione syfony mogą zostać opróżnione (jeżeli nie, to zakleszczenie jest

nieosiągalne) . W ten sposób można znaleźć w jednej weryfikacji wiele zakleszczeń, w tym zakleszczenia częściowe (choć być może nie wszystkie, gdyż syfon może zostać opróżniony na wiele sposobów).

Opisane w literaturze metody wykrywania zakleszczeń przy pomocy syfonów działają tylko dla czysto cyklicznych systemów. Systemy zawierające część początkową lub terminalną zawierają syfony które są opróżnialne, ale nie stanowią zakleszczeń. W pracach [H5], [H9] habilitant przedstawił analizę syfonów połączoną z weryfikacją modelową, co pozwala na znajdowanie zakleszczeń w systemach o dowolnej strukturze a nie tylko czysto cyklicznych.

- [H9] Wbudowany weryfikator temporalny TempoRG jest oparty o reprezentację przestrzeni osiągalności *explicite*, co pozwala na badanie raczej prostych przypadków. Duże systemy mogą być badane przez zewnętrzne weryfikatory, wykorzystujące redukcje częściowo-porządkowe oraz symboliczną reprezentację przestrzeni osiągalności [Clarke et al., 1999]. W systemie Dedan habilitant wbudował konwertery do języków wejściowych innych weryfikatorów. Jednak i w takiej weryfikacji może nastąpić przepełnienie dostępnej pamięci, jak pokazano w przypadku Karlsruhe Production Cell dla 5 detali [H7]. Do weryfikacji takich systemów są używane algorytmy niewyczerpującego przeszukiwania przestrzeni osiągalności, jak  $A^*$  [Francesca et al., 2011], algorytmy genetyczne [Godefroid and Khurshid, 2002] i mrówkowe [Francesca et al., 2011]. Jednak algorytmy te wykrywają wyłącznie totalne zakleszczenia, nie odróżniają ich od terminacji, a ponadto zawierają dużą liczbę parametrów których wartości są najczęściej ustalane arbitralnie, bez uzasadnienia. W monografii [H9] przedstawiono oryginalny algorytm „2 wagabundów”, który polega na losowym przeszukiwaniu przestrzeni osiągalności. Algorytm ten jako jedyny znajduje częściowe zakleszczenia, a ponadto jest bezparametryczny. Polega on na zastosowaniu dwóch procesów przeszukujących, z których jeden stawia hipotezę zakleszczenia, a drugi weryfikuje tę hipotezę. Podobna procedura dotyczy znajdowania rozproszonej terminacji.

Jakkolwiek w algorytmie możliwe są zarówno znalezienie nieistniejącego zakleszczenia, jak i pominięcie istniejącego, to pokazano na wielu przykładach jego skuteczność. O ile nieznanie zakleszczenia jest możliwe również w innych niewyczerpujących algorytmach, to możliwość znalezienia nieistniejącego zakleszczenia jest ceną za znajdowanie zakleszczeń częściowych. Jednak błędny wynik jest możliwy tylko w bardzo dużych systemach.

- [H9] Automaty Czasowe TA pozwalają na weryfikację z ograniczeniami czasu rzeczywistego [Alur and Dill, 1994]. Jednak, podobnie do automatów Büchiego, Automaty Czasowe są modelem synchronicznym. Habilitant opracował translację IMDS z ograniczeniami czasu rzeczywistego do automatów czasowych, z implementacją kanałów asynchronicznych IMDS jako złożenia dwóch kanałów synchronicznych TA. Pozwala to na przypisanie zakresów czasu trwania akcji i opóźnień kanałów komunikacyjnych. Przedstawiono dwie wersje ograniczeń czasowych: ograniczenie czasu przebywania w stanach lub ramy czasowe dla propagacji komunikatów pomiędzy danymi serwerami. Wykonanie dwóch wersji wynika z ograniczeń formalizmu automatów czasowych programu Uppaal, który jest

używany do weryfikacji. Ograniczenia te wykluczają jednoczesne występowanie ograniczeń czasowych dla stanów serwerów i dla czasu propagacji komunikatów. W obu wersjach można nałożyć ograniczenia czasowe na trwanie akcji.

Eksport modelu IMDS do Automatów Czasowych w programie Dedan pozwala na weryfikację w środowisku Uppaal [Behrmann et al., 2011]. W [H9] pokazano weryfikację prostego przykładu, lecz próba weryfikacji Karlsruhe Production Cell w programie Uppaal zakończyła się niepowodzeniem z powodu zbyt dużej zajętości pamięci. Trwają prace nad połączeniem weryfikacji z ograniczeniami czasu rzeczywistego z niewyczerpującym przeszukiwaniem.

**Wkład habilitanta:** translacja IMDS do sieci Petriego i wykazanie ich równoważności (100%), znajdowanie zakleszczeń systemów o dowolnej strukturze przy pomocy syfonów i weryfikacji modelowej (80%), niewyczerpujący algorytm weryfikacji częściowych zakleszczeń i terminacji oraz jego implementacja i badania (100%), translacja IMDS do automatów czasowych i badania weryfikacyjne (100%)

**Wkład w rozwój:** jak wyżej

### Automaty rozproszone DA<sup>3</sup>

Zaproponowane przez Habilitanta automaty rozproszone DA<sup>3</sup> są równoważne formalizmowi IMDS. Automaty mają dwie postacie: serwerową (każdy automat opisuje serwer) i agentową (każdy agent jest opisany automatem). Konstrukcja automatów (typu Mealy'ego [Dick and Yao, 2014]) podkreśla dualizm komunikacyjny: w automatach serwerowych węzły są stanami, tranzycje akcjami a etykiety tranzycji to pary: komunikat wejściowy i wyjściowy akcji. W automatach agentowych węzły to komunikaty należące do poszczególnych agentów, a tranzycje jak poprzednio są akcjami, ich etykiety to pary: stan wejściowy i nowy stan serwera (stan wyjściowy akcji). Złożenie zarówno automatów serwerowych jak i agentowych ma postać odpowiadającą LTS systemu IMDS. Automaty rozproszone służą do graficznego definiowania systemu rozproszonego i do symulacji w kategoriach elementów składowych systemu. Można przy pomocy automatów symulować kontrprzykład wynikowy weryfikacji modelowej systemu. Powyższy opis wskazuje, że automaty rozproszone DA<sup>3</sup> stanowią głównie wkład praktyczny do weryfikacji.

**Wkład habilitanta:** opracowanie i implementacja koncepcji automatów rozproszonych (100%), oprogramowanie symulacji kontrprzykładów w automatach rozproszonych i symulacji systemów rozproszonych w kategoriach komponentów (100%)

**Wkład w rozwój:** opracowanie koncepcji automatów rozproszonych (100%)

### Sprawiedliwość w weryfikacji temporalnej systemów rozproszonych

W systemach rozproszonych występują trzy zjawiska, które trzeba rozpatrywać w kontekście sprawiedliwości (ang. fairness). Są to: sprawiedliwość pomiędzy serwerami, która modeluje niezależność wykonywania akcji w różnych serwerach, sprawiedliwość w ramach serwera, która oznacza sprawiedliwe szeregowanie w nim akcji, oraz sprawiedliwość w ramach agenta, która modeluje niedeterministyczne wybory agentów. O ile dwie pierwsze formy wymagają

tak zwanej słabej sprawiedliwości weryfikatora (ang. justice), o tyle w trzecim przypadku użyte zewnętrzne weryfikatory (Spin [Holzmann, 1995], NuSMV [Cimatti et al., 2002] i Uppaal [Behrmann et al., 2011]) raportują fałszywe zakleszczenia wynikające z braku silnej sprawiedliwości (ang. compassion). Co prawda weryfikator NuSMV w jednej z form weryfikacji (LTL) zapewnia silną sprawiedliwość, ale jest ona osiągnięta poprzez zapisanie wszystkich możliwych rozejść niedeterministycznych jako prefiksów formuły temporalnej. Nawet niewielki model bufora międzyprocesowego wymaga 21 takich prefiksów, co prowadzi do nieakceptowalnie długich czasów weryfikacji. Habilitant opracował własny, sprawiedliwy algorytm weryfikacji CBS (Checking by Spheres [H8] [H9]), który ewaluje formuły biorąc pod uwagę kształt przestrzeni osiągalności. Pierwsza wersja algorytmu CBS została opisana w rozprawie doktorskiej. Obecna wersja została ograniczona do ewaluacji formuł wykrywających zakleszczenia i terminację. Wersja ta używa wstecznej osiągalności do określenia kształtu przestrzeni osiągalności i stosuje reguły weryfikacji zależne od kształtu tej przestrzeni.

**Wkład habilitanta:** opracowanie i implementacja sprawiedliwego algorytmu weryfikacji (100%)

**Wkład w rozwój:** opracowanie sprawiedliwego algorytmu weryfikacji (100%)

### **Modelowanie autonomicznego transportu miejskiego, rozproszone algorytmy zarządzania**

W ramach prac nad modelowaniem autonomicznego transportu miejskiego habilitant opracował wspólnie z Waldemarem Grabskim (wkład po 50%) algorytmy zarządzania flotą pojazdów, w tym najważniejszy rozproszony algorytm zarządzania pustymi pojazdami. W literaturze opisano kilka takich algorytmów (np. [Andréasson, 2003], [Carnegie et al., 2007], [Lees-Miller and Wilson, 2012]), ale wszystkie one opierają się na centralnej bazie danych pozycji pojazdów, aktualnych żądań przewozu i ich historii. Zaproponowany algorytm rozproszony opiera się na wymianie danych pomiędzy sterownikami sąsiednich przystanków i pojazdów w określonej odległości zwanej horyzontem. Algorytm używa ważonej sumy parametrów opisujących statyczny i dynamiczny stan sieci. W różnych sytuacjach różne składowe uzyskują przewagę, dostosowując działanie algorytmu do bieżących warunków. Ograniczenie horyzontu pozwala na rozproszoną implementację algorytmu, co wspiera skalowalność systemu transportowego.

W pracy [H1] opisano samo działanie algorytmu bez składowej predykcyjnej, a w pracy [H2] przedstawiono algorytm z włączoną predykcją i porównanie z wynikami innych algorytmów. W pracy [Daszczuk and Mieścicki, 2015], która nie została włączona do cyklu publikacji, przedstawiono działanie algorytmu w warunkach skrajnej nierównowagi żądań przewozu, w których inne algorytmy powodują pogorszenie działania systemu. Prace te były prowadzone w ramach projektu Eco-Mobilność, kierowanego przez Włodzimierza Choromańskiego na Wydziale Transportu (we współpracy z wydziałami: Elektroniki, Elektrycznym i Mechatroniki, [Choromański, 2011]), o budżecie ponad 6mln EUR. W ramach tych prac powstał symulator Feniks, użyty w badaniach [H1], [H2] i opisany w [Daszczuk, 2016]. Za prace te habilitant

otrzymał nagrodę zespołową I stopnia JM Rektora PW w roku 2014, oraz nagrodę zespołową II stopnia Prezesa Rady Ministrów RP w roku 2016.

Od roku 2016 symulator jest rozwijany przez studentów kierunku Informatyka pod kierownictwem habilitanta i W. Grabskiego. W ramach tych badań powstało kilka prac dyplomowych (niektóre poparte referatami na międzynarodowych konferencjach), dotyczących problemów energetycznych [Daszczuk and Belz, 2018], transportu intermodalnego [Zieliński, 2018] i rozpraszania samego symulatora [Lorenc et al., 2019] [Gałęcki and Daszczuk, 2019]. Obecnie są prowadzone dalsze prace dyplomantów.

Pewnym problemem w symulacji systemów PRT jest określenie miar jakości pracy systemu. Jako miarę jakości algorytmów zarządzania powszechnie stosuje się średni czas oczekiwania na pojazd przez pasażerów [Lees-Miller and Wilson, 2012] [Zheng et al., 2009] [Andréasson, 2003]. W osobistych kontaktach Ingmar Andréasson, uznany autorytet w tej dziedzinie, przychylił się do opinii habilitanta że jest to miara ułomna ponieważ mogą zdarzać się pojedyncze przypadki długiego oczekiwania, psujące opinię o systemie transportowym. W pracy [H2] habilitant zaproponował miarę nawiązującą do odchyłeń średniokwadratowych używanych w elektronice: średniokwadratowy czas oczekiwania (ang. Average Squared Waiting Time - ASWT):

$$ASWT = \sqrt{\frac{\sum_{full\ trips} T_{ft}^2}{number\ of\ full\ trips}}$$

gdzie  $T_{ft}$  oznacza czas oczekiwania w pełnym kursie, *full trips* – pełne kursy (z pasażerami), *number of full trips* – liczba pełnych kursów w czasie symulacji.

Drugim parametrem jest miara przeciążenia sieci. W pracy [Schweizer et al., 2012] przyjmuje się na przykład jako maksymalną taką liczbę pojazdów, że na każdym odcinku porusza się maksymalna liczba pojazdów z dozwoloną prędkością i odstępem. Takie postawienie sprawy jest nierealistyczne biorąc pod uwagę losowy rozkład żądań, losowe położenie pojazdów w sieci i konieczność rozwiązania konfliktów pierwszeństwa na połączeniach odcinków toru. Tu również habilitant zaproponował miarę średniokwadratową, lecz o naturze względnej [Daszczuk and Mieścicki, 2015] [Grabski and Daszczuk, 2017b]. Podstawą jest określenie spóźnienia względnego, to znaczy wielkości względnej o którą wzrasta czas podróży w stosunku do nominalnego czasu podróży z maksymalną dopuszczalną prędkością. Tak jak poprzednio, przyjęto miarę średniokwadratową – średniokwadratowe spóźnienie względne (ang. Average Squared Delay – ASD), to znaczy:

$$ASD = \sqrt{\frac{\sum_{full\ trips} \Delta_{ft}^2}{number\ of\ full\ trips}}$$

gdzie  $\Delta_{ft}$  oznacza spóźnienie względne w konkretnym pełnym kursie, a pozostałe parametry jak poprzednio. Dla typowych sieci o odległościach między przystankami rzędu do kilku kilometrów, rzadko zapełniona pojazdami sieć wykazuje spóźnienie względne rzędu 20%. Dla nietypowych sieci można przeprowadzić prosty eksperyment symulacyjny z niewielką liczbą

pojazdów w celu określenia podstawowego ASD. Następnie należy zwiększać liczbę pojazdów tak długo, aż parametr ASD wzrośnie do zadanej wartości (na przykład o połowę). Wzrost ten oznacza, że pojazdy zbyt często wchodzą ze sobą w interakcje i występują konflikty (na połączeniach odcinków toru i na wjazdach do przystanków). Tak otrzymaną liczbę pojazdów należy przyjąć za maksimum, powyżej którego sieć jest przeciążona.

Trzecim parametrem jest zdolność przewozowa sieci (ang. Maximum Ridership –  $RS_{max}$ ), tak jak poprzednie dwie oparta na eksperymencie [H1] [H2]. Dla danej sieci i danej liczby pojazdów przeprowadza się eksperyment z bardzo dużą liczbą pasażerów oczekującą na każdym przystanku. W ten sposób każdy pojazd kończący kurs na przystanku jest natychmiast zajmowany przez następnego pasażerów i pojazdy nigdy nie stoją oczekując na przystankach. Warunkiem powodzenia tego eksperymentu jest przyjęcie zrównoważonej (na przykład wypełnionej równomiernie) *macierzy źródło-cel* (ang. Origin-Destination Matrix, **ODM**). Tak określoną zdolność przewozową potwierdzają inne eksperymenty z losowym zapotrzebowaniem: gdy liczba żądań zbliża się do  $RS_{max}$ , kolejki pasażerów rosną nieograniczenie a czas oczekiwania wzrasta eksponencjalnie.

Habilitant zaproponował również miarę dla równomierności struktury sieci autonomicznego transportu, opartą o znormalizowaną entropię [Daszczuk and Belz, 2018].

W ramach prac nad topologią modeli, opracowano zbiór *modeli referencyjnych* (ang. benchmarks) sieci PRT [Mieścicki and Daszczuk, 2013].

**Wkład habilitanta:** rozproszone algorytmy zarządzania flotą (50%), symulator Feniks (50%), badania symulacyjne (70%), modele referencyjne topologii sieci PRT (30%), średniokwadratowe miary pracy sieci (100%), miary równomierności sieci i popytu oparte o entropię (100%), zdolność przewozowa (50%)

**Wkład w rozwój:** rozproszony algorytm koordynacji pojazdów (50%), badania symulacyjne (70%), modele referencyjne topologii sieci PRT (30%), średniokwadratowe miary efektywności pracy sieci (100%)

#### **Istotne publikacje poza cyklem:**

- [Choromański, Daszczuk, Grabski, et al., 2013] – referat na konferencji międzynarodowej APM-ATN, Phoenix, USA: konstrukcja symulatora Feniks i badania symulacyjne sieci PRT, **MNiSW:15** (50%),
- [Choromański, Daszczuk, Dyduch, et al., 2013] – referat na konferencji międzynarodowej WCTR, Rio de Janeiro, Brazylia: konstrukcja symulatora Feniks i badania symulacyjne sieci PRT, **WoS, MNiSW:15** (30%),
- [Daszczuk and Mieścicki, 2015] – referat na konferencji LogiTrans, Szczyrk, Polska: badanie algorytmu zarządzania flotą w ekstremalnie zmiennych warunkach zapotrzebowania (80%),
- [Daszczuk, 2016] – referat na konferencji LogiTrans, Szczyrk, Polska: konstrukcja zdarzeniowego symulatora sieci PRT, **MNiSW:7** (100%),

- [Grabski and Daszczuk, 2017a] – artykuł w Transport Problems: współpraca sieci PRT z szybką koleją miejską, **WoS**, **MNiSW:14** (50%),
- [Daszczuk and Belz, 2018] – referat na międzynarodowej konferencji BGC Geomatics (IEEE), Olsztyn, Polska: badanie pracy sieci PRT z ograniczeniami energii, **WoS**, **MNiSW:15** (70%)

**Nagroda zespołowa I stopnia JM Rektora PW** w roku 2014 za opracowanie symulatora transportu miejskiego, opracowanie algorytmów zarządzania flotą pojazdów i badania symulacyjne tych algorytmów

**Nagroda zespołowa II stopnia Prezesa Rady Ministrów RP** w roku 2016 za powyższe opracowania oraz za istotne publikacje

#### 5.4. Podsumowanie

We współczesnej informatyce coraz większe znaczenia mają systemy rozproszone. Od dawna są rozwijane systemy typu grid i klastry serwerów, a zdobywają popularność przetwarzanie w chmurze i internet rzeczy. Wszystkie te systemy wymagają odpowiednich do ich struktury i sposobu działania technik i formalizmów do projektowania, modelowania, specyfikacji, weryfikacji i symulacji. Proste rozszerzenie formalizmów dla systemów równoległych do środowisk rozproszonych nie odzwierciedla ich podstawowych i naturalnych cech takich jak lokalność operacji, autonomia decyzji, asynchronizm akcji i komunikacji.

Zaproponowany przez habilitanta Zintegrowany Model Systemów Rozproszonych IMDS łączy wszystkie wymienione cechy systemów rozproszonych. Ponadto, łączy w jednym modelu dwa paradygmaty programowania rozproszonego: rezydentnych serwerów komunikujących się przez komunikaty w modelu klient-serwer, oraz migrujących agentów w modelu zdalnych wywołań procedur lub w ogólniejszym modelu migrujących agentów.

Istotnym wkładem habilitanta jest opracowanie zasad weryfikacji modelowej, w której ogólne formuły temporalne, niezależne od struktury modelu, pozwalają wykryć totalne i częściowe zakleszczenia komunikacyjne i na zasobach oraz rozproszoną terminację. Dzięki ogólnym formułom weryfikacja może zostać przeprowadzona automatycznie, co pozwala na szybkie prototypowanie systemów rozproszonych poprzez natychmiastowe sprawdzanie ich własności. Opracowano również kilka innych metod weryfikacji o specyficznych cechach.

Generalną cechą systemów rozproszonych jest brak stanu globalnego, co z jednej strony jest problemem a z drugiej jest szansą na lokalne decyzje i decentralizację systemów. Pozwala to na konstruowanie rozproszonych algorytmów w paradygmacie internetu rzeczy, w którym niezależne węzły uzgadniają skoordynowane działanie. Taki algorytm habilitant zaproponował (wspólnie z W. Grabskim) dla zarządzania flotą pojazdów w systemie autonomicznego transportu miejskiego. Dzięki rozproszeniu i współpracy węzłów algorytm dostosowuje się do aktualnej sytuacji zapotrzebowania i warunków ruchu, uwzględniając prognozowanie na podstawie sytuacji w przeszłości.

## 6. Prace przed doktoratem

### 6.1. Automaty współbieżne CSM i weryfikacja modelowa

Większość prac w karierze naukowej habilitanta na Politechnice Warszawskiej, zarówno przed jak i po doktoracie, dotyczyła systemów rozproszonych (wcześniejsze prace w PSK MERA-SYSTEM również miały związek z tą tematyką). Doświadczenia związane ze specyfikacją i weryfikacją systemów rozproszonych obejmowały między innymi:

- Pierścieniowa sieć mikrokomputerów sterujących kasami [Daszczuk, 1988] – nieformalna specyfikacja spowodowała błędy które trzeba było usuwać w czasie uruchamiania protokołu.
- System operacyjny MOST dla komputera o konstrukcji rozproszonej – zostały zaprojektowane algorytmy odzyskiwania spójności systemu po utracie połączenia między jego modułami. Mechanizmy wielopoziomowych połączeń pomiędzy modułami pozawalały na uzyskanie wielu semantycznych zależności jednocześnie. Prace zostały opisane w [Daszczuk, 1991].
- Rozproszony system do identyfikacji i sterowania ciągłymi (w dziedzinie czasu) obiektami w czasie rzeczywistym, znacznie zautomatyzowany – prace nad wykazaniem poprawności przy pomocy automatów CSM i analizy ich przestrzeni osiągalności [Mieścicki, Daszczuk, and Nowacki, 1998].
- Rozproszony system sterowania czujkami w kopalni (umowa z Sultzer GmbH) – doświadczenia z praktycznej realizacji systemu rozproszonego.
- Rozproszony system typu SCADA do monitorowania obiektów przemysłowych ESS/SIG/PowerWatch – formalna weryfikacja protokołów, znalezienie błędu w protokole, który nigdy nie został zaobserwowany w praktyce, ale potencjalnie mógł prowadzić do zakleszczenia [Grabski et al., 1999].
- Badanie poprawności protokołu wymiany danych w sieci GSM, formalne wykazanie poprawności protokołu monitorowania przez wiadomości tekstowe SMS [Daszczuk, 1998] [Mieścicki, Flisiak, et al., 1998].
- Badania nad zachowaniem i współpracą mobilnych robotów „ograglaków”, budowanych w Laboratorium Systemów Rozproszonych Instytutu Informatyki PW, prace prowadzone wraz z Henrykiem Dobrowolskim i Jerzym Mieścickim, w ramach prac powstało ok. 10 prac dyplomowych, w tym 4 pod kierownictwem habilitanta (np. [Burak, 2003]).
- Analiza formalna fragmentów systemu awioniki dla śmigłowca poprzez specyfikację w CSM i weryfikację temporalną.

Prace nad systemami rozproszonymi były prowadzone między innymi w ramach grantów KBN [Mieścicki, 1991] (habilitant był wykonawcą) [Mieścicki, 1995] (habilitant był głównym wykonawcą), [Mieścicki, 2001] (habilitant był głównym wykonawcą). Główną osią tych prac był formalizm Concurrent State Machines (CSM, zwany również automatami współbieżnymi), opracowany przez Jerzego Mieścickiego [Mieścicki, 1994a].



Formalizm CSM jest oparty na modelu automatowym. Charakteryzuje się tym, że:

- W odróżnieniu od wielu systemów modelowania wykorzystujących przeplotowe semantyki [R. J. Glabbeek and Goltz, 1990], w modelu CSM automaty wykonują ruch synchronicznie. Ruch ten zawsze jest możliwy, ponieważ automaty są zupełne. Asynchronizm może być modelowany poprzez zastosowanie przejść ze stanu do siebie samego (ang. self-loops), z etykietą będącą negacją alternatywy etykiet przejść do innych stanów, lub nawet jako przejście bezwarunkowe (skutkujące niedeterminizmem jeżeli istnieją inne krawędzie wychodzące ze stanu).
- Zamiast pojedynczych symboli na przejściach, stosowane są wyrażenia Boolowskie (przekładające się na rodziny zbiorów symboli, włączając zbiór pusty dla przejść spontanicznych), co oznacza możliwą koincydencję sygnałów wejściowych automatu.

Formalizm CSM posłużył jako baza do implementacji przez zespół pracowników i studentów Instytutu Informatyki środowiska modelowania systemów współbieżnych jako automatów CSM w środowisku COSMA. Prace nad weryfikacją temporalną systemów [Baier and Katoen, 2008] [Katoen, 1999] habilitant rozpoczął w czasie stażu w Loyola University w Nowym Orleanie i kontynuował w Instytucie Informatyki PW. Opracował oryginalną logikę temporalną QsCTL (z kwantyfikacją po stanach w formułach i operatorami relatywnymi do automatów składowych) oraz oryginalny algorytm ewaluacji formuł temporalnych CBS (Checking by Spheres), pozwalający na wczesne kończenie ewaluacji przy spełnieniu określonych warunków [Daszczuk, 2001] [Daszczuk, 2002c]. Algorytm został zaimplementowany w programie TempoRG [Daszczuk, 2003a] w ramach systemu COSMA. Przeprowadzono weryfikację protokołów wymiany danych w sieci GSM [Daszczuk, Mieścicki, Krystosik, et al., 1998] [Daszczuk, 1998], poprawność zdarzeniowego protokołu nawiązywania i rozłączania połączenia w konstruowanym w Instytucie Informatyki systemie monitorowania ESS [Grabski et al., 1999]. W wyniku tych badań udało się zlokalizować i usunąć błąd logiczny w protokole.

Dalsze prace polegały na integracji weryfikatora TempoRG z systemem modelowania COSMA, generowaniu rozgałęzionych kontrprzykładów (w przypadku kiedy za negatywną ewaluację formuły odpowiadają dwie podformuły) [Daszczuk, 2002a], opracowaniu metody redukcji przestrzeni stanów dla automatów współbieżnych CSM (koincydencyjnych) [Daszczuk, 2000b] [Daszczuk, 2002b], makrogeneracji i organizacji bibliotek typowych automatów [Daszczuk, 2004] oraz weryfikacji symbolicznej [Daszczuk, 2000a] [Daszczuk, 2002d]. Przeprowadzono wiele badań konkretnych przypadków [Daszczuk, Mieścicki, et al., 2000] [Daszczuk, Grabski, et al., 2001] [Mieścicki et al., 2003] [Mieścicki et al., 2004b] [Mieścicki and Daszczuk, 2005] [Mieścicki and Daszczuk, 2006]. Prace te zostały podsumowane w rozprawie doktorskiej [Daszczuk, 2003b], w której zdefiniowano formalnie automaty CSM (niezależnie od autora formalizmu CSM, w sposób specyficzny ze względu na potrzeby weryfikacji modelowej), oraz przedstawiono algorytm weryfikacji CBS, metodę redukcji przestrzeni stanów i generowanie drzewiastych kontrprzykładów.

Po obronie doktoratu habilitant kontynuował prace nad weryfikacją systemów z ograniczeniami czasu rzeczywistego. Zdefiniował automaty CSM z ograniczeniami czasowymi i przedstawił sposób budowy przestrzeni osiągalności automatów CSM z czasem rzeczywistym

[Daszczuk, 2007], jednak badania nad weryfikacją z czasem rzeczywistym nie były kontynuowane, między innymi z powodu rozpoczęcia prac nad modelowaniem systemów transportowych i nad nowym formalizmem dla systemów rozproszonych IMDS.

**Wkład habilitanta:** samodzielne zdefiniowanie formalizmu CSM (Concurrent State Machines) na potrzeby weryfikacji modelowej (100%), logika temporalna QsCTL (100%), algorytm ewaluacji formuł temporalnych CBS (100%), zdefiniowanie rozgałęzionych kontrprzykładów (100%), metoda redukcji przestrzeni stanów w systemach koincydencyjnych (nie zaimplementowana, 100%), zdefiniowanie automatów CSM z ograniczeniami czasu rzeczywistego (TCSM, 100%), wyznaczanie przestrzeni stanów automatów TCSM (100%), badania temporalne wielu przykładów (50%)

**Wkład w rozwój:** wszystkie powyższe elementy

**Istotne publikacje:**

- [Mieścicki et al., 1996] – referat na międzynarodowej konferencji IDPS'96, Houston, USA: specyfikacja systemów rozproszonych w formalizmie CSM (20%),
- [Czejdo et al., 1998] – referat na konferencji IDPS'98, Berlin, Niemcy: specyfikacja systemów rozproszonych w formalizmie CSM (30%),
- [Daszczuk, Mieścicki, et al., 2001] – referat na konferencji MIXDES, Zakopane, Polska: modelowanie sterowania światłami drogowymi w formalizmie CSM, **WoS**, **MNiSW:10** (25%),
- [Daszczuk, Grabski, et al., 2001] – referat na międzynarodowej konferencji Euromicro DSD, Warszawa, Polska: specyfikacja i weryfikacja systemu rozproszonego w formalizmie CSM, **WoS**, **MNiSW:10** (30%),
- [Daszczuk, 2001] – referat na międzynarodowej konferencji Euromicro DSD, Warszawa, Polska: algorytm weryfikacji temporalnej z kwantyfikacją po stanach i specyficznymi operatorami temporalnymi, **WoS**, **MNiSW:10** (100%),
- [Daszczuk, 2003b] – rozprawa doktorska, opracowana logika temporalna, algorytm ewaluacji formuł, redukcja przestrzeni stanów, drzewiaste kontrprzykłady (100%),
- [Mieścicki et al., 2004a] – referat na międzynarodowej konferencji ECCOMAS, Jyväskylä, Finlandia: specyfikacja i weryfikacja systemów rozproszonych w formalizmie CSM, **WoS**, **MNiSW:10** (30%),
- [Mieścicki et al., 2004b] – referat na międzynarodowej konferencji ECCOMAS, Jyväskylä, Finlandia: specyfikacja i weryfikacja systemów rozproszonych w formalizmie CSM, **WoS**, **MNiSW:10** (20%),
- [Mieścicki and Daszczuk, 2006] – artykuł w Annales UMCS, Informatica AI: zastosowanie formalizmu CSM do weryfikacji z czasem rzeczywistym, **MNiSW:6** (50%),
- [Daszczuk, 2007] – artykuł w Computer Science, AGH: rozszerzenie formalizmu CSM o mechanizmy zbliżone do automatów czasowych, **MNiSW:6** (100%).

**Nagroda zespołowa JM Rektora PW** za prace nad specyfikacją i weryfikacją systemów rozproszonych - 1997

## 6.2. Systemy automatycznej regulacji i identyfikacja obiektów

Od zatrudnienia w Politechnice Warszawskiej do roku 2000 habilitant brał udział, pod kierownictwem Jerzego Mieścickiego, wraz z Konstantym Janem Kurmanem, w szeregu prac związanych z systemami automatycznej regulacji, zgodnie z oryginalną metodą odsprzęgania pętli regulacyjnych opracowaną przez K.J. Kurmana. Prace były prowadzone w ramach grantu MEN [Mieścicki, 1990], grantu KBN [Miller, 1997], grantu rektorskiego [Mieścicki, 1994b] oraz umów dwustronnych w kilku przedsiębiorstwach: Cukrowni Malbork (wyparka), Cukrowni Garbów (wyparka), Wistom Tomaszów (instancja suszenia wiskozy) i Petrochemii Płockiej (desorber w instalacji odsiarczania gazów suchych), oraz w eksperymentalnych instalacjach Politechniki Warszawskiej: instalacji zapełniania cylindrów, kolumny destylacyjnej D-1 w Laboratorium Procesów Technologicznych PW oraz paleniska fluidalnego w Instytucie Techniki Ciepłej PW. Prace te pozwoliły na opracowanie nowych algorytmów automatycznej regulacji [Mieścicki, Kurman, et al., 1998] [Mieścicki et al., 2000].

Problemem przy opracowywaniu algorytmów sterowania była identyfikacja obiektów. Klasyczne zasady identyfikacji nakazują dokonania skoku jednostkowego w otwartej pętli regulacji i na tej podstawie wnioskuje się o parametrach transmitancji obiektu: wzmocnieniu, inercji i opóźnieniu. Jednak zarówno w obiektach przemysłowych, jak i w instalacjach eksperymentalnych Politechniki Warszawskiej, względy bezpieczeństwa i praktyki procesowej wykluczały eksperymenty w otwartej pętli regulacji, a nawet wykonanie skoku jednostkowego o istotnej wielkości w zamkniętej pętli sterowania. Stawiało to pod znakiem zapytania możliwość dokonania identyfikacji obiektu w celu opracowania algorytmu sterowania.

W tej sytuacji w zespole opracowano nowatorską metodę identyfikacji obiektów MIKOZ. Metoda polega na dodawaniu do sygnału sterującego w rzeczywistej pętli sterowania kilku sinusoid o wzajemnie pierwszych pulsacjach, oraz analizie odpowiedzi obiektu [Daszczuk et al., 1994] [Mieścicki et al., 1995a] [Mieścicki et al., 1995b]. Metoda została skutecznie zastosowana w przypadku kilku obiektów, w tym kotła paleniska fluidalnego. Udało się zidentyfikować 8 inercji obiektu, w tym jedną dwukrotną (czyli dwie inercje o identycznej wartości). Analiza fizycznej struktury obiektu potwierdziła znakomitą skuteczność metody MIKOZ.

Prace nad identyfikacją i sterowaniem obiektów przemysłowych wymagały opracowania zasad współpracy komputerów zaangażowanych w ten proces i rozwiązania problemów dotyczących regulacji w środowisku rozproszonym [Mieścicki, Daszczuk, Grabski, et al., 1998] [Grabski et al., 1998] [Mieścicki, Daszczuk, and Nowacki, 1998].

Habilitant był jedynym autorem pierwszej wersji oprogramowania KURMAN służącego do identyfikacji i sterowania, w tym realizacji regulatorów w czasie rzeczywistym [Mieścicki et al., 1992] [Kurman et al., 1992]. W późniejszych fazach budowy tego środowiska, elementy rozbudowy zostały wykonane przez studentów Instytutu Informatyki [Daszczuk, Mieścicki, Kochan, et al., 1998].

Do zastosowania metody MIKOZ stworzono rozproszony system automatyzujący poszczególne etapy identyfikacji i sterowania. Przeprowadzono badanie poprawności współpracy

komputerów w rozproszonym systemie KURMAN przy pomocy automatów CSM (o których była mowa powyżej), wyznaczenia ich przestrzeni osiągalności i jej analizy [Daszczuk, Mieścicki, Kochan, et al., 1998].

**Wkład habilitanta:** współudział w opracowaniu metody MIKOZ (30%), sterowanie w czasie rzeczywistym na podstawie transmitancji regulatorów zapisanych w dziedzinie operatora  $s$  (100%), realizacja oprogramowania do identyfikacji i sterowania KURMAN (80%), badania identyfikacyjne obiektów (100%), badanie poprawności systemu KURMAN (30%)

**Wkład w rozwój:** metoda MIKOZ (30%), sterowanie w czasie rzeczywistym na podstawie transmitancji regulatorów zapisanych w dziedzinie operatora  $s$  (100%)

**Prace te zostały podsumowane w raportach badawczych IIPW:**

- [Mieścicki et al., 1995b] – raport badawczy IIPW: metoda identyfikacji obiektów i realizacji oddziaływań zwrotnych MIKOZ (30%),
- [Mieścicki, Kurman, et al., 1998] – raport badawczy IIPW: identyfikacja i sterowanie dla paleniska fluidalnego (30%),
- [Mieścicki et al., 2000] – raport badawczy IIPW: identyfikacja i sterowanie dla desorbera w instalacji odsiarczania gazów suchych (30%)

### 6.3. Współpraca z Sultzer GmbH – sieć czujek

W latach 1994-1995 we współpracy z firmą Sultzer GmbH (Szwajcaria) opracowano oryginalny rozproszony system monitorowania czujek dymu/gazu w kopalniach. Habilitant Kierował tym tematem, wykonanym w zespole z Henrykiem Dobrowolskim i Arturem Krystosikiem. Praca zakończyła się wdrożeniem u zamawiającego.

**Wkład habilitanta:** projekt systemu (20%), kierowanie zespołem projektowym i wykonawczym

### 6.4. Współpraca z DeTeMobil (obecnie T-Mobile) – badania socjologiczne i protokoły komunikacyjne w sieci GSM

W latach 1995-1998 prowadzono wspólne prace z DeTeMobil GmbH (oraz później z Polską Telefonią Cyfrową, w której udziały pozyskała firma DeTeMobil, obecnie T-Mobile), w ramach dwóch kolejnych kierowanych przeze mnie umów dwustronnych. Prace dotyczyły instalacji eksperymentalnej sieci GSM w regionie o słabym zaludnieniu (Suwalskie). Prace miały miejsce jeszcze przed budową pierwszych komercyjnych sieci GSM w Polsce i obejmowały badania socjologiczne nad zmianami społecznymi wywołanymi przez telefonię komórkową w obszarach wiejskich (nawet w Niemczech zasięg telefonii komórkowej obejmował wówczas głównie duże skupiska zamieszkania). Badania socjologiczne były prowadzone pod kierownictwem Andrzeja Stąsieka i stanowią część raportu końcowego, przekazanego parterowi niemieckiemu.

Drugą część badań stanowiło opracowanie wymagań i oryginalnych protokołów wymiany danych w sieciach GSM (transmisja danych w sieci GSM była wtedy dopiero w fazie początkowej), w szczególności organizacji obliczeń rozproszonych z wykorzystaniem sieci GSM

[Daszczuk, Mieścicki, Krystosik, et al., 1998] i protokołu monitorowania przy pomocy wiadomości SMS [Mieścicki, Flisiak, et al., 1998]. Ponieważ minimalizacja liczby komunikatów w komunikacji SMS jest kluczowa, habilitant przeprowadził badania nad możliwością usuwania niedoczasów (ang. timeout) z protokołu i dowiódł przy pomocy temporalnej analizy przestrzeni osiągalności, że można usunąć niedoczas z jednej ze stron protokołu nie tracąc jego poprawności [Daszczuk, 1998].

W ramach projektu habilitant był głównym organizatorem trzech konferencji.

**Wkład habilitanta:** kierowanie projektem i zespołem, protokoły wymiany danych w sieciach GSM (20%), weryfikacja formalna protokołu (100%)

**Wkład w rozwój:** protokół monitorowania przy pomocy wiadomości SMS (20%), weryfikacja formalna protokołu (100%)

**Prace te zostały podsumowane w raportach badawczych IIPW:**

- [Daszczuk, Mieścicki, Krystosik, et al., 1998] – raport badawczy IIPW: organizacja systemu przetwarzania z siecią wymiany danych opartą na sieci GSM (20%),
- [Mieścicki, Flisiak, et al., 1998] – raport badawczy IIPW: specyfikacja i weryfikacja protokołu monitorowania opartego o wiadomości SMS (20%),
- [Daszczuk, 1998] – raport badawczy IIPW: weryfikacja temporalna protokołu wymiany danych w sieci GSM (100%)

## 6.5. Eksploracja danych – umowa z Polską Telefonią Cyfrową

W latach 1998-2001 realizowano dwie umowy badawcze we współpracy z PTC, pierwszą kierował Jerzy Mieścicki a drugą Mieczysław Muraszkiwicz. Obie umowy dotyczyły eksploracji danych w zasobach baz danych sieci ERA GSM (obecnie T-Mobile Polska). W pracach tych habilitant zajmował się organizacją pozyskania danych: identyfikacja procesów biznesowych, baz danych, przepływów danych pomiędzy komórkami organizacyjnymi i bazami danych, sposobach podejmowania decyzji i wsparciu tych decyzji przez przechowywane i pozyskiwane dane oraz obliczenia na nich. Zespół badawczy dokonał eksploracji danych (jak dane z systemu sygnalizacji SS7 czy też statystyki obciążenia elementów sieci) wieloma metodami, w tym metodami opracowanymi specjalnie na potrzeby realizacji prac w ramach wspomnianych umów [Daszczuk, Gawrysiak, et al., 2000]. Część wyników została wykorzystana przez PTC w sposób biznesowy.

W ramach tych prac prowadzono zarządzanie projektem poprzez stronę internetową (ang. web-based project management), którą to stronę prowadził habilitant.

**Wkład habilitanta:** organizacja pozyskania danych (50%)

**Istotna publikacja:** [Daszczuk, Gawrysiak, et al., 2000] – referat na międzynarodowej konferencji SCI/ISAS, Orlando, USA: opis procesu pozyskania danych (50%) i wnioskowania przy pomocy algorytmów eksploracji danych (10%)

## 6.6. Monitorowanie obiektów przemysłowych poprzez sieć zarządzania

W latach 1994-2000 habilitant kierował opracowaniem i rozbudową systemu SCADA (Supervisory Control And Data Acquisition – wówczas nazwa ta nie była jeszcze szeroko stosowana, przynajmniej w Polsce) dla Elektrociepłowni Siekierki, o nazwie ESS. Zespół pod kierownictwem habilitanta (od strony organizacyjnej tematem kierował Janusz Lewandowski z wydziału MEL PW) wykonał system monitorowania, przenosząc i integrując dane z systemów Westinghouse, Siemens i innych do sieci zarządzania [Daszczuk et al., 1995]. Prace te były wykonane w ramach 4 umów z Elektrociepłownią Siekierki i jedną umową z Elektrociepłownią Żerań oraz były wsparte grantem rektorskim [Lewandowski, 1994]. System został wyróżniony nagrodą Siemens za prace badawczo-wdrożeniowe w roku 1996.

W kolejnych latach system był pod kierownictwem habilitanta rozbudowywany. Wykonano podsystemy monitorowania: powiadamianie przez sieć zarządzania, alarmowanie przez wiadomości SMS, planowanie i rozliczenie produkcji energii elektrycznej, bilans mocy. Część z tych prac było prowadzonych w ramach programu międzywydziałowego PATIA, również we współpracy z Janem Kościelnym z Wydziału Mechatroniki [Kościelny, 1999]. Wnioski z prac nad systemem ESS były publikowane w [Mieścicki, Daszczuk, Grabski, et al., 1998] [Grabski et al., 1998] [Grabski et al., 1999].

System ESS (później również pod nazwami SIG, PowerWatch) został wdrożony w Elektrociepłowni Siekierki, później został przekazany do dystrybucji zewnętrznej firmie i był wdrożony w elektrowniach i elektrociepłowniach: Połaniec, Żarnowiec, Kozienice, Żerań.

System ESS, podobnie jak system KURMAN, składał się z wielu współpracujących komputerów wykonawczych, monitorujących, archiwizujących i interfejsowych. Przeprowadzono badanie temporalne systemu korzystając modelowania w automatach CSM i weryfikacji modelowej w systemie COSMA. Wykryto możliwe zakleszczenie w protokole nawiązywania/rozłączania połączenia.

**Wkład habilitanta:** kierowanie zespołem projektowym i wykonawczym, koncepcja monitorowania obiektów przemysłowych poprzez sieć zarządzania (30%), koncepcja alarmowania i powiadamiania przez wiadomości SMS (30%), badanie poprawności protokołów (30%)

**Wkład w rozwój:** koncepcja monitorowania obiektów przemysłowych poprzez sieć zarządzania (30%), koncepcja alarmowania i powiadamiania przez wiadomości SMS (30%), badanie poprawności protokołów (30%)

### Istotne publikacje:

- [Daszczuk et al., 1995] – referat na konferencji Problemy Badawcze Energetyki Ciepłej: budowa systemu monitorowania w oparciu o sprzężenie sieci przemysłowej z siecią zarządzania (30%),

- [Mieścicki, Daszczuk, Grabski, et al., 1998] – referat na konferencji Sieci Komputerowe'98, Gliwice, Polska: integracja systemu sterowania z siecią zarządzania, **MNiSW:4** (2008, z 1998 niedostępny) (25%),
- [Grabski et al., 1998] – referat na konferencji krajowej Sieci Komputerowe'98, Gliwice, Polska: praktyczne podejście do projektowania heterogenicznego systemu rozproszonego, **MNiSW:4** (2008, z 1998 niedostępny) (25%),
- [Grabski et al., 1999] – raport badawczy IIPW: weryfikacja temporalna protokołu monitorowania (25%)

**Prace zostały wyróżnione nagrodą zespołową Siemens za prace badawczo-wdrożeniowe w roku 1996**

### 6.7. Roboty mobilne

W ramach Laboratorium Systemów Rozproszonych habilitant brał udział w konstrukcji mobilnych robotów zbudowanych na planie koła, zwanych „okrągłakami”. W ramach prac ok. 10 studentów zostały zbudowane fizyczne roboty, jak również platforma do ich zdalnego programowania i symulatory zachowań grupy robotów. Powstało ok. 10 prac dyplomowych, w tym 4 pod kierownictwem habilitanta [Zawistowski, 1996] [Modzelewski, 1998] [Zabrowski and Pabiś, 2001] [Burak, 2003].

### 6.8. Prace w przedsiębiorstwie MERA-SYSTEM

Przed zatrudnieniem w Politechnice Warszawskiej, w latach 1982-1990, habilitant prowadził prace nad obiektowymi systemami operacyjnymi: SOM-5 dla MERY-400 [Chrobot, 1980] i MOST dla wielomikroprocesorowego komputera MS-86, który powstawał w przedsiębiorstwie MERA-SYSTEM. Konstrukcja MS-86 zastała ukończona na etapie modelu. Prace nie były kontynuowane z powodu pojawienia się na rynku olbrzymiej liczby komputerów klasy PC importowanych z Dalekiego Wschodu.

Habilitant opracował również protokół dla pierścieniowej sieci mikrokomputerów pracujących w kasach sklepowych, co wówczas było jednym z nielicznych tego typu rozwiązań [Daszczuk, 1988]. W tym czasie habilitant był szczególnie zainteresowany semantycznymi powiązaniem pomiędzy systemami operacyjnymi poszczególnych jednostek wielokomputera, jednocześnie na wielu poziomach, oraz narzędziami wspomagającymi konstrukcję wielokomputera (uruchamianie modułów przy pomocy analizy sygnatur, rozproszony debugger do uruchamiania obiektów systemu operacyjnego, testowanie niezmienników systemu operacyjnego przez proces tracenia czasu). Prace te były podsumowane publikacjami [Daszczuk, 1987] [Daszczuk and Gondzio, 1987a] [Daszczuk and Gondzio, 1987b] [Daszczuk, 1987] [Daszczuk, 1991].

**Wkład habilitanta:** elementy systemu operacyjnego SOM-5 (10%), koncepcja systemu operacyjnego MOST (30%), koncepcja semantycznych powiązań na wielu poziomach pomiędzy systemami operacyjnymi wielokomputera (100%), pierścieniowa sieć mikrokomputerów (70%), kierowanie zespołem projektowym i wykonawczym

**Wkład w rozwój:** semantyczne powiązania na wielu poziomach pomiędzy systemami operacyjnymi wielokomputera (100%), testowanie niezmienników systemu operacyjnego przez proces tracenia czasu (100%), pierścieniowa sieć mikrokomputerów (70%)

**Istotne publikacje:**

- [Daszczuk, 1991] – artykuł w The Computer Journal (Oxford University Press): zależności semantyczne w systemie operacyjnym dla komputera rozproszonego, **JCR, IF: 1.000**(2008, z 1991 niedostępny), **MNiSW:15** (2008, z 1991 niedostępny) (100%),
- [Daszczuk, 1987] – artykuł w Microprocessors and Microsystems: koncepcja monitorowania niezmienników systemu operacyjnego przez proces tracenia czasu, **JCR, IF: 0.652**(2008, z 1987 niedostępny), **MNiSW:15** (2008, z 1991 niedostępny) (100%),
- [Daszczuk and Gondzio, 1987a] – referat na konferencji SMMA, Budapeszt, Węgry: koncepcja debuggera dla systemu wielomikroprocesorowego (50%)  
[Daszczuk and Gondzio, 1987b] – referat na konferencji ISMMS, Eisenach, NRD: koncepcja debuggera dla rozproszonego systemu operacyjnego (50%),

Na marginesie prac nad systemem operacyjnym, habilitant kierował projektem systemu obsługi magazynów wysokiego składowania STER w przedsiębiorstwach ARDOM Siedlce, Poznań i Kluczbork (był również jednym z wykonawców). Szczególną cechą tego systemu były zabezpieczenia przeciwwawaryjne oraz skuteczna rekonstrukcja poawaryjna. Po upadku i ponownym uruchomieniu systemu, podsystem rekonstrukcji poawaryjnej powodował powrót na ekrany operatorów przerwanych dokumentów, w miejscu gdzie zostało przerwane wprowadzanie. Za wdrożenie systemu STER habilitant otrzymał nagrodę Ministra Rynku Wewnętrznego w roku 1990. Podsystem zabezpieczeń przeciwwawaryjnych i rekonstrukcji poawaryjnej został wyróżniony medalem Mikrolaur na targach Komputer '90.

**Wkład habilitanta:** projekt systemu STER (80%), wykonanie systemu STER (60%), zabezpieczenia przeciwwawaryjne i rekonstrukcja poawaryjna (80%).

**Wkład w rozwój:** koncepcja zabezpieczeń przeciwwawaryjnych i rekonstrukcji poawaryjnej (80%)

**Nagroda zespołowa Ministra Rynku Wewnętrznego 1990,  
Medal targów Komputer 1990 Mikrolaur**

## 7. Projekty badawczo – rozwojowe

Cala aktywność naukowa habilitanta jest związana z realizacją projektów badawczo-rozwojowych, w większości realizowanych we współpracy z przemysłem. Projekty zostały omówione poprzedniej części autoreferatu. Najważniejsze z nich:

**Przed doktoratem:**

- 1992-1998 Identyfikacja i sterowanie w czasie rzeczywistym obiektów przemysłowych: cukrownie Malbork i Garbów, Wistom Tomaszów, Petrochemia



Płocka, Wydział Chemii, Wydział MEL – główny wykonawca, współautor metody identyfikacji i sterowania.

- 1995-2002 Monitorowanie obiektów przemysłowych poprzez sieć zarządzania: elektrownie i elektrociepłownie Siekierki, Żerań, Połaniec, Kozienice, Żarnowiec – kierownik i jeden z wykonawców.
- 1995-1998 Sterowanie rozproszonymi czujkami w kopani: Sultzer GmbH – kierownik.
- 1996-1997 Budowa eksperymentalnej sieci GSM: DeTeMobil (dziś T-Mobile) - kierownik.
- 1998-2000 Eksploracja danych: Polska Telefonii Cyfrowa – jeden z wykonawców.

#### **Po doktoracie:**

- 2004-2006 Specyfikacja i weryfikacja fragmentów systemu awioniki śmigłowca: Instytut Techniczny Wojsk Lotniczych – jeden z wykonawców.
- 2011-2015 Eko-mobilność: transport przyjazny środowisku – realizowany w ramach wspomaganego przez Unię Europejską Programu Operacyjnego Innowacyjna Gospodarka – kierownik i jeden z wykonawców symulatora Feniks i badań symulacyjnych, współautor oryginalnego algorytmu zarządzania.
- 2018 Projekt i wykonanie elementów nowoczesnego interfejsu użytkownika dla systemu zarządzania przepływem pracy: Syndatis Sp.z o.o. – główny wykonawca.
- Obecnie – udział w inicjatywnie prorektora PW prof. Rajmunda Bacewicza - udziału Politechniki Warszawskiej w programie rządowym ElektroMobilność

## **8. Praca w dydaktyce**

Przygotowanie i wygłaszanie wykładów Wstęp do Informatyki, Systemy Operacyjne I i II, Systemy Rozproszone, Distributed Computing and Systems.

Prowadzenie zajęć laboratoryjnych i projektowych w przedmiotach Język Pascal, Język C, Struktury Języków Programowania, Systemy Operacyjne I i II, Systemy Rozproszone.

Przygotowanie konspektu i wykładu oraz prowadzenie wykładu Distributed Computing and Systems w języku angielskim.

Od 15 lat habilitant prowadzi seminarium dyplomowe inżynierskie.

5 studentów jest współautorami referatów habilitanta na konferencjach, w tym 3 na międzynarodowych, indeksowanych w WoS.

## **9. Praca organizacyjna w dydaktyce**

#### **Przed doktoratem:**

W latach 1996-2000 habilitant zbudował, a potem brał udział w rozbudowie Laboratorium Systemów Rozproszonych, w ramach 5 grantów dziekańskich (trzy z nich kierowane przez habilitanta) i korzystając z zasobów Instytutu Informatyki. Laboratorium to służyło do wykonywania prac dyplomowych związanych z:

- specyfikacją i weryfikacją systemów rozproszonych, konstrukcją systemu COSMA,
- budową stanowisk laboratoryjnych w ramach prac nad systemem KURMAN,
- pracami nad protokołami transmisji danych w sieci GSM,
- budową mobilnych robotów „okrągłaków”.

W laboratorium powstało ok. 20 prac dyplomowych

Habilitant brał udział w pracach zespołu Instytutu Informatyki ds. programu studiów na kierunku Informatyka, zespołu Instytutu Informatyki ds. programu studiów dwustopniowych.

Przez dwa lata habilitant był opiekunem I roku studiów na kierunku Informatyka.

#### **Po doktoracie:**

Habilitant opracował zestaw ćwiczeń laboratoryjnych do przedmiotu Systemy Operacyjne i ćwiczeń projektowych z przedmiotów Systemy Rozproszone i Distributed Computing and Systems.

Habilitant wykonał kilka programów służących do wykonywania ćwiczeń studenckich w ramach przedmiotów Wstęp do Informatyki i Systemy Operacyjne:

- uniwersalny parser,
- program do obliczeń algorytmem genetycznym,
- maszyna Turinga,
- weryfikator temporalny.

Habilitant szkolił się na studiach podyplomowych z ochrony własności intelektualnej na Uniwersytecie Warszawskim, które ukończył z wynikiem bardzo dobrym. Co prawda uruchomienie zaawansowanego przedmiotu „Prawo Autorskie w Informatyce” nie doszło do skutku (zmiana koncepcji w dziekanacie dydaktycznym), ale habilitant organizował szkolenia dla pracowników Instytutu Informatyki z prawa autorskiego oraz wielokrotnie udzielał konsultacji pracownikom Wydziału w tych sprawach.

Habilitant miał istotny wkład w opracowanie programu studiów kierunku Informatyka, w tym:

- projekt „Innowacyjne polsko- i anglojęzyczne programy studiów drugiego stopnia w zakresie informatyki” w ramach którego habilitant przygotował treść i materiały do przedmiotu Distributed Computing and Systems.
- programu NERW, w którym habilitant brał udział w opracowaniu programu studiów dla nowej wersji kierunku informatyka, w szczególności opracowywał program przedmiotów związanych z algorytmiką.

**W roku 2018 habilitant otrzymał Medal Komisji Edukacji Narodowej.**

## **10. Współpraca międzynarodowa**

Habilitant w ramach swoich badań współpracuje z Bogdanem Czejdo i Sambitem Bhattacharya (Uniwersytet Stanowy Północnej Karoliny w Fayetteville), temat: specyfikacja i weryfikacja

systemów rozproszonych, logika systemów automatycznego prowadzenia pojazdów, oraz z Włodzimierzem Zuberkiem (St. Johns Memorial University of Newfoundland) nad specyfikacją i weryfikacją systemów rozproszonych, w tym nad sieciami Petriego.

Przez kilka lat trwała współpraca z DeTeMobil (obecnie T-Mobile) nad protokołami transmisji danych w sieciach GSM, oraz nad monitorowaniem obiektów poprzez komunikaty SMS.

## 11. Referencje

- Alur R, and Dill D L. (1994). A theory of timed automata. *Theoretical Computer Science*, 126(2), 183–235. doi: 10.1016/0304-3975(94)90010-8
- Andréasson I. (2003). Reallocation of Empty PRT vehicles en route. In *TRB Annual Meeting, Washington DC, 12-16 Jan. 2003* (pp. 1–13). Transportation Research Board.
- Baier C, and Katoen J-P. (2008). *Principles of Model Checking*. Cambridge, MA: MIT Press.
- Balan M S. (2009). Serializing the Parallelism in Parallel Communicating Pushdown Automata Systems. *Electronic Proceedings in Theoretical Computer Science*, 3, 59–68. doi: 10.4204/EPTCS.3.5
- Behrmann G, David A, ... Yi W. (2011). Developing UPPAAL over 15 years. *Software: Practice and Experience*, 41(2), 133–142. doi: 10.1002/spe.1006
- Bollig B, and Leucker M. (2004). Message-Passing Automata Are Expressively Equivalent to EMSO Logic. In *15th International Conference CONCUR 2004 - Concurrency Theory, London, UK, 31 Aug. - 3 Sept. 2004* (pp. 146–160). Berlin Heidelberg: Springer. doi: 10.1007/978-3-540-28644-8\_10
- Burak M. (2003). *Współpraca mobilnych agentów w systemie wieloagentowym*, MSc Thesis. Warsaw University of Technology.
- Carnegie J A, Voorhees A M, and Hoffman P S. (2007). *Viability of Personal Rapid Transit In New Jersey*. url: <https://www.nj.gov/transportation/refdata/research/reports/NJ-2007-001.pdf>
- Choromański W. (2011). Eco-Mobility project, co-funded by the European Regional Development Fund within the framework of Innovative Economy Operational Programme (WND-POIG.01.03.01-14-154/09), 2011-2015.
- Choromański W, **Daszczuk W B**, Dyduch J, Maciejewski M, Brach P, Grabski W. (2013). PRT (Personal Rapid Transit) Network Simulation. In J. Victor (Ed.), *WCTR - World Conf. on Transport Research 2013, Rio de Janeiro, Brasil, 7–10 July 2013* (pp. 1–18). Rio de Janeiro. url: <http://www.wctrs-society.com/wp/wp-content/uploads/abstracts/rio/selected/3070.pdf>
- Choromański W, **Daszczuk W B**, Grabski W, Dyduch J, Maciejewski M, Brach P. (2013). Personal Rapid Transit (PRT) Computer Network Simulation and Analysis of Flow Capacity. In W. H. Leder & W. J. Sproule (Eds.), *Automated People Movers and Transit Systems 2013, Phoenix, AZ, 21–24 April 2013* (pp. 296–312). Reston, VA: American Society of Civil Engineers. doi: 10.1061/9780784412862.022
- Chrobot S. (1980). Operating system SOM-5 for the Mera- 400 minicomputer. *Informatyka*, 12,

15–18.

- Chrobot S, and **Daszczuk W B.** (2006). Communication Dualism in Distributed Systems with Petri Net Interpretation. *Theoretical and Applied Informatics*, 18(4), 261–278. url: <https://taai.iitis.pl/taai/article/view/250/taai-vol.18-no.4-pp.261>
- Cimatti A, Clarke E, ... Tacchella A. (2002). NuSMV 2: An OpenSource Tool for Symbolic Model Checking. In E. Brinksma & K. G. Larsen (Eds.), *CAV 2002: Computer Aided Verification, Copenhagen, Denmark, 27–31 July 2002, LNCS 2404* (pp. 359–364). Berlin Heidelberg: Springer. doi: 10.1007/3-540-45657-0\_29
- Clarke E M, Grumberg O, and Peled D. (1999). *Model Checking*. Cambridge, MA: MIT Press.
- Czejdo B, and **Daszczuk W B.** (2019). Framework to Verify Distributed IoT Solutions for Traffic Analysis in ATN Stations. In *14th International Conference DepCoS-RELCOMEX'19, Brunów, Poland, 1-5 July 2019* (p. in Print). Cham, Switzerland: Springer International Publishing.
- Czejdo B, **Daszczuk W B**, and Mieścicki J. (1998). Concurrent Software Design Based on Constraints on State Diagrams. In *Third World Conference on Integrated Design & Process Technology, Berlin, Germany, 5-9 July 1998*. Berlin: Society for Design and Process Sciences.
- Daszczuk W B.** (1987). Invariant testing technique for debugging a structured operating system. *Microprocessors and Microsystems*, 11(4), 205–208. doi: 10.1016/0141-9331(87)90339-5
- Daszczuk W B.** (1988). Pierścieniowa sieć mikrokomputerów. In *Sieci Komputerowe, Wrocław, 1988*.
- Daszczuk W B.** (1991). A Structured Semantic Design of Distributed Operating Systems. *The Computer Journal*, 34(6), 482–492. doi: 10.1093/comjnl/34.6.482
- Daszczuk W B.** (1998). *Verification of Design Decisions in Communication Protocols by Evaluation of Temporal Logic Formulas, ICS WUT Reserach Report No 22/98*.
- Daszczuk W B.** (2000a). *Evaluation of Temporal Formulas Based on 'Checking in Spheres' Algorithm, Implemented using ROBDD Representation, ICS WUT Research Report No 9/2000*.
- Daszczuk W B.** (2000b). *State space reduction for reachability graph of CSM automata, ICS WUT Research Report No 10/2000*.
- Daszczuk W B.** (2001). Evaluation of temporal formulas based on “Checking By Spheres.” In *Proceedings Euromicro Symposium on Digital Systems Design, Warsaw, Poland, 4-6 Sept. 2001* (pp. 158–164). IEEE. doi: 10.1109/DSD.2001.952267
- Daszczuk W B.** (2002a). *Critical trees: counterexamples in model checking of CSM systems using CBS algorithm, ICS WUT Research Report No.8/2002*.
- Daszczuk W B.** (2002b). *Off-line and On-line versions of RSCM Algorithm for Reduction of CSM State Space in Model Checking using QsCTL Logic, ICS WUT Research Report No.15/2002*.
- Daszczuk W B.** (2002c). *Proof of correctness of CBS (Checking By Spheres) algorithm for*

*evaluation of temporal formulas in QsCTL logic, ICS WUT Research Report No 13/2002.*

- Daszczuk W B.** (2002d). *Symbolic Version (using BDD and SAT) of CBS (Checking By Spheres) Algorithm for Evaluation of Temporal Formulas in QsCTL Logic, ICS WUT Research Report No.14/2002.*
- Daszczuk W B.** (2003a). *Temporal model checking in the COSMA environment (the operation of TempoRG program), ICS WUT Research Report No 7/2003.*
- Daszczuk W B.** (2003b). *Verification of Temporal Properties in Concurrent Systems.* Warsaw University of Technology. url: <https://repo.pw.edu.pl/docstore/download/WEiTI-0b7425b5-2375-417b-b0fa-b1f61aed0623/Daszczuk.pdf>
- Daszczuk W B.** (2004). *Macrogeneration and Automata Libraries for COSMA design environment, ICS WUT Research Report No 2/2004.*
- Daszczuk W B.** (2007). Timed concurrent state machines. *Computer Science, AGH University of Science and Technology, 8, 23–36.* doi: 10.7494/csci.2007.8.3.23
- Daszczuk W B.** (2016). Discrete Event Simulation of Personal Rapid Transit (PRT) Systems. *Autobusy-TEST, 17(3), 1302–1310.* url: [http://www.autobusy-test.com.pl/images/stories/Do\\_pobrania/2016/nr\\_6/logistyka/11\\_I\\_daszczuk.pdf](http://www.autobusy-test.com.pl/images/stories/Do_pobrania/2016/nr_6/logistyka/11_I_daszczuk.pdf)
- Daszczuk W B,** and Belz A. (2018). Energy Constraints in Operation of Autonomous Transit Networks. In *2018 Baltic Geodetic Congress (BGC Geomatics), Olsztyn, Poland, 21-23 June 2018* (pp. 174–180). IEEE. doi: 10.1109/BGC-Geomatics.2018.00039
- Daszczuk W B,** Gawrysiak P, Gerszberg T, Kryszkiewicz M, Mieścicki J, Muraszkiewicz M, Okoniewski M, Rybiński H, Traczyk T, Walczak Z. (2000). Data Mining for Technical Operation of Telecommunications Companies: a Case Study. In J. K. Lee (Ed.), *World Multiconference on Systemics, Cybernetics and Informatics: SCI/ISAS 2000, Orlando, FL, 23-26 July 2000* (pp. 64–69). Orlando, FL: The Institute of Orlando.
- Daszczuk W B,** and Gondzio M. (1987a). Debugger Family in Distributed Environment. In *5th Symposium on Microcomputer and Microprocessor Applications, Budapest, Hungary, 29 Sept. - 1 Oct. 1987.* OMMIKK-Technoinform.
- Daszczuk W B,** and Gondzio M. (1987b). Debugging Aids for Development of Distributed Multi-Microprocessor Computer. In *2nd International Symposium on Multi-microprocessor Systems, Eisenach, Germany.* Institut für Informatik und Rechentechnik.
- Daszczuk W B,** Grabski W, Mieścicki J, Wytrębowski J. (2001). System modeling in the COSMA environment. In *Proceedings Euromicro Symposium on Digital Systems Design, Warsaw, Poland, 4-6 Sept. 2001* (pp. 152–157). Warsaw, Poland: IEEE Comput. Soc. doi: 10.1109/DSD.2001.952264
- Daszczuk W B,** Kurman K J, and Mieścicki J. (1994). *Pomiary identyfikacyjne wyparki w Cukrowni Garbów, ICS WUT Research Report No 1/94.*
- Daszczuk W B,** and Mieścicki J. (2015). Distributed management of Personal Rapid Transit (PRT) vehicles under unusual transport conditions. *Logistyka, 4, 2896–2901.*
- Daszczuk W B,** Mieścicki J, Kochan A, Zajac K, Skurzak K, Duplicki K. (1998). *A System for Identification of Fluidized Bed Furnace at the Institute of Heat Engineering, WUT (In*

*Polish: System identyfikacji paleniska fluidalnego w Instytucie Techniki Ciepłej PW), ICS WUT Research Report No 23/98.*

- Daszczuk W B**, Mieścicki J, Krystosik A, Dobrowolski H. (1998). *The organization of Mobile Computing Systems using the GSM network as the communication subsystem, ICS WUT Technical Report No.15/1998.*
- Daszczuk W B**, Mieścicki J, Lewandowski J, Świrski K. (1995). System for the Integration of the Block Digital Control System with the Company Network (in Polish: System Integracji Blokowego Cyfrowego Systemu Kontroli i Sterowania z Siecią Zakładową). In *Problemy Badawcze Energetyki Ciepłej, Warszawa, 5-8 Grudnia 1995* (p. tom 1, 101-107). Warszawa: Oficyna Wydawnicza Politechniki Warszawskiej.
- Daszczuk W B**, Mieścicki J, Nowacki M, Wytrębowski J. (2000). *Traffic Light Controller specified and verified using the COSMA environment: A case study, ICS WUT Research Report No.12/2000.*
- Daszczuk W B**, Mieścicki J, Nowacki M, Wytrębowski J. (2001). System Level Specification and Verification Using Concurrent State Machines and COSMA Environment. In *8th International Conference on Mixed Design of Integrated Circuits and Systems, MIXDES'01, Zakopane, Poland, 21-23 June 2001* (pp. 525–532).
- Dick G, and Yao X. (2014). Model representation and cooperative coevolution for finite-state machine evolution. In *2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6-11 July 2014* (pp. 2700–2707). New York, NY: IEEE. doi: 10.1109/CEC.2014.6900622
- Dijkstra E W. (1971). Hierarchical ordering of sequential processes. *Acta Informatica*, 1(2), 115–138. doi: 10.1007/BF00289519
- Francesca G, Santone A, ... Villani M L. (2011). Ant Colony Optimization for Deadlock Detection in Concurrent Systems. In *2011 IEEE 35th Annual Computer Software and Applications Conference, Munich, Germany, 18-22 July 2011* (pp. 108–117). IEEE. doi: 10.1109/COMPSAC.2011.22
- Gałecki T, and **Daszczuk W B**. (2019). Shapp: workload management system for massive distributed calculations. In *8th Computer Science On-line Conference, Zlin, Czech Republic, 24-27 April 2019* (accepted). Cham, Switzerland: Springer International Publishing.
- Glabbeek R J, and Goltz U. (1990). Equivalences and refinement. In *LITP Spring School on Theoretical Computer Science La Roche Posay, France, 23–27 April 1990, LNCS 469* (pp. 309–333). Berlin Heidelberg: Springer-Verlag. doi: 10.1007/3-540-53479-2\_13
- Godefroid P, and Khurshid S. (2002). Exploring Very Large State Spaces Using Genetic Algorithms. In J.-P. Katoen & P. Stevens (Eds.), *International Conference on Tools and Algorithms for the Construction and Analysis of Systems -TACAS 2002, Grenoble, France, 8–12 April 2002, LNCS vol. 2280* (pp. 266–280). Berlin Heidelberg: Springer. doi: 10.1007/3-540-46002-0\_19
- Grabski W, and **Daszczuk W B**. (2017a). A study on cooperation of urban transport means: PRT and light rail. *Transport Problems*, 12(4), 5–14. doi: 10.20858/tp.2017.12.4.1
- Grabski W, and **Daszczuk W B**. (2017b). Priority Rules on ATN (PRT) Intersections. *Autobusy-*

TEST, 18(12), 1503–1506. url: [http://www.autobusy-test.com.pl/images/stories/Do\\_pobrania/2017/nr12/Logistyka/283\\_074\\_A\\_L\\_GRABSKI\\_DASZCZUK.pdf](http://www.autobusy-test.com.pl/images/stories/Do_pobrania/2017/nr12/Logistyka/283_074_A_L_GRABSKI_DASZCZUK.pdf)

Grabski W, **Daszczuk W B**, ... Mieścicki J. (1998). Integration of Industrial DCS with a Plant Management Network. *Zeszyty Naukowe. Informatyka/Politechnika Śląska*, 34, 569–575.

Grabski W, **Daszczuk W B**, ... Dobrowolski H. (1999). *Verification of the event protocol for establishing and disconnecting a connection in the ESS system (in Polish: Weryfikacja zdarzeniowego protokołu nawiązywania i rozłączania połączenia w systemie ESS)*, ICS WUT Research Report No.10/1999.

Heiner M, Schwarick M, and Wegener J-T. (2015). Charlie – An Extensible Petri Net Analysis Tool. In *36th International Conference, PETRI NETS 2015, Brussels, Belgium, 21-26 June 2015* (pp. 200–211). Cham, Switzerland: Springer International Publishing. doi: 10.1007/978-3-319-19488-2\_10

Hoare C A R. (1978). Communicating sequential processes. *Communications of the ACM*, 21(8), 666–677. doi: 10.1145/359576.359585

Holzmann G J. (1995). Tutorial: Proving properties of concurrent systems with SPIN. In *6th International Conference on Concurrency Theory, CONCUR'95, Philadelphia, PA, 21–24 Aug. 1995* (pp. 453–455). Berlin Heidelberg: Springer-Verlag. doi: 10.1007/3-540-60218-6\_34

Jia W, and Zhou W. (2005). *Distributed Network Systems. From Concepts to Implementations*. New York: NETA vol. 15, Springer. doi: 10.1007/b102545

Johnsen E B, Blanchette J C, ... Owe O. (2009). Intra-Object versus Inter-Object: Concurrency and Reasoning in Creol. *Electronic Notes in Theoretical Computer Science*, 243, 89–103. doi: 10.1016/j.entcs.2009.07.007

Katoen J-P. (1999). *Concepts, algorithms, and tools for model checking*. Erlangen-Nürnberg, Germany.

Kessler C, and Keller J. (2007). Models for Parallel Computing: Review and Perspectives. In *PARS-Mitteilungen, 13-29 Dec. 2007* (pp. 13–29). url: <http://www.ida.liu.se/~chrke/modelsurvey.pdf>

Kościelny J. (1999). Integracja Systemu Nadzoru Bloków Energetycznych ESS z Systemem Diagnostyki Procesów Przemysłowych DIAG, grant PATIA IAIR/83/1411/99. Program PATIA, Warsaw University of Technology.

Kurman K J, **Daszczuk W B**, and Mieścicki J. (1992). *Projektowanie jednowymiarowych układów automatycznej regulacji przy użyciu systemu KURMAN*, ICS WUT Research Report No 14/92.

Lauer H C, and Needham R M. (1979). On the duality of operating system structures. *ACM SIGOPS Operating Systems Review*, 13(2), 3–19. doi: 10.1145/850657.850658

Lees-Miller J D, and Wilson R E. (2012). Proactive empty vehicle redistribution for personal rapid transit and taxis. *Transportation Planning and Technology*, 35(1), 17–30. doi: 10.1080/03081060.2012.635414

- Lewandowski J. (1994). Opracowanie Oprogramowania do Celów Tworzenia i Wdrażania Kompleksowych, Rozproszonych Systemów Sterowania i Zarządzania W Zakładach Energetycznych, grant rektorski 2. 273/091/94. Institute of Heat Engineering, Warsaw University of Technology.
- Lewerentz C, and Lindner T (Eds.). (1995). *Formal Development of Reactive Systems, LNCS 891*. Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/3-540-58867-1
- Lorenc J, **Daszczuk W B**, and Grabski W. (2019). Data Model Design in Automatic Transit System (PRT) Simulation Software. In R. Silhavy (Ed.), *7th Computer Science On-line Conference, Zlin, Czech Republic, 25-28 April 2018, AISC Vol. 673* (pp. 121–131). Cham, Switzerland: Springer International Publishing. doi: 10.1007/978-3-319-91186-1\_14
- Manna Z, and Pnueli A. (1992). *The Temporal Logic of Reactive and Concurrent Systems*. New York, NY: Springer. doi: 10.1007/978-1-4612-0931-7
- May D. (1983). OCCAM. *ACM SIGPLAN Notices*, 18(4), 69–79. doi: 10.1145/948176.948183
- Mieścicki J. (1990). Wspomaganie Eksperymentem Projektowanie Wielowymiarowych Układów Automatycznej Regulacji, grant MEN T/02/062/90-2, 1990-1994.
- Mieścicki J. (1991). Opracowanie metody modelowania zachowań systemów współbieżnych, grant KBN 8 0568 91 01, 1991-1994. Warsaw University of Technology, Institute of Computer Science.
- Mieścicki J. (1994a). Boolean formulas and families of sets. *Bulletin of the Polish Academy of Sciences. Technical Sciences*, 42(1), 111–123.
- Mieścicki J. (1994b). Identyfikacja i Podnoszenie Jakości Regulacji w Instalacjach Chemicznych, grant rektorski 1.503/032/653/1, 1994-1995.
- Mieścicki J. (1995). Metodologia i Narzędzia Case do Tworzenia i Modyfikacji Współbieżnego Oprogramowania do Celów Projektowania Urządzeń Elektronicznych, grant KBN 8 T11C 007 08, 1995-1997. Warsaw University of Technology, Institute of Computer Science.
- Mieścicki J. (2001). Wspomagane Komputerowo Projektowanie Systemów Reaktywnych z Użyciem Metod Formalnych, grant KBN 7 T11C 013 20, 2001-2003. Warsaw University of Technology, Institute of Computer Science.
- Mieścicki J, Baszun M, **Daszczuk W B**, Czejdo B. (1996). Verification of Concurrent Engineering Software Using CSM Models. In *2nd World Conf. on Integrated Design and Process Technology, Austin, TX, 1-4 Dec. 1996* (pp. 322–330). Dallas, TX: SDPS.
- Mieścicki J, Czejdo B, and **Daszczuk W B**. (2004a). Model Checking In The COSMA Environment as a Support For The Design of Pipelined Processing. In *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2004, Jyväskylä, Finland, 24-28 July 2004*. Jyväskylä, Finland.
- Mieścicki J, Czejdo B, and **Daszczuk W B**. (2004b). Multi-phase model checking of a three-stage pipeline using the COSMA tool. In *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2004, Jyväskylä, Finland, 24-28 July 2004*. Jyväskylä, Finland.
- Mieścicki J, and **Daszczuk W B**. (2005). Formal verification of a three-stage pipeline in the



- COSMA environment. In M. Ganzha, M. Paprzycki, J. Wachowicz, & K. Węcel (Eds.), *Conference: XXI Fall Meeting of Polish Information Processing Society, Wisła, Poland, 5-9 Dec. 2005* (pp. 195–203). url: <http://www.proceedings2005.imcsit.org/docs/41.pdf>
- Mieścicki J, and **Daszczuk W B.** (2006). Behavioral and real-time verification of a pipeline in the COSMA environment. *Annales UMCS, Informatica Vol. AI, 4AI*, 254–265. doi: 10.17951/ai.2006.4.1.254-265
- Mieścicki J, and **Daszczuk W B.** (2013). Proposed benchmarks for PRT networks simulation. *Archives of Transport*, 27–28(3–4), 123–133.
- Mieścicki J, **Daszczuk W B**, Grabski W, Krystosik A. (1998). Practical approach to distributed systems' design. *Zeszyty Naukowe. Informatyka/Politechnika Śląska*, 34, 205–213.
- Mieścicki J, **Daszczuk W B**, Kurman K J, Byzia T, Smażyński P, Adamiec R. (1992). *KURMAN: The system supporting the design of multidimensional automated control systems (in Polish KURMAN: System wspomagający projektowanie wielowymiarowych układów automatycznej regulacji)*, ICS WUT Research Report No 10/92.
- Mieścicki J, **Daszczuk W B**, and Nowacki M. (1998). *Badanie poprawności zachowań systemu współbieżnego na przykładzie systemu do pomiarów identyfikacyjnych*, ICS WUT Research Report No 11/98.
- Mieścicki J, Dobrowolski H, and **Daszczuk W B.** (2003). *Model checking of multi-agent system's communication protocols specified using the Z notation*, ICS WUT Research Report No.17/2003.
- Mieścicki J, Flisiak D, Czuchaj A, **Daszczuk W B.** (1998). *Specification and verification of SMS-based data stream protocol*, ICS WUT Research Report No 16/98.
- Mieścicki J, Kurman K J, and **Daszczuk W B.** (1995a). *Identyfikacja kolumny destylacyjnej D-1 w Laboratorium Procesów Technologicznych PW*, ICS WUT Research Report No 27/95.
- Mieścicki J, Kurman K J, and **Daszczuk W B.** (1995b). *MIKOZ: The method for identification and coordination of feedback interactions in the regulation of continuous technological processes (in Polish: MIKOZ: Metoda identyfikacji i koordynacji oddziaływań zwrotnych)*, ICS WUT Research Report No. 26/95.
- Mieścicki J, Kurman K J, and **Daszczuk W B.** (1998). *Identyfikacja i projekt algorytmu regulacji dla ciśnieniowego paleniska fluidalnego w Instytucie Techniki Ciepłej PW*, ICS WUT Research Report No.25/98.
- Mieścicki J, Kurman K J, Kochan A, **Daszczuk W B.** (2000). *Nowy algorytm regulacji dla desorbera w instalacji odsiarczania gazów suchych*, ICS WUT Technical Report No.1/2000.
- Miller A. (1997). *Identyfikacja i Sterowanie Paleniskiem Fluidalnym*, grant KBN 8 T10B 022 11, 1997 - 1998.
- Milner R. (1980). *A Calculus of Communicating Systems*, LNCS vol. 92. (R. Milner, Ed.) (Vol. 92). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/3-540-10235-3
- Milner R. (1983). Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25(3), 267–310. doi: 10.1016/0304-3975(83)90114-7

- Modzelewski M. (1998). *Architektura sterownika prostego robota Okrągłaka*, MSc Thesis. Warsaw University of Technology.
- Rosa N S, and Cunha P R F. (2004). A Software Architecture-Based Approach for Formalising Middleware Behaviour. *Electronic Notes in Theoretical Computer Science*, 108, 39–51. doi: 10.1016/j.entcs.2004.01.011
- Savoiu N, Shukla S K, and Gupta R K. (2002). Automated concurrency re-assignment in high level system models for efficient system-level simulation. In *Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition, Paris, France, 4-8 March 2002* (pp. 875–881). IEEE Comput. Soc. doi: 10.1109/DATE.2002.998404
- Schweizer J, Danesi A, ... Traversi E. (2012). Comparison of static vehicle flow assignment methods and microsimulations for a personal rapid transit network. *Journal of Advanced Transportation*, 46(4), 340–350. doi: 10.1002/atr.1196
- van Glabbeek R, Goltz U, and Schicke J-W. (2008). On Synchronous and Asynchronous Interaction in Distributed Systems. In *33rd International Symposium, MFCS 2008, Toruń, Poland, 25-29 Aug. 2008* (pp. 16–35). Berlin Heidelberg: Springer. doi: 10.1007/978-3-540-85238-4\_2
- Zabrowski P, and Pabiś N. (2001). *Platforma do badania systemów wieloagentowych w środowisku fizycznym*, MSc Thesis. Warsaw University of Technology.
- Zawistowski M. (1996). *Środowisko tworzenia aplikacji dla grupy symulowanych i rzeczywistych robotów*, MSc Thesis. Warsaw University of Technology.
- Zheng P, Jeffery D, and McDonald M. (2009). Development and evaluation of traffic management strategies for personal rapid transit. In *Industrial Simulation Conference 2009, Loughborough, UK, 1-3 June 2009* (pp. 191–195). url: <https://faculty.washington.edu/jbs/itrans/jeffery-mod-simu.pdf>
- Zieliński W. (2018). *Współpraca PRT z innymi środkami transportu na przykładzie symulatora Feniks*. Warsaw University of Technology. url: <https://apd.usos.pw.edu.pl/diplomas/28676/>
- Zielonka W. (1987). Notes on finite asynchronous automata. *RAIRO - Theoretical Informatics and Applications*, 21(2), 99–135. doi: 10.1051/ita/1987210200991
- Zuberek W M. (2009). Siphon-Based Verification of Component Compatibility. In W. Zamojski (Ed.), *2009 Fourth International Conference on Dependability of Computer Systems, DepCos-RELCOMEX'09, Brunów, Poland, 30 June-2 July 2009* (pp. 123–132). Piscataway, NJ: IEEE. doi: 10.1109/DepCoS-RELCOMEX.2009.35