

# Implementacja GUI dla systemu który jeszcze nie istnieje

Michał Tanaś

Uniwersytet im. A.Mickiewicza w Poznaniu, ITTI sp. z.o.o.

*mtanas@itti.com.pl*

Andrzej Adamczyk

ITTI sp. z.o.o.

*andrzej.adamczyk@itti.com.pl*

Grzegorz Taberski

ITTI sp. z.o.o.

*grzegorz.taberski@itti.com.pl*

**Abstrakt.** W artykule prezentujemy nowatorski sposób projektowania i implementacji Graficznego Interfejsu Użytkownika. W klasycznym podejściu do procesu projektowania GUI jest tworzony w oparciu o ustalony uprzednio szczegółowy projekt aplikacji bazowej, co przy uwzględnieniu faktu iż GUI jest często najbardziej rozbudowaną i pracochłonną częścią aplikacji IT powoduje opóźnienia i trudności w procesie tworzenia aplikacji. Prezentowane przez nas alternatywne podejście umożliwia tworzenie GUI wcześniej niż znany jest szczegółowy projekt systemu IT, co znacznie przyspiesza projektowanie systemu jako całości oraz umożliwia użycie samego GUI jako demonstratora działania jeszcze nie gotowego systemu. Przedstawiona metodyka projektowania GUI jest omówiona na przykładzie historii projektowania rzeczywistego systemu IT

**Informacja o autorach.** MICHAŁ TANAS – z wykształcenia informatyk, absolwent Uniwersytetu im. A. Mickiewicza w Poznaniu. Doktorat w dziedzinie informatyki na Politechnice Clausthal (RFN). Od 2000 roku projektuje, implementuje i utrzymuje systemy IT w Centrum Informatyki UAM. Od 2005 roku adiunkt w Zakładzie Informatyki Stosowanej na Wydziale Fizyki UAM. Od 2007 roku starszy konsultant w ITTI Sp. z o.o.

ANDRZEJ ADAMCZYK w latach 1991-1993 studiował na Politechnice Poznańskiej na kierunku Telekomunikacja i Elektronika. W latach 1993-1996 kontynuował studia w EFP - Francusko-Polskiej Wyższej Szkole Technik Informatyczno-Komunikacyjnych w Poznaniu. W 1995 roku odbył półroczny staż w laboratorium LAAS-CNRS w Tuluzie we Francji. Stopień magistra inżyniera uzyskał w 1996 roku w dziedzinie systemów rozproszonych. Od 1996 roku pracował na stanowisku Kierownik Pracowni Systemów Informatycznych w Instytucie Technik Telekomunikacyjnych i Informatycznych (ITTI) w Poznaniu. W Zespole Systemów Informatycznych i Multimediiów pod kierownictwem prof. dr hab. inż. Czesława Jędrzejka prowadził projekty w następujących dziedzinach: inżynieria oprogramowania, technologie inter- i intranetowe, aplikacje i systemy baz danych oraz technologia i metodyka zdalnego kształcenia multimedialnego. Koordynował projekty Piątego Programu Ramowego, programu Leonardo da Vinci i PHARE. Przez rok kierował działaniami informatycznymi firmy ALMA S.A. na stanowisku Dyrektora Departamentu Systemów i Aplikacji. Obecnie w ITTI na stanowisku starszego konsultanta realizuje projekty związane z analizą, projektowaniem i budową systemów informatycznych oraz z zagadnieniami bezpieczeństwa informacji. Pełni też w ITTI funkcję Dyrektora ds. Produkcji.

GRZEGORZ TABERSKI ukończył Informatykę Stosowaną na Wydziale Fizyki Uniwersytetu im. A. Mickiewicza w roku 2008. W latach 2004-2007 studiował Astronomię. W roku 2005 odbył miesięczny staż w Joint Institute for Nuclear Research - Laboratory of Information Technologies w Dubnej (Rosja). W 2006 odbył praktyki w Pionie Eksploatacji sieci firmy PTC. Od 2007 roku pracuje w ITTI Sp. z o.o. na stanowisku młodszego konsultanta, gdzie uczestniczy w projektach europejskich i krajowych z dziedzin na styku biznesu i technologii IT.

## 1. Wstęp<sup>1</sup>

W artykule zaprezentowany został nowatorski sposób projektowania i implementacji graficznego interfejsu użytkownika (GUI). W klasycznym podejściu do procesu projektowania, GUI jest tworzony w oparciu o ustalony uprzednio szczegółowy projekt aplikacji bazowej, co przy uwzględnieniu faktu iż GUI jest często najbardziej rozbudowaną i pracochłonną częścią aplikacji IT powoduje opóźnienia i trudności w procesie tworzenia aplikacji. Prezentowane przez nas alternatywne podejście umożliwia tworzenie GUI wcześniej niż znany jest szczegółowy projekt systemu IT, co znacznie przyspiesza projektowanie systemu jako całości oraz umożliwia użycie samego GUI jako demonstratora działania jeszcze nie gotowego systemu.

Przedstawiona metodyka projektowania GUI jest omówiona na przykładzie historii projektowania rzeczywistego systemu IT w ramach projektu badawczego zamawianego pt. „Modele zagrożeń aglomeracji miejskiej wraz z systemem zarządzania kryzysowego na przykładzie miasta stołecznego Warszawy”. Celem projektu, który wciąż trwa jest stworzenie informatycznego systemu wspomagania zarządzania kryzysowego i pilotażowe wdrożenie go w ramach Biura Bezpieczeństwa i Zarządzania Kryzysowego w Warszawie. Zatem produkt projektu ma być konkretny i namacalny.

Wkrótce po rozpoczęciu projektu okazało się, że logika systemu nie będzie gotowa w momencie prac nad GUI. Jednocześnie nie było warunków pozwalających na modyfikację harmonogramu realizacji poszczególnych zadań w sposób taki, by nie wpływał on na datę zakończenia całego przedsięwzięcia. Dlatego należało znaleźć inne rozwiązanie w zaistniałej sytuacji. To rozwiązanie opisane jest w kolejnych rozdziałach niniejszego artykułu.

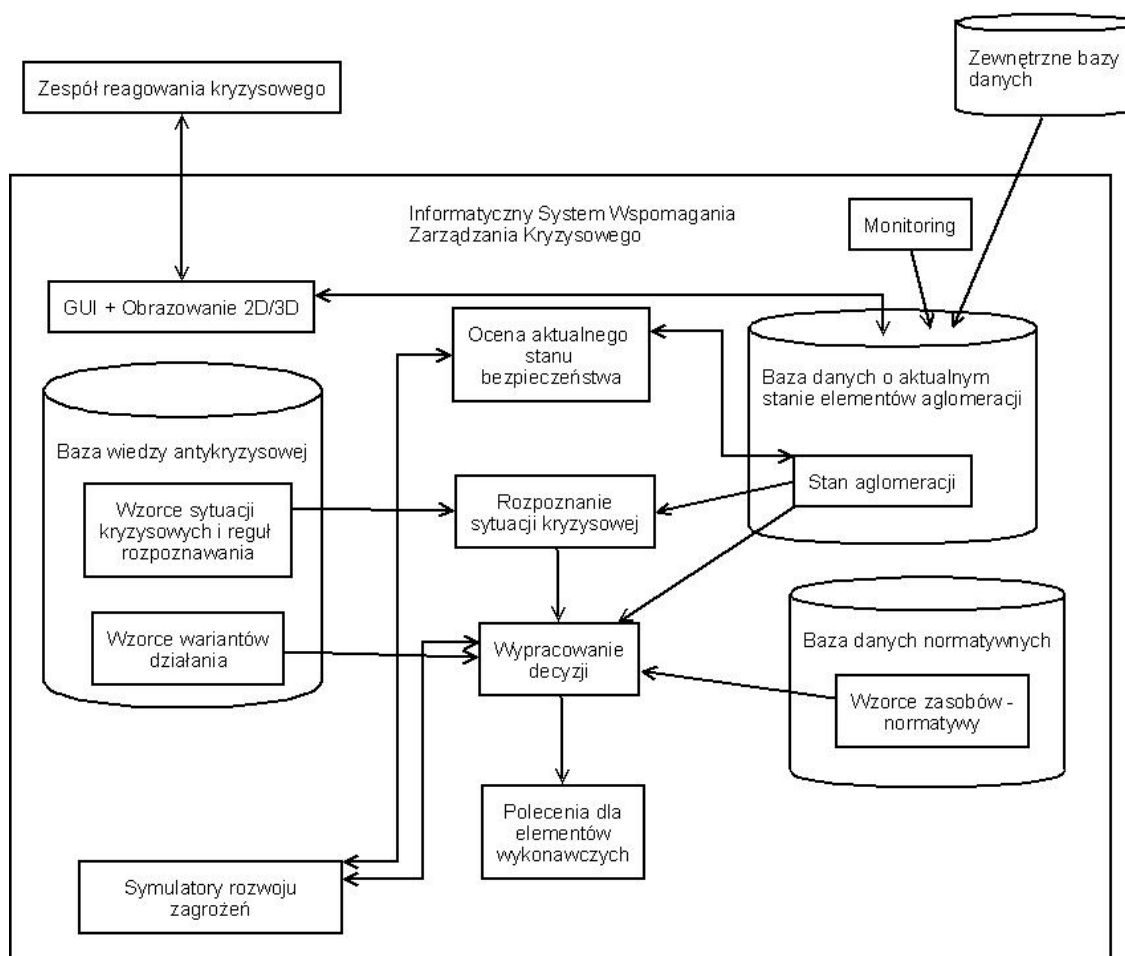
## 2. Architektura ISWZK

ISWZK w założeniu jest systemem wspomagającym całość działań służb antykryzysowych na obszarze miasta stołecznego Warszawy. W ogólnym zarysie ISWZK składa się z:

- części dyspozytorskiej wspomagającej zarządzanie zasobami antykryzysowymi w czasie akcji
- części symulacyjnej umożliwiającej przygotowanie planów antykryzysowych oraz zapewniającej wspomaganie decyzji poprzez symulacje „co by było gdyby”
- Baz danych zawierających informacje o infrastrukturze, zasobach antykryzysowych, itp.

---

<sup>1</sup> Współfinansowanie przez NCBiR w ramach projektu badawczego zamawianego nr PBZ-MIN-011/013/2004



Rys. 1. Idea Informatycznego Systemu Wspomagania Zarządzania Kryzysowego dla m. stołecznego Warszawy

Idea interesującej nas części symulacyjnej systemu przedstawiona została na Rys. 1

Baza danych o aktualnym stanie elementów aglomeracji zawiera dane określające aktualny stan elementów mających wpływ na stan bezpieczeństwa aglomeracji. Można wymienić tu takie elementy jak:

- źródła zagrożeń (uwzględniamy te źródła, które ze względu na swoje położenie i zasięg konkretnego typu zagrożenia wpływają na stan bezpieczeństwa aglomeracji),
- zakłady wykorzystujące niebezpieczne substancje chemiczne, biologiczne, radiacyjne, wybuchowe,
- magazyny niebezpiecznych substancji,
- transporty substancji niebezpiecznych,
- źródła klęsk żywiołowych,
- stan infrastruktury miejskiej:
  - sieciowych – sieci energetycznej, ciepłej, wodociągowo- kanalizacyjnej, gazowej, sieci łączności, sieci transportowej,
  - systemów powiadamiania i alarmowania,

- stan elementów wykonawczych (służb miejskich) wykorzystywanych w sytuacjach kryzysowych:
  - straży pożarnej,
  - policji,
  - służb medycznych,
  - pogotowia gazowego, energetycznego itd.,
- rozmieszczenie populacji mieszkańców, dane o dobowej migracji ludności.

Źródłami danych, zasilających bazę danych o aktualnym stanie bezpieczeństwa są:

- pomiary pochodzące z sieci sensorów monitorujących źródła zagrożeń oraz stan elementów infrastruktury miejskich,
- dane napływające z baz danych służb miejskich,
- dane napływające z systemów wczesnego ostrzegania (SWO).

Dane zawarte w bazie powinny umożliwiać modułowi **oceny aktualnego stanu bezpieczeństwa** ciągłą ocenę aktualnego stanu bezpieczeństwa aglomeracji oraz prognozę możliwości wystąpienia zagrożeń. Prognoza wygenerowana przez ten moduł zawiera określenie typu zagrożenia oraz rozkładu czasu do jego wystąpienia.

W chwili pojawienia się możliwości wystąpienia lub wystąpienia zagrożenia, moduł **rozpoznawania sytuacji kryzysowej** wspomaga rozpoznanie sytuacji kryzysowej, wykorzystując zawarte w bazie wiedzy antykryzysowej wzorce sytuacji kryzysowej oraz reguły ich rozpoznawania. Na podstawie zidentyfikowanej sytuacji kryzysowej, system udostępnia zespołowi reagowania kryzysowego warianty działania dostosowane do zaistniałej sytuacji.

Moduł **wypracowania decyzji** udostępnia funkcje wspomagania decyzji w zakresie planowania działań we wszystkich fazach zarządzania kryzysowego.

Istotnym elementem ISWZK są symulatory rozprzestrzeniania się zagrożeń. Umożliwiają one symulowanie prawdopodobnych scenariuszy rozwoju przewidywanych lub zaistniałych zagrożeń oraz oceniają ich skutki. Zatem mogą służyć prognozowaniu rozprzestrzeniania się zagrożeń. Ponadto wyniki symulacji mogą być podstawą do oceny skuteczności planowanych działań antykryzysowych, czyli mogą służyć wspomaganiu decyzji w różnych fazach zarządzania kryzysowego. Jeszcze inny sposób wykorzystania proponowanych symulatorów, to prowadzenie szkoleń indywidualnych oraz grupowych, czyli ćwiczeń zespołu reagowania kryzysowego.

Kolejnym istotnym elementem ISWZK jest graficzny interfejs użytkownika (GUI) wraz ze zobrazowaniem 2D i 3D. Elementy te umożliwią personelowi Centrum Zarządzania Kryzysowego lepszą ocenę stanu bezpieczeństwa aglomeracji. Szczególnie dotyczy to wizualizacji miejsc wystąpienia zagrożenia oraz jego rozszerzania się w przestrzeni i czasie.

Ostatnim elementem systemu jest baza danych normatywnych. Zawiera ona takie elementy jak: dane techniczne i przeznaczenie sprzętu istotnego z punktu widzenia zwalczania zagrożeń, akty prawne normujące zasady postępowania w sytuacjach występowania zagrożeń.

Elementami wykonawczymi, do których przekazywane są decyzje będą również, oprócz służb takich jak straż pożarna, policja czy służby medyczne, media informujące w odpowiedni sposób mieszkańców aglomeracji o zbliżającym się lub istniejącym już zagrożeniu oraz właściwym w tej sytuacji sposobie postępowania.

### 3. Konceptcje tworzenia GUI

Warto zauważyć że przedstawiona w poprzednim rozdziale architektura systemu ISWZK jest architekturą stosunkowo elastyczną i tym samym umożliwia zastosowanie szerokiego wachlarza różnych technik integracji poszczególnych komponentów systemu. W niniejszym rozdziale zostaną przedstawione cztery podstawowe metodologie przy pomocy których możliwe jest zaprojektowanie GUI dla przedstawionego w poprzednim rozdziale systemu.

#### 3.1. Konceptcja 1. Aplikacja monolityczna

Pierwszą i najbardziej klasyczną koncepcją tworzenia GUI jest ścisła integracja wszystkich komponentów systemu do postaci aplikacji i zaprogramowanie komunikacji pomiędzy GUI a resztą systemu przy pomocy bezpośredniego wywoływania procedur na poziomie kodu wykonywalnego. Pomimo oczywistych zalet takich jak:

- Zwartość aplikacji, teoretycznie możliwy jest pojedynczy skrypt VBA realizujący całość zadań
- Brak zależności od zewnętrznych w stosunku do ISWZK komponentów softwareowych
- Najwyższa możliwa efektywność kodu

sposób ten ma jednak bardzo istotną wadę. Mianowicie:

- Prace nad GUI nie mogą się rozpocząć do czasu aż będzie znane kompletne API systemu ISWZK

Oznacza to że w praktyce wymuszone jest stosowanie nieefektywnej metodyki „wodospadu”, uniemożliwiającej równoległe tworzenie komponentów systemu i tym samym w bardzo wysokim stopniu wydłużającej czas powstawania tegoż systemu. Z powodu tej wady ten sposób implementacji GUI został odrzucony.

#### 3.2. Konceptcja 2. Magistrala komunikacyjna

Oczywistym remedium na wady koncepcji aplikacji monolitycznej jest rozluźnienie związków pomiędzy poszczególnymi komponentami systemu. Ponieważ nie ma istotnej potrzeby aby poszczególne komponenty ISWZK korzystały nawzajem ze swojej funkcjonalności poprzez bezpośrednie wywoływanie funkcji na poziomie skryptów VBA, natomiast oczywistym jest że powinna istnieć możliwość wymiany danych i poleceń pomiędzy poszczególnymi komponentami systemu, kolejną koncepcją implementacji GUI stało się oparcie systemu o wspólną szynę komunikatów ActiveMQ. Niestety w praktyce okazało się że koncepcja ta pod względem harmonogramowym niewiele różni się od koncepcji aplikacji monolitycznej. W szczególności w dalszym ciągu do rozpoczęcia prac nad GUI wymagana jest zbyt szczegółowa specyfikacja typów komunikatów wymienianych pomiędzy sobą przez poszczególne komponenty systemu ISWZK oraz zbyt szczegółowy opis okoliczności w których te komunikaty są wymieniane. Tym samym okazało się że koncepcja magistrali komunikacyjnej nie umożliwia w istotny sposób zrównoleglenia prac nad komponentami systemu i w związku z tym koncepcja ta została również odrzucona.

#### 3.3. Konceptcja 3. Komponenty luźno powiązane

W kolejnej koncepcji nastąpiło zatem dalsze rozluźnienie powiązań pomiędzy komponentami ISWZK i tym samym narodziła się koncepcja oparcia systemu o technikę WebServices. Ponieważ w tej koncepcji zniknęła potrzeba posiadania szczegółowych i ostatecznych specyfikacji API i protokołów komunikacyjnych systemu, możliwe tym samym się stało równoległe rozpoczęcie prac nad wszystkimi komponentami systemu ISWZK praktycznie od samego początku projektu,

bez konieczności oczekiwania na wykonanie innych prac. Tym samym możliwe stało się znaczące skrócenie czasu wymaganego do stworzenia systemu.

Najważniejsze zalety koncepcji WebServices to:

- Prace nad dowolnym komponentem mogą rozpocząć się w dowolnym momencie, bez czekania na zakończenie prac nad innymi komponentami systemu
- Do rozpoczęcia prac nad komponentem wymagany jest jedynie ramowy szkic funkcjonalny komunikacji jaką ten komponent będzie prowadzić z innymi komponentami
- Zmiany w pojedynczym komponencie podjęte na późniejszym etapie projektu nie implikują kosztownych zmian w powiązanych z nim innych komponentach.

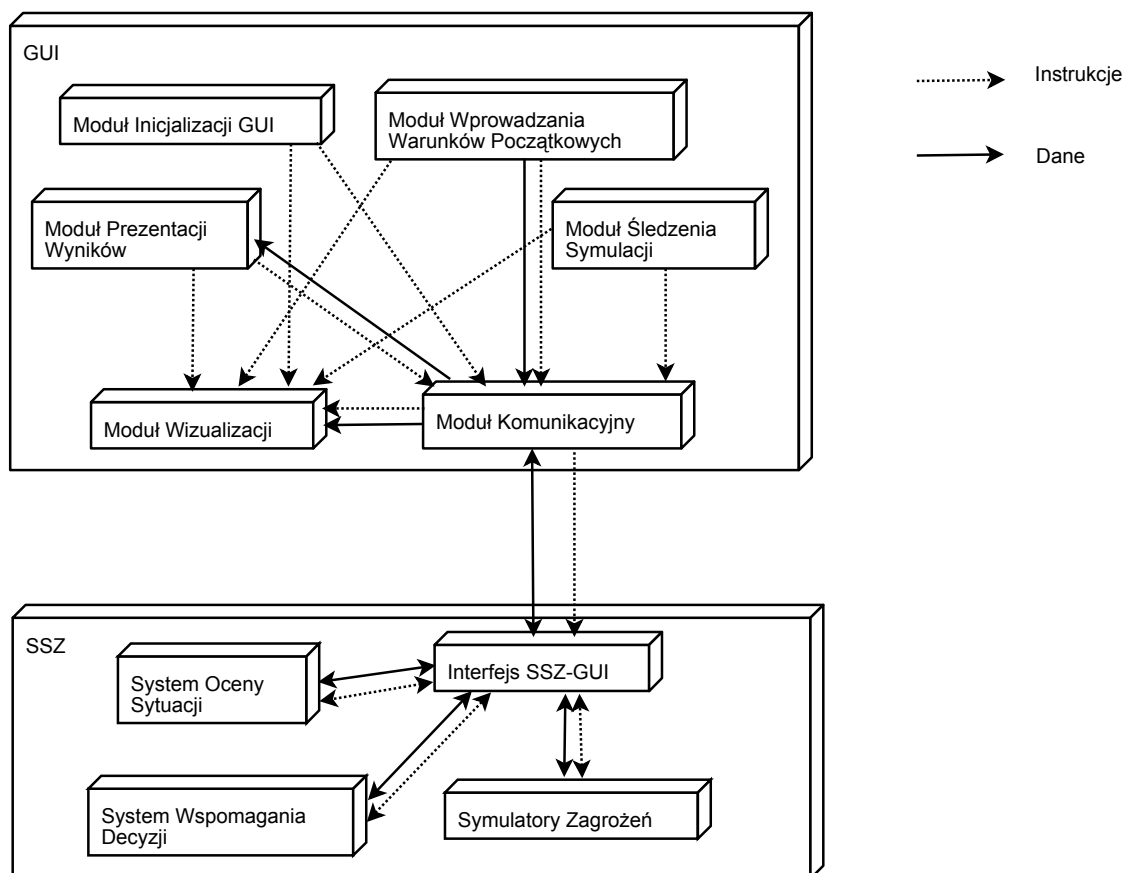
Ostatecznie koncepcja luźno powiązanych komponentów została przyjęta do realizacji.

### 3.4. Koncepcja 3.5. GUI jako samodzielna aplikacja

Ponieważ na zaawansowanym etapie projektu okazało się że prace nad GUI są prowadzone szybciej niż nad pozostałymi komponentami systemu ISWZK, powstała stąd niewielka modyfikacja koncepcji tworzenia GUI. Mianowicie okazało się że istnieje możliwość takiej modyfikacji GUI aby mogło ono działać jako samodzielna aplikacja nie powiązana z resztą systemu, stanowiąc tym samym demonstrator działania jeszcze nie ukończonego systemu. Modyfikacja ta polegała na zaimplementowaniu całości komunikacji pomiędzy GUI a pozostałymi komponentami w postaci specjalnych funkcji tzw. *wrapperów*, które zamiast wywoływać odpowiednie usługi WebServices korzystają z odpowiednio przygotowanych scenariuszy komunikacji i danych merytorycznych znajdujących się na lokalnej maszynie demonstratora. Stworzone w ten sposób GUI do nie istniejącego jeszcze systemu zostało pokazane w czasie demonstracji dla klientów systemu ISWZK i zostało przez klientów ocenione pozytywnie, udowadniając tym samym że pisanie GUI dla systemu którego jeszcze nie ma jest technicznie możliwe i celowe.

## 4. Architektura GUI

W ostatecznej wersji stworzony przez nas GUI systemu ISWZK posiada następującą strukturę logiczną, którą przedstawia Rys 2



Rys 2. Struktura logiczna GUI informatycznego systemu wspomagania zarządzania antykrzyszowego  
Kluczowe moduły GUI są następujące.

#### 4.1. Moduł Komunikacji

Najbardziej interesującym w kontekście niniejszego artykułu modułem jest moduł komunikacji. Odpowiada za wymianę danych pomiędzy Graficznym Interfejsem Użytkownika a Systemem Symulacji Zagrożeń (czyli symulacyjną częścią ISWZK). Moduł ten

1. Wysyła do Systemu Symulacji Zagrożeń:
  - a. Dane zawierające warunki początkowe symulacji.
  - b. Polecenie rozpoczęcia przetwarzania symulacji.
  - c. Polecenie przerywania przetwarzania symulacji.
  - d. Polecenie udostępnienia stanu zaawansowania symulacji.
  - e. Polecenia udostępnienia danych z Systemu Wspomagania Decyzji
  - f. Polecenia udostępnienia danych z Systemu Oceny Sytuacji
2. Przyjmuje z Systemu Symulacji Zagrożeń dane zawierające:
  - a. Wyniki symulacji
  - b. Wyniki działania Systemu Wspomagania Decyzji
  - c. Wyniki działania Systemu Oceny Sytuacji

3. Wysyła do Modułu Wizualizacji polecenia zmiany formy prezentacji danych.
4. Wysyła do Modułu Wizualizacji dane otrzymane z Systemu Symulacji Zagrożeń
5. Przyjmuje z Modułu Inicjalizacji GUI polecenia uzyskania stałych elementów symulacji.
6. Przyjmuje z Modułu Wprowadzania Warunków Początkowych polecenia
  - a. Przesłania wprowadzonych warunków początkowych do Systemu Symulacji Zagrożeń
  - b. Przesłania do Systemu Symulacji Zagrożeń polecenia rozpoczęcia przetwarzania symulacji.
7. Przyjmuje z Modułu Śledzenia Symulacji polecenia
  - a. Przerwania symulacji
  - b. Uzyskania informacji o stanie zaawansowania symulacji
8. Przyjmuje z Modułu Prezentacji Wyników polecenia
  - a. Uzyskania informacji o wynikach symulacji
  - b. Uzyskania informacji z Systemu Wspomagania Decyzji
  - c. Uzyskania informacji z Systemu Oceny Sytuacji

Moduł komunikacji został napisany w taki sposób że za każdą czynność komunikacji odpowiada osobna funkcja VBA, tzw. Funkcja *wrapper*. W obecnej implementacji GUI każda taka funkcja korzysta z przeprogramowanych danych merytorycznych i scenariuszy znajdujących się na lokalnym dysku maszyny na której demonstrowany jest GUI. W przyszłości natomiast poprzez prostą zmianę ciał funkcji *wrapperów* możliwe jest przekształcenie demonstratora we w pełni funkcjonalne GUI systemu ISWZK, jak tylko system ten powstanie. Mianowicie wystarczająca jest zamiana operacji I/O znajdujących się w ciałach funkcji *wrapperów* i skierowanych obecnie do lokalnych zasobów dyskowych poprzez wywołania odpowiednich usług zdalnych, które w przyszłości zostaną udostępnione dla GUI przez ISWZK poprzez mechanizm *WebServices*. Zauważyć należy iż poza zmianą ciał funkcji *wrapperów* do przekształcenia demonstratora we w pełni funkcjonalne GUI nie będzie wymagana żadna inna zmiana w kodzie źródłowym demonstratora. Można więc uznać że obecna wersja GUI jest w pełni działającym komponentem systemu ISWZK, posługująca się jedynie tymczasowymi mechanizmami wymiany danych.

## 4.2. Moduł Wizualizacji Danych

Odpowiada za wizualizację, formatowanie i prezentację danych zarówno w wprowadzonych przez Operatora Graficznego Interfejsu Użytkownika jak i będących wynikiem działania Systemu Symulacji Zagrożeń w sposób który jest zgodny z wymaganiami Operatora Graficznego Interfejsu Użytkownika. Moduł ten:

1. Otrzymuje dane z Modułu Komunikacji, które zostały wcześniej udostępnione przez System Symulacji Zagrożeń.
2. Otrzymuje z Modułu Inicjalizacji GUI polecenia ustalenia początkowej formy wizualizacji.
3. Otrzymuje z Modułu Wprowadzania Warunków Początkowych polecenia zmiany formy prezentacji danych.
4. Otrzymuje z Modułu Śledzenia Symulacji polecenia udostępnienia Operatorowi Graficznego Interfejsu Użytkownika informacji o stanie zaawansowania symulacji.
5. Otrzymuje z Modułu Prezentacji Wyników polecenia zmiany formy prezentacji danych oraz polecenia wizualizacji wyników symulacji.

W zależności od wybranego przez Operatora trybu pracy Moduł Wizualizacji Danych przedstawia wybrany fragment aglomeracji Miasta Stołecznego Warszawy w postaci widoku dwuwymiarowego lub trójwymiarowego. W obu trybach pracy do wizualizacji został wykorzystany System informacji Geograficznej ArcGIS, w przypadku wizualizacji dwuwymiarowej został wykorzystany komponent ArcMap systemu ArcGIS, natomiast w przypadku wizualizacji trójwymiarowej komponent ArcScene. W przypadku wizualizacji trójwymiarowej możliwe jest zapisanie wygenerowanej sceny w formacie VRML lub jako film Avi.

### 4.3. Moduł Inicjalizacji Graficznego Interfejsu Użytkownika

Odpowiada za prawidłowe osiągnięcie przez Graficzny Interfejs Użytkownika stanu gotowości do pracy a w szczególności za ustalenie domyślnych wartości początkowych danych zmiennych. Moduł ten

1. Wysyła do Modułu Wizualizacji polecenie ustawienia domyślnych wartości początkowych dla danych zmiennych.
2. Wysyła do Modułu Komunikacji polecenie uzyskania z Systemu Symulacji Zagrożeń danych stałych.

Moduł ten został zaimplementowany przy pomocy skryptów VBA będących wewnętrznym językiem programowania systemu ArcGIS.

### 4.4. Moduł Wprowadzania Warunków Początkowych

Umożliwia Operatorowi Graficznego Interfejsu Użytkownika wprowadzanie i modyfikację warunków początkowych symulacji. Moduł ten

1. Wysyła do Modułu Wizualizacji polecenia zmiany formy prezentacji danych.
2. Wysyła do Modułu Komunikacyjnego polecenia uzyskania danych z Systemu Symulacji Zagrożeń
3. Wysyła do Modułu Komunikacyjnego wprowadzone przez Operatora warunki początkowe.
4. Wysyła do Modułu Komunikacyjnego polecenie rozpoczęcia przetwarzania symulacji.

Moduł ten został zaimplementowany przy pomocy standardowych mechanizmów tworzenia graficznych interfejsów użytkowników, tzn. przy pomocy aplikacji sterowanej zdarzeniami (ang: event driver programming).

### 4.5. Moduł Śledzenia Symulacji

Utrzymuje kontakt z Systemem Symulacji Zagrożeń śledząc przebieg aktualnie obliczanej symulacji i informując Operatora Graficznego Interfejsu Użytkownika o postępach w obliczaniu bieżącej symulacji. Moduł ten

1. Wysyła do Modułu Wizualizacji polecenia zmiany formy prezentacji danych.
2. Wysyła do Modułu Komunikacyjnego polecenia przerwania symulacji.
3. Wysyła do Modułu Komunikacyjnego polecenia uzyskania bieżącego stanu zaawansowania symulacji.

Moduł ten został zaimplementowany przy pomocy mechanizmu zdarzeń periodycznych, tzn. w powiązaniu z zegarem systemowym maszyny klienckiej będzie co określony kwant czasu generowane zdarzenie sprawdzenia stanu symulacji (tzw. polling). Dzięki temu uniknięto konieczności pracy Graficznego Interfejsu Użytkownika w pasywnym trybie komunikacyjnym (tzn. jako serwer), co minimalizuje koszty licencyjne systemu.

#### 4.6. Moduł Prezentacji Wyników

Umożliwia Operatorowi Graficznego Interfejsu Użytkownika przeglądanie i ocenę wyników zakończonej symulacji, jak również zadawanie pytań skierowanych do Systemu Wspomagania Decyzji oraz do Systemu Oceny Sytuacji. Moduł ten:

1. Wysyła do Modułu Wizualizacji polecenia zmiany formy prezentacji danych.
2. Wysyła do Modułu Komunikacyjnego polecenia
  - a. Uzyskania danych będących wynikami symulacji.
  - b. Uzyskania danych z Systemu Wspomagania Decyzji
  - c. Uzyskania danych z Systemu Oceny Sytuacji
3. Przyjmuje z Modułu Komunikacyjnego
  - a. Dane uzyskane z Systemu Wspomagania Decyzji
  - b. Dane Uzyskane z Systemu Oceny Sytuacji

Moduł ten został zaimplementowany przy pomocy standardowych mechanizmów tworzenia graficznych interfejsów użytkowników, tzn. przy pomocy aplikacji sterowanej zdarzeniami (ang: *event driver programming*).

### 5. Metody integracji GUI z ISWZK, gdy ten ostatni powstanie

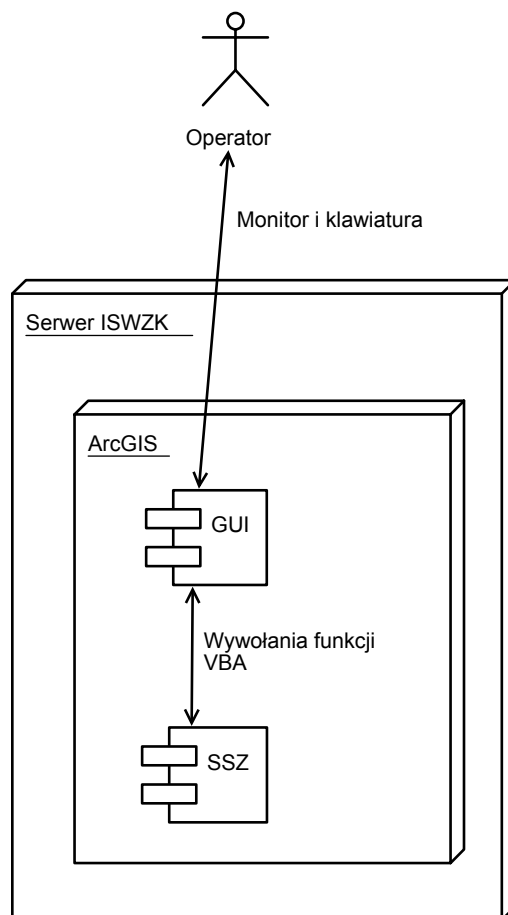
Jak już wspomniano w rozdziale 0, GUI zostało zaprojektowane w taki sposób, iż mimo że obecnie jest niezależną aplikacją możliwa jest jego prosta integracja z pozostałymi komponentami systemu ISWZK w miarę ich powstawania. Ze względu na to iż cała komunikacja została zaimplementowana przy pomocy funkcji *wrapperów*, poprzez prostą zmianę ciał tych funkcji możliwe jest osiągnięcie jednego lub więcej z opisanych niżej sposobów integracji. Warto zauważyć iż GUI jest tak zaprojektowane że zapewniona jest możliwość zastosowania każdego ze sposobów integracji GUI omówionych w rozdziale 3

#### 5.1. System monolityczny (lokalna konsola)

Istnieje możliwość osadzenia zarówno GUI jak i komponentu komunikacyjnego ISWZK na tym samym węźle sprzętowym i na tej samej instancji aplikacji ArcGIS co czyni z ISWZK system monolityczny z punktu widzenia integracji GUI z SSZ. W takim przypadku narzucającą się metodą integracji jest integracja ścisła tzn. umożliwienie bezpośredniego skrośnego wywoływania funkcji VBA pomiędzy tymi komponentami. Wady i zalety tej metody są następujące:

- Brak opóźnień komunikacyjnych pomiędzy GUI a SSZ
- Dostęp do pełnych możliwości zarówno GUI jak i SSZ przez drugi komponent
- Brak dodatkowych wymagań licencyjnych aplikacji ArcGIS
- Możliwość równoczesnego użycia wyłącznie jednej instancji GUI, a co za tym idzie niemożność korzystania z ISWZK przez więcej niż jednego operatora równocześnie
- Duże wymagania sprzętowe węzła, który musi równocześnie wykonywać zarówno obliczenia związane z przetwarzaniem danych jak i z wizualizacją tych danych
- Brak separacji pomiędzy GUI a resztą ISWZK zwiększa możliwość propagacji błędnych danych wprowadzonych przez Operatora do reszty systemu ISWZK
- Brak możliwości separacji geograficznej Operatora od węzła ISWZK

Z powyższych wad i zalet wynika że integracja monolityczna powinna być przeznaczona do zastosowań specjalnych takich jak konsola administracyjna superadministratora systemu. Schemat systemu ISWZK w wersji monolitycznej przedstawia Rys. 3



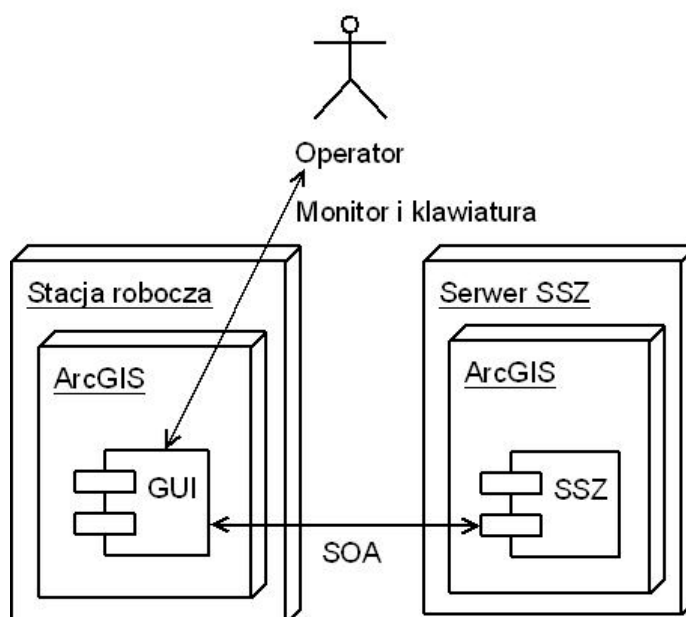
Rys. 3. Integracja GUI w postaci aplikacji monolitycznej

## 5.2. Wyspecjalizowana stacja robocza w systemie rozproszonym (tzw. thick client)

Inną możliwością integracji GUI z pozostałymi komponentami ISWZK jest osadzenie GUI na wyspecjalizowanej stacji roboczej zawierającej wyspecjalizowane aplikacje umożliwiające pracę GUI, oraz komunikującej się z pozostałymi komponentami ISWZK osadzonymi na innych węzłach za pomocą protokołu *WebServices* w wersji SOA. Wady i zalety tej metody są następujące:

- Wysoka wydajność dzięki rozdzieleniu wymogów sprzętowych GUI i SSZ na różne węzły systemu.
- Możliwość dowolnej separacji geograficznej operatora od serwerów ISWZK
- Możliwość pracy dowolnej ilości operatorów równocześnie
- Duże koszty, wynikające zarówno z konieczności zapewnienia dostatecznej ilości węzłów dla operatorów jak i z konieczności zakupu większej ilości licencji na aplikację ArcGIS
- Duże wymagania sprzętowe ArcGIS stawiają pod znakiem zapytania możliwość wykorzystania tego typu integracji dla urządzeń mobilnych niższej klasy niż laptopy typu *desktop replacement*.

Z powyższych wad i zalet wynika że integracja typu thick client powinna być przeznaczona dla stałych i niemobilnych operatorów ISWZK. Schemat systemu ISWZK w wersji thick client przedstawia Rys 4



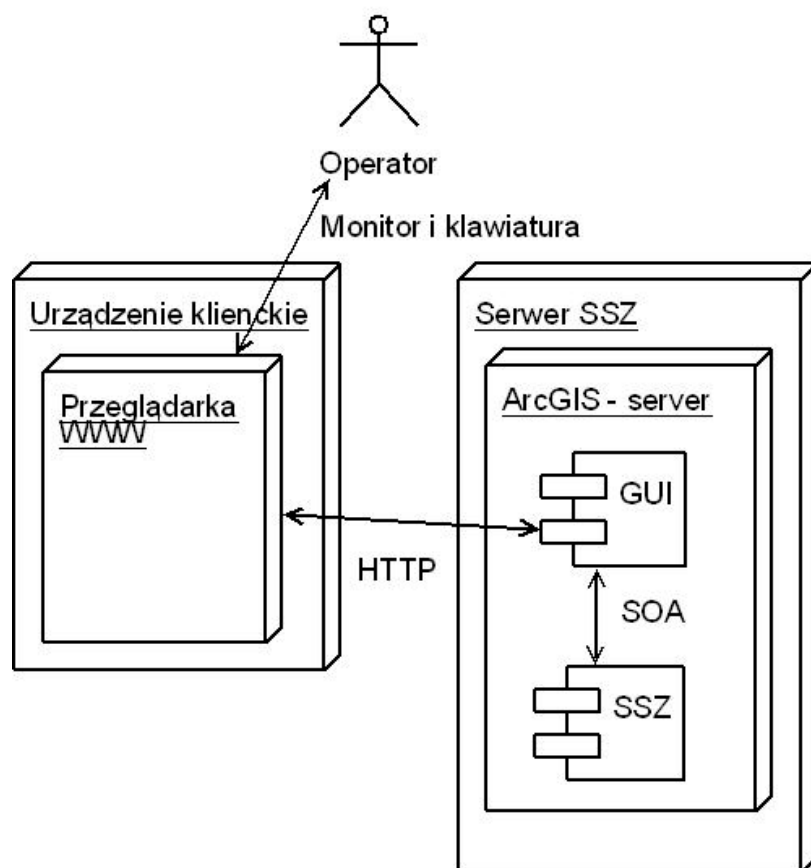
Rys 4. Integracja GUI z wykorzystaniem wyspecjalizowanej stacji roboczej

### 5.3. Urządzenie klienckie nie wymagające dostępu do wyspecjalizowanego oprogramowania (tzw. thin client)

Kolejną metodą integracji GUI z ISWZK jest użycie urządzenia klienckiego nie wykorzystującego wyspecjalizowanego oprogramowania, tzw. thin client. Z technicznego punktu widzenia sposób integracji GUI z ISWZK podobnie jak w poprzednim przypadku opiera się na protokole Web-Services w wersji SOA, natomiast sposób ten ma zdecydowanie inne implikacje dla operatorów systemu. Wady i zalety tej metody są następujące:

- Możliwość pracy na dowolnych urządzeniach klienckich, również mobilnych
- Możliwość dowolnej separacji geograficznej operatorów od serwerów ISWZK
- Brak dodatkowych wymagań licencyjnych aplikacji ArcGIS, z wyjątkiem konieczności użycia wersji serwer
- Możliwość dowolnego skalowania węzłów zawierających GUI w zależności od potrzeb
- Problemy wydajnościowe, w szczególności duże zapotrzebowanie na pasmo przenoszenia medium łączącego urządzenie klienckie z serwerem GUI

Z powyższych wad i zalet wynika że integracja typu thin client powinna być przeznaczona dla operatorów mobilnych, działających w terenie sytuacji kryzysowej. Schemat systemu ISWZK w wersji monolitycznej przedstawia Rys 5



Rys 5. Integracja GUI z wykorzystaniem cienkiego klienta

## 6. Podsumowanie

Przedstawiona koncepcja została stworzona i wykorzystana przy budowie GUI konkretnego systemu informatycznego, ale nic nie stoi na przeszkodzie, by była ona wykorzystywana również w innych przypadkach, kiedy tworząc GUI nie dysponujemy jeszcze działającym rdzeniem systemu.

Do chwili pisania tego tekstu integracja pomiędzy GUI i resztą systemu w omawianym projekcie jeszcze nie nastąpiła, dlatego nie można opisać jeszcze skuteczności przedstawionego podejścia. Jednak mimo tego, że nie nastąpiła weryfikacja proponowanego rozwiązania, wydaje się ono ze swej natury na tyle elastyczne, że prawdopodobnie może mieć zastosowanie w wielu różnorodnych środowiskach i architekturach systemowych. Wśród wad należy oczywiście wymienić zwiększoną pracochłonność, która wynika z konieczności zaimplementowania bardziej rozbudowanego interfejsu pomiędzy GUI i przyszłym systemem. Niemniej niedogodność tę można złagodzić zakładając, że stworzone fragmenty kodu tego interfejsu można użyć wielokrotnie przy budowie kolejnych systemów.

Zachęcamy do wypróbowania tego podejścia we własnych projektach, oczywiście jeśli nastąpią ku temu sprzyjające okoliczności.

**Bibliografia**

- [Lau07] Laurel B.: The Art of Human-Computer Interface Design, Addison-Wesley, 2007, ISBN 0-201-51797-3
- [Amd01] Amdahl G.: Disaster Response, ESRI Press, 2001, ISBN: 1-879102-88-9
- [Gre02] Greene R.: Confronting Catastrophe, A GIS Handbook, ESRI Press, 2002, ISBN 1-58948-040-6

