

Wizualizacja procesów przemysłowych w środowisku .NET

mgr inż. Paweł SROCZYŃSKI

Absolwent Wydziału Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego. Aktywny członek Koła naukowego UZ.NET. Programista .NET zajmujący się zaawansowanymi aplikacjami webowymi w jednej z zielonogórskich firm. Zainteresowania badawcze obejmują zagadnienia związane z projektowaniem interfejsu użytkownika w różnych technologiach oraz wizualizację procesów.

e-mail: P.Sroczynski@gmail.com



mgr inż. Piotr BUBACZ

Asystent w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego, absolwent Zintegrowanych Studiów Zagranicznych Uniwersytetu Zielonogórskiego i Fachhochschule Giessen-Friedberg (Niemcy). Opiekun koła UZ.NET. Zainteresowania badawcze obejmują sieci komputerowe, projektowanie systemów cyfrowych oraz formalnych metod oprogramowania sterowników logicznych.

e-mail: P.Bubacz@iie.uz.zgora.pl



prof. dr hab. inż. Marian ADAMSKI

Dyrektor Instytutu Informatyki i Elektroniki Uniwersytetu Zielonogórskiego. Zainteresowania badawcze obejmują projektowanie mikrosystemów cyfrowych oraz formalne metody programowania sterowników logicznych. Członek IEEE, IEE, ACM, Polskiego Towarzystwa Elektrotechniki Teoretycznej i Stosowanej oraz Polskiego Towarzystwa Informatycznego.

e-mail: M.Adamski@iie.uz.zgora.pl



Streszczenie

Wizualizacja ułatwia i automatyzuje sterowanie, diagnozowanie oraz monitorowanie procesów przemysłowych. Na podstawie analizy wady i zalety dwóch ważniejszych systemów do wizualizacji, dostępnych aktualnie na rynku, określono wymagania dla autorskiej aplikacji w środowisku .NET. Przedstawiono jej architekturę oraz przykład realizacji wizualizacji.

Abstract

Visualization eases and automates control and fault diagnosis of industrial processes. Based on analysis of two major visualization system on market, the requirements for new applications in .NET environment are defined. The architecture and sample visualization is presented.

Słowa kluczowe: wizualizacja procesów przemysłowych, .NET, WPF

Keywords: visualization of industrial processes, .NET, WPF

1. Wprowadzenie

Wizualizacja procesów przemysłowych jest ważnym zagadnieniem w przemyśle i badaniach naukowych. Ułatwia i automatyzuje sterowanie, diagnozowanie oraz monitorowanie procesów przemysłowych. Postęp naukowo-techniczny wymusza na systemach i aplikacjach do wizualizacji wymagania przede wszystkim co do jakości, wydajności, rozszerzalności, skalowalności oraz innowacyjności przy jednoczesnym zmniejszaniu kosztów wizualizacji. Wymagania te decydują o poziomie i tempie rozwoju przemysłowego. Z tego też względu dla zaspokojenia rosnących potrzeb przemysłu, stosowane są coraz bardziej zaawansowane narzędzia usprawniające wizualizację procesów przemysłowych. Aplikacje dostępne na rynku są zbyt proste lub zbyt drogie do zastosowania w badaniach naukowych i małych firmach przemysłowych. Dodatkowo nie umożliwiają łatwej rozbudowy o nowe elementy graficzne, moduły komunikacyjne oraz możliwość rozszerzania funkcjonalności w wielu językach programowania.

Przedmiotem niniejszej publikacji jest prezentacja wymagań, koncepcji i realizacji systemu umożliwiającego wizualizację procesów opisanych dowolnymi sygnałami wejściowymi. Aplikacja jest na tyle elastyczna, że umożliwia nie tylko odbieranie i wizualizowanie sygnałów z systemu przemysłowego, ale również komunikację z układami sterowania i aktywnymi elementami wykonawczymi wpływającymi na działanie obserwowanego systemu.

2. Wizualizacja

Wizualizacja [3] jest to graficzna reprezentacja modeli procesów technologicznych lub rzeczywistych prezentowanych na ekranie komputera. W systemach wizualizacji istnieje możliwość interakcji człowieka z komputerem (ang. Human-Machine Interaction – HMI [3]), dzięki czemu możliwe jest wpływanie na obserwowany proces przez użytkownika.

Technika wizualizacji obejmuje takie obszary jak tworzenie obrazów, diagramów oraz animacji. Ten ostatni jest szczególnym rodzajem wizualizacji. Animacja jest to przedstawienie danych zmieniających się w czasie.

Projektowanie wizualizacji

Przy projektowaniu wizualizacji należy zwrócić szczególną uwagę na takie elementy jak źródło i rodzaj prezentowanych danych oraz sposoby prezentowania informacji. Projektowanie wizualizacji polega na stworzeniu spójnego scenariusza, w którym możliwa jest komunikacja zarówno jedno-, jak i dwukierunkowa. Dzięki komputerowi możliwy jest podział różnych sygnałów oraz odpowiedni wybór konfiguracji, jaka jest potrzebna operatorowi. Można wyróżnić trzy podstawowe podejścia w projektowaniu systemów wizualizacyjnych [2]:

- **podejście terytorialne** – jest sposobem rozmieszczenia kontenerów graficznych, tak, aby każdy z osobnych elementów dostarczał informacji na temat urządzenia lub grupy urządzeń zlokalizowanym w jednym miejscu instalacji,
- **podejście technologiczne** – dotyczy grupowania danych ze względu na ich związek z określoną technologią np. zgromadzenie w jednym miejscu odmiennych parametrów elektrycznych z kilku różnych elementów oraz ich porównanie,
- **podejście hierarchiczne** – wykorzystywane są wszystkie informacje przychodzące z całego obsługiwanego procesu. Główny kontener graficzny zawiera symboliczny opis całego obiektu i po kliknięciu na odpowiednie pole przybliża szczegóły.

Techniki wizualizacji procesów przemysłowych

Istnieje wiele technik wizualizacji danych, opierających się o liczby, wykresy, animacje i wiele innych. Jednak nie ma lepszej wizualizacji danych dowolnego typu, niż wizualizacja graficzna, odzwierciedlająca rzeczywiste obiekty.

Z analizy różnorodnych sposobów wizualizacji wynika, że nie ma jednego, odpowiedniego sposobu wizualizacji danych dowolnego typu. Typowa wizualizacja tworzona jest poprzez określenie:

- danych wejściowych,
- zestawu obiektów wizualnych,
- zależności pomiędzy danymi wejściowymi a obiektami wizualnymi.

Na tej podstawie możliwe jest stwierdzenie, że najodpowiedniejszym systemem wizualizacji jest ten, który jest w stanie zaprezentować zestaw danych wejściowych w dowolny sposób, poprzez abstrakcyjne określenie znaczenia poszczególnych elementów i pozostawienie interpretacji wizualizacji człowiekowi.

Narzędzia stosowane do wizualizacji procesów

Na rynku dostępnych jest wiele aplikacji służących do kontroli oraz wizualizacji procesów produkcyjnych i technologicznych. Procesy mogą być przedstawiane w sposób estetyczny, elegancki i przejrzysty. Jednak czasem nie wystarczy wizualizowanie danych za pomocą konwencjonalnych metod.

W nowoczesnych systemach pojawiają się nie tylko klasyczne narzędzia i sposoby wizualizacji, ale również te nowoczesne [1]:

- diagramy, mapy reprezentujące słowa, idee, zadania, rzeczy, sklasyfikowane i połączone tematycznie i semantycznie za pomocą struktur drzewiastych przedstawionych w sposób multimedialny,
- prezentacja wiadomości przy pomocy rozrzuconych bloków tekstowych,
- prezentacja danych przy pomocy animacji.

Większość narzędzi jest wyspecjalizowanych do konkretnych zadań, lecz istnieją również takie, które umożliwiają zastosowanie do szerokiej gamy problemów. Korzystanie z tego typu narzędzi znacznie ułatwia i przyspiesza projektowanie wizualizacji oraz wdrażanie systemów udostępniających użytkownikom dane bezpośrednio z systemów sterowania i produkcji.

Użytkownik powinien jednak zwrócić szczególną uwagę na cel wizualizacji i przydatność systemu do wizualizacji. Przydatność takiego systemu wyznacza oferowane przez niego możliwości, jego dostępność oraz koszty związane z nabyciem i utrzymaniem. W dalszej części pracy opisane zostały dwa systemy wizualizacji danych oraz przedstawiono ich wady i zalety.

Aplikacja „Intouch”

Jest to przemysłowe oprogramowanie zaprojektowane do wizualizacji i kontroli procesów produkcyjnych. Wykorzystany jest tu system SCADA (HMI). Oferuje ono łatwość projektowania aplikacji oraz rozległą funkcjonalność, umożliwiającą szybkie projektowanie, testowanie oraz wdrażanie systemów udostępniających użytkownikom dane bezpośrednio z systemów sterowania i produkcji. Jest to także otwarte i elastyczne oprogramowanie, umożliwiające dostosowanie aplikacji do indywidualnych potrzeb przy zachowaniu szerokiego wachlarza połączeń z urządzeniami oraz systemami spotykanymi w przemyśle.

Aplikacja ta jest stosowana od prostych aplikacji jednoinstanowi-skowych, poprzez rozproszone systemy sieciowe o architekturze klient/ serwer, aż po systemy korzystające z możliwości Wonderware System Platform czy też Usług Terminalowych.

Program umożliwia różnorodną prezentację graficzną procesu produkcyjnego. W środowisku projektowym użytkownik znajdzie wiele narzędzi do rysowania obiektów. Możliwe jest także importowanie plików graficznych z innych programów graficznych

Zalety	Wady
– wiele możliwości komunikacji z zewnętrznymi mediami, – przejrzystość i duża czytelność, – graficzne przedstawienie danych, – rozwiązanie ściśle dedykowane dla systemów przemysłowych, – dynamiczna aktualizacja danych.	– duży stopień skomplikowania przy tworzeniu wizualizacji, – ubogi język programowania (QuickScript), – rozwiązanie ściśle dedykowane dla systemów przemysłowych (brak uniwersalności).

Tab. 1. Wady i zalety aplikacji „Intouch”

Tab. 1. Advantages and disadvantages of application „Intouch”

System „Cimplicity”

Oprogramowanie Proficy Machine Edition to kompletny, zintegrowany system do tworzenia aplikacji automatyki. Produkty Machine Edition są w pełni zintegrowane [1]:

- korzystają z tej samej bazy danych projektu oraz zapewniana jest pełna synchronizacja danych pomiędzy aplikacjami,
- posiadają wspólny zestaw narzędzi – użytkownik korzysta zawsze z tego samego interfejsu obsługi,
- w pełni obsługują metodę „przeciągnij i upuść”,
- posiadają cechy systemu uniwersalnego. Użytkownik posiada możliwość wyboru typu urządzeń, do których przesyłane będą projekty oraz na których będą one uruchamiane.

System posiada edytor paneli, który jest ściśle zintegrowany ze wszystkimi narzędziami „Machine Edition”, co ułatwia tworzenie paneli graficznych.

Zalety	Wady
– przejrzystość i duża czytelność, – graficzne przedstawienie danych, – możliwość wykorzystania z innymi systemami, – dynamiczna aktualizacja danych.	– duża złożoność systemu, – rozwiązanie ściśle dedykowane do systemów przemysłowych, – cena.

Tab. 2. Wady i zalety aplikacji „Cimplicity”

Tab. 2. Advantages and disadvantages of application „Cimplicity”

Przedstawione systemy i aplikacje do wizualizacji są oprogramowaniem dedykowanym, zorientowanym na konkretne zastosowania. W każdym z systemów stopień skomplikowania projektu wizualizacji jest zbyt wysoki. Jest to oprogramowanie, w którym moduły do wizualizacji są tylko częścią tych systemów, zatem nie jest możliwe ich wykorzystanie do wizualizacji w innych dziedzinach życia.

Powyższe przykłady oprogramowania, w swojej idei nie są zorientowane tylko na wizualizację. Niektóre z nich, to aplikacje hybrydowe, wyspecjalizowane do konkretnych zadań. Oczywiście na rynku istnieje więcej narzędzi do tworzenia wizualizacji, jednak żadne z nich nie umożliwiają wizualizacji procesów dowolnego typu, nie udostępnia interfejsów do rozwoju aplikacji (API) lub często posiada zbyt wiele ograniczeń spowodowanych brakiem elastyczności, bądź stopniem skomplikowania.

„Visualisation Studio” w założeniach jest aplikacją do wizualizacji praktycznie każdego rodzaju procesów, a ograniczenia takiej wizualizacji mogą wynikać z ograniczeń technologii .NET Framework, wyobraźni, bądź możliwości projektanta wizualizacji.

3. Aplikacja „Visualisation Studio”

Aplikacja „Visualisation Studio” umożliwi stworzenie wizualizacji procesów przemysłowych opisanych sygnałami wejściowymi poprzez modelowanie elementów wizualnych. Na rysunku 1 przedstawiono zrzut ekranowy aplikacji „Visualisation Studio”.



Rys. 1. Przykładowy zrzut ekranowy z aplikacji „Visualisation Studio”

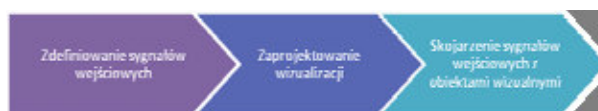
Fig. 1. Sample screen from „Visualisation Studio”

Analiza wymagań

Na podstawie analizy problemu oraz założeń projektowych aplikacja powinna spełniać następujące wymagania:

- **rozszerzalność** – możliwość rozwoju przez użytkownika poprzez nowe moduły,
- **interoperacyjność** z zewnętrznymi systemami – możliwość połączenia z dowolnym zewnętrznym systemem,
- **graficzny interfejs** – możliwość projektowania wizualizacji na zasadzie „przeciągnij i upuść”,
- **niezależność** – niezależność od dziedzin życia takich jak biznes, ekonomia, medycyna.

Zdefiniowanie sygnałów wejściowych polega na stworzeniu adaptera sygnałów, czyli na określeniu źródła danych oraz sposobu ich dostarczenia. W kolejnym kroku należy zaprojektować wizualizację tzn. stworzyć koncepcję wizualizacji danych a następnie stworzyć graficzną reprezentację procesu przemysłowego. Ostatnim krokiem jest skojarzenie sygnałów w celu określenia ich zachowania (zmiana rozmiaru, położenia, koloru itd.) (Rys. 2).



Rys. 2. Etapy projektowania wizualizacji

Fig. 2. Steps of visualization composition

Aplikacja umożliwia wizualizację danych wejściowych dowolnego typu. Dane wejściowe mogą mieć format zarówno numeryczny, jak i tekstowy. Format danych wejściowych może być rozszerzany poprzez konwertery – obiekty odpowiedzialne za konwersję między formatami (np. konwersja formatu numerycznego na tekstowy lub odwrotnie) i umożliwiające odzwierciedlenie wartości sygnału poprzez konkretną własność obiektu. Możliwe jest również sterowanie procesem poprzez definicję tzw. akcji w adapterze umożliwiających automatyzację procesu. Tak zdefiniowana infrastruktura programu powoduje, że zmiana zachowania systemu produkcyjnego pociąga za sobą zmianę sygnałów wejściowych, a co za tym idzie zmianę zachowania symulacji.

Na rysunku 3 przedstawiono abstrakcyjny model wykorzystania elementów aplikacji „Visualisation Studio”.



Rys. 3. Infrastruktura programu
Fig. 3. Program infrastructure

Obiekt wizualny jest podstawowym elementem graficznym wykorzystywanym w wizualizacji. Posiada opisujące go własności, które mogą być wykorzystywane i modyfikowane przez użytkownika. Obiekty wizualne posiadają tzw. własności podstawowe (pozycja, rozmiar) i rozszerzone (kolor wypełnienia, lokalizacja pliku graficznego). Istnieje możliwość rozszerzania aplikacji o obiekty graficzne. Dane wejściowe mogą być kojarzone z własnością obiektu wizualnego tak, jak pozwala na to konwerter.

Architektura aplikacji

Do zaprojektowania aplikacji niezbędny jest wybór architektury aplikacji, obiektów, z których będzie się składać oraz zakresu odpowiedzialności poszczególnych elementów składowych. Model warstwowy wydaje się najlepiej pasować do specyfiki aplikacji do wizualizacji.

Dane mogą być dostarczane do aplikacji z różnych urządzeń. Technologia nie ogranicza sposobu ich przekazywania ani przechowywania, zaś operacje mogą być wykonywane zarówno na jednym, jak i na wielu komputerach jednocześnie w przypadku zewnętrznego źródła sygnałów wejściowych.

Do implementacji aplikacji zdecydowano się wykorzystać .NET Framework 3.5 (WPF) [4]. Aplikacja została podzielona na trzy warstwy: warstwę prezentacji, logiki biznesowej oraz danych.

Warstwa prezentacji, reprezentowana w technologii .NET Framework 3.5 przy pomocy WPF, jest odpowiedzialna za część wizualną aplikacji oraz interakcję z użytkownikiem. Odgrywa bardzo ważną rolę ze względu na możliwość projektowania wizualizacji, jak również ze względu na samo przeprowadzanie wizualizacji danego procesu.

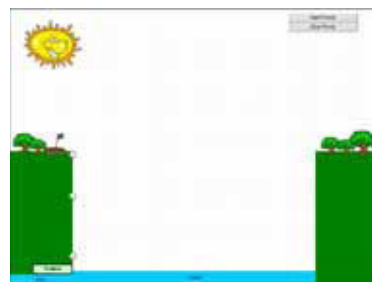
Warstwa logiki biznesowej jest częścią aplikacji odpowiedzialną za logiczne operacje takie jak wczytywanie modułów (konwertery, obiekty wizualne, adaptery). Zawiera część odpowiedzialną za m.in. zapis i odczyt wizualizacji, kontrolę danych odbieranych na potrzeby wizualizacji oraz definicję interfejsów określających kształt komponentów wizualnych, adapterów czy też konwerterów. Określa również sposób w jaki dane są przetwarzane na potrzeby wizualizacji.

Warstwa danych i dostępu do danych odpowiedzialna jest za zapis i odczyt zaprojektowanej wizualizacji do postaci plików XML oraz komunikację z zewnętrznymi systemami w celu wysyłania sygnałów sterujących lub odbioru sygnałów wejściowych. Znajdują się w niej tzw. adaptery umożliwiające dwustronną komunikację z zewnętrznymi źródłami danych (HTTP, FTP, bazy danych).

4. Przykład wizualizacji – możliwości aplikacji

Wizualizacja stanu zbiornika wody - pompownia

Rysunek 32 przedstawia wizualizację, która składa się m.in. z komponentów odzwierciedlających: stan wody, wodę, pompę



Rys. 4. Przykład wizualizacji
Fig. 4. Example of visualization

Wizualizacja polega na zmianie stanu wody w zbiorniku wodnym. Czujniki stanu wody zmieniają kolor na czerwony, gdy znajdują się pod powierzchnią i na zielony w przeciwnym wypadku. Pompa włącza się w momencie, gdy stan wody osiąga najwyższy poziom. Pompa wypompowując wodę powoduje obniżenie stanu wody. Wyłącza się, gdy poziom wody osiągnie stan najniższy. Takie zachowanie uwarunkowane jest zdefiniowanymi relacjami, które w zależności od stanu wody załączają bądź wyłączają pompę. Możliwe jest również włączanie lub wyłączanie pompy w dowolnym momencie kontrolując tym samym poziom wody.

Sygnał „WaterState” wpływa na zmianę poziomu wody, reprezentowanej na obszarze roboczym w postaci błękitnego prostokąta zmieniającego swoją wysokość. Sygnały z czujników („Sensor1”, „Sensor2”, „Sensor3”) skojarzone są z kolorem wypełnienia komponentów reprezentujących poszczególne sensory. Sygnał „PumpEnabled” odpowiada za zmianę koloru wypełnienia prostokąta reprezentującego pompę wodną.

5. Wnioski

W publikacji przedstawiono aplikację „Visualisation Studio” umożliwiającą wizualizację procesów przemysłowych. Z analizy istniejących narzędzi wynika, iż trudno znaleźć takie, które oferuje dostateczną elastyczność lub na tyle bogatą funkcjonalność, aby można było je stosować w dowolny sposób. Istniejące narzędzia do wizualizacji są najczęściej albo dedykowane do konkretnych zastosowań, lub stanowią integralną część większych systemów, najczęściej bardzo kosztownych. Stąd idea stworzenia samodzielnej aplikacji „Visualisation Studio”, umożliwiającej projektowanie wizualizacji procesów przemysłowych. Do stworzenia aplikacji wykorzystano możliwości technologii .NET Framework, w tym najnowszych rozwiązań do tworzenia wizualnych interfejsów użytkownika: biblioteki Windows Presentation Foundation.

Projektując aplikację „Visualisation Studio” wykorzystano architekturę trójwarstwową, z wydzielonymi warstwami: danych, logiki biznesowej i prezentacji. Podejście takie pozwoliło zmniejszyć ilość zależności pomiędzy komponentami, stworzyć czytelny podział odpowiedzialności poszczególnych grup komponentów, a w efekcie uprościć rozwój aplikacji.

Aplikacja udostępnia możliwość rozszerzania jej o nowe elementy, bez konieczności rekompilacji kodu źródłowego aplikacji. Udostępnienie punktów rozszerzalnych ma szczególne znaczenie dla użytkowników końcowych aplikacji, którzy mogą samodzielnie rozszerzać ją o nowe możliwości.

6. Literatura

- [1] GE Fanuc Automation Americas, Inc., „Proficy Machine Edition - Pierwsze kroki”, 2005
- [2] Rafał Cupek „Metody wizualizacji rozproszonych procesów przemysłowych”, Praca doktorska, Gliwice, maj 1998
- [3] „Wikipedia: Knowledge visualization”
- [4] Materiały informacyjne ze strony internetowej msdn.microsoft.com