



Asix.Evo - Elementy wizualizacji

*Dok. Nr PLP7E005
Wersja: 2014-03-24*

ASKOM® i **Asix®** to zastrzeżone znaki firmy **ASKOM Sp. z o. o., Gliwice**. Inne występujące w tekście znaki firmowe bądź towarowe są zastrzeżonymi znakami ich właścicieli.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną lub inną powoduje naruszenie praw autorskich niniejszej publikacji.

ASKOM Sp. z o. o. nie bierze żadnej odpowiedzialności za jakiegokolwiek szkody wynikłe z wykorzystywania zawartych w publikacji treści.

Copyright © 2014, ASKOM Sp. z o. o., Gliwice

ASKOM

ASKOM Sp. z o. o., ul. Józefa Sowińskiego 13, 44-121 Gliwice,
tel. +48 32 3018100, fax +48 32 3018101,

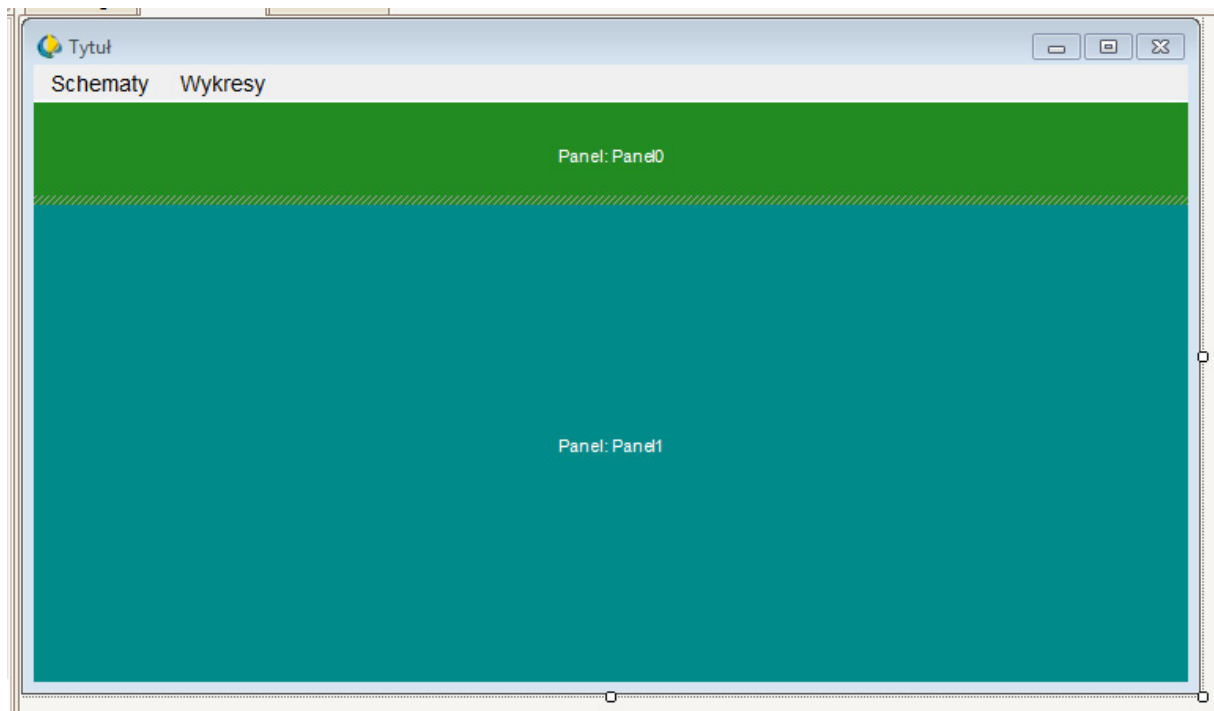
<http://www.askom.com.pl>, e-mail: biuro@askom.com.pl

Spis treści

| | |
|----------------------------------------------------------------------|----|
| 1 Hierarchia elementów wizualizacyjnych | 4 |
| 2 Właściwości i zdarzenia elementów wizualizacyjnych | 6 |
| 2.1 Prefiksy definicji właściwości | 8 |
| 3 Okna i panele | 12 |
| 3.1 Właściwości okien | 12 |
| 3.2 Właściwości paneli | 14 |
| 4 Diagramy | 16 |
| 4.1 Właściwości diagramów | 18 |
| 4.2 Zdarzenia diagramów | 18 |
| 4.3 Właściwości grup | 19 |
| 4.4 Zdarzenia grup | 19 |
| 4.5 Właściwości osadzonych wzorców | 19 |
| 4.6 Zdarzenia osadzonych wzorców | 19 |
| 5 Menu | 20 |
| 5.1 Parametry i kontrola elementów menu | 21 |
| 5.2 Właściwości menu | 22 |
| 5.3 Właściwości elementów menu | 22 |
| 6 Obiekty | 24 |
| 6.1 Grupy właściwości | 26 |
| 6.1.1 Właściwości podstawowe | 26 |
| 6.1.2 Grupy wielokrotne | 27 |
| 6.1.3 Właściwości stanowe | 28 |
| 6.2 Standardowe właściwości obiektów | 30 |
| 6.3 Standardowe zdarzenia obiektów | 32 |
| 7 Wzorce | 34 |
| 8 Właściwości globalne | 36 |
| 8.1 Zmiana wartości właściwości w trybie wykonywania aplikacji | 37 |

1 Hierarchia elementów wizualizacyjnych

Każda aplikacja Asix.Evo jest wyświetlana w tzw. oknach wizualizacyjnych. Liczba używanych okien jest dowolna, ale zawsze musi być użyte przynajmniej jedno. Jest to okno startowe aplikacji. Okna wizualizacyjne są obiektami systemu Windows i posiadają typowe właściwości okien, np. linia nagłówka, przyciski systemowe, menu systemowe, ramka. Oprócz tych elementów typowych dla systemu operacyjnego, okno wizualizacyjne składa się z opcjonalnego menu oraz jednego lub kilku paneli. Poniższa ilustracja pokazuje okno z menu i dwoma panelami w trakcie edycji.



Rys. Okno aplikacji w trybie edycji.

Panele okien są to kontenery przeznaczone do wyświetlania diagramów synoptycznych. W trakcie wyświetlania okna, diagramy w panelach mogą być wymieniane przy pomocy akcji operatorskich.

Diagram synoptyczny to główny element wizualizacyjny aplikacji służący do prezentacji stanu i obsługi kontrolowanego procesu. Diagram składa się z osadzonych w nim obiektów i wzorców.



Rys. Diagram z osadzonymi obiektami.

Obiekt to pojedynczy element diagramu. Sposób prezentacji i zachowanie obiektu zależy od jego typu. Dostępne są proste obiekty graficzne oraz bardzo złożone obiekty służące do prezentacji dużej ilości informacji, np. obiekty tabel alarmowych.

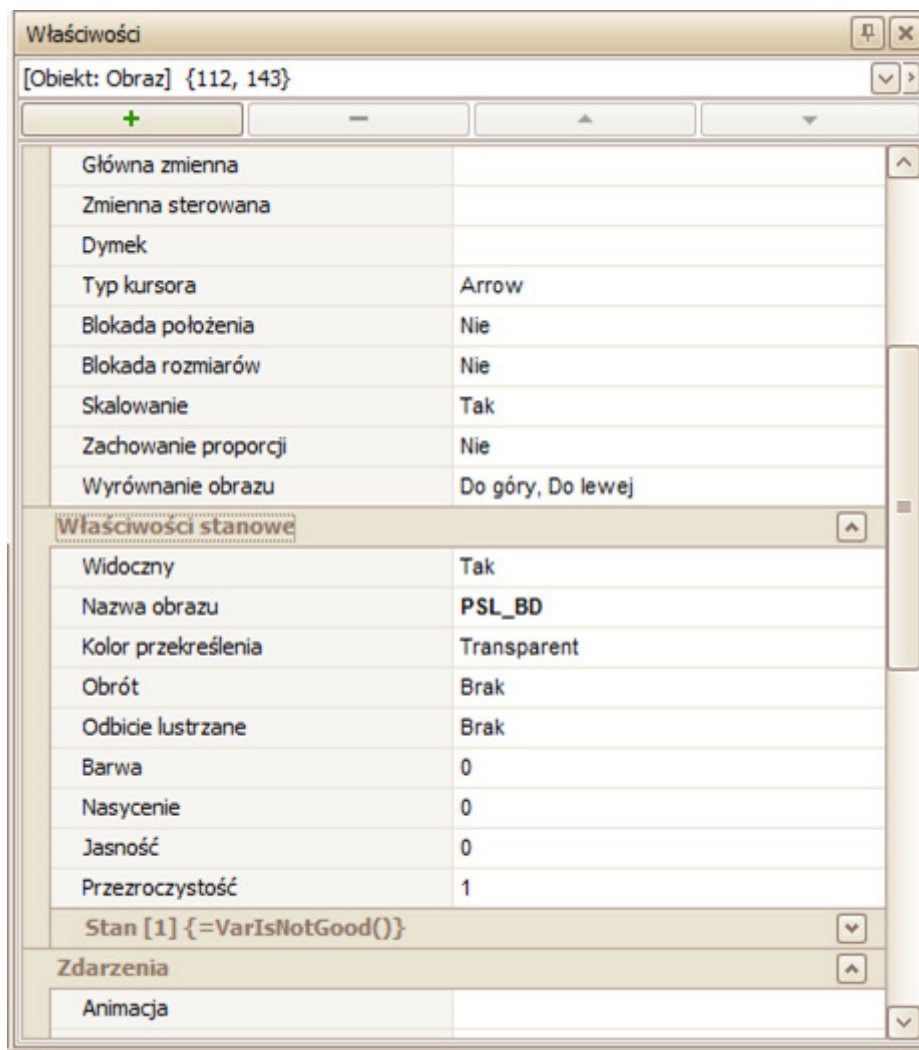
Wzorce to wstępnie sparametryzowane grupy obiektów. Przyspieszają proces tworzenia aplikacji poprzez wielokrotne osadzanie na diagramach przygotowanych wcześniej fragmentów wizualizacji.

Wszystkie elementy wizualizacyjne parametryzowane są poprzez zestawy właściwości i zdarzeń. Zestawy dostępnych właściwości i zdarzeń oraz możliwe metody ich parametryzacji są zależne od typu elementu. Dodatkowym elementem ułatwiającym parametryzację są tzw. właściwości globalne. Są to centralnie definiowane właściwości nie powiązane z żadnym konkretnym elementem aplikacji. Projektant aplikacji może odwołać się wielokrotnie do właściwości globalnej w definicjach różnych elementów. Stwarza to możliwość łatwej modyfikacji wyglądu lub zachowania aplikacji bez konieczności zmiany definicji właściwości wielu elementów składowych.

2 Właściwości i zdarzenia elementów wizualizacyjnych

Właściwości i zdarzenia elementów są definiowane w panelu *Właściwości*. Po wybraniu elementu w panelu jest wyświetlany odpowiedni zestaw właściwości i zdarzeń.

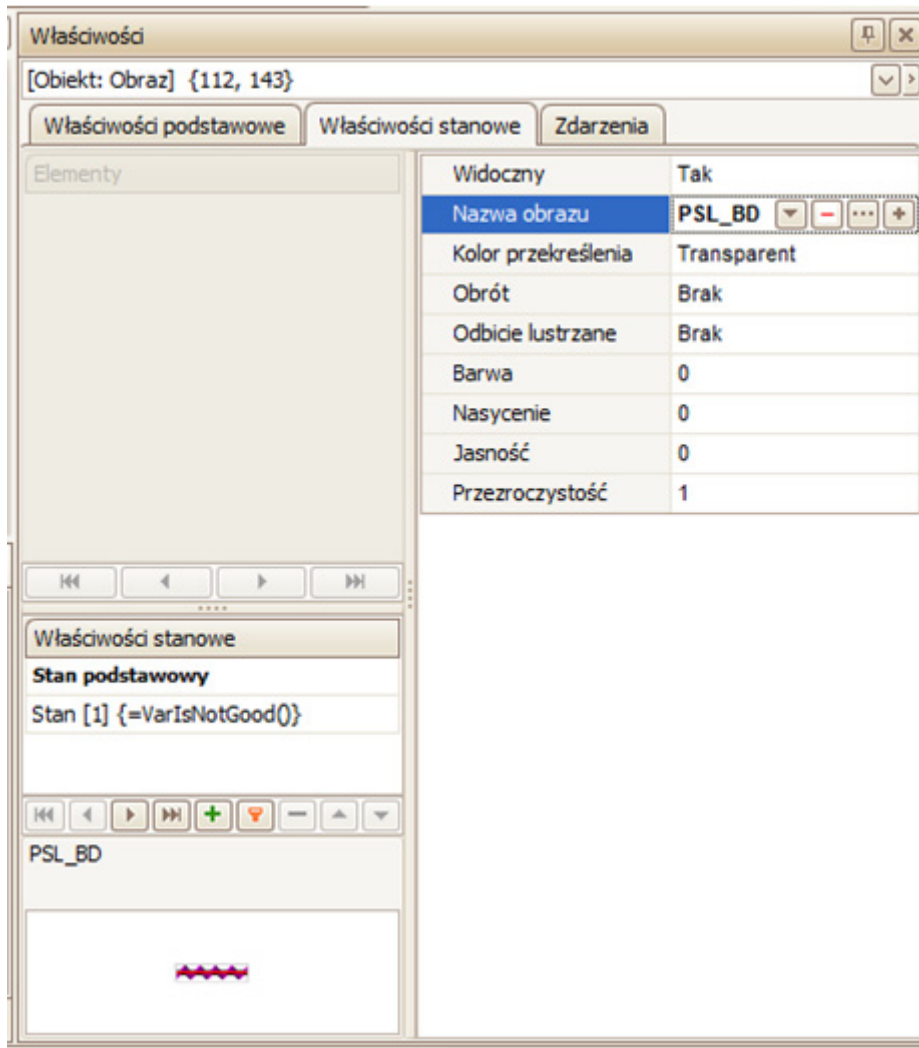
Dostępne są dwie wersje panelu właściwości – można zmienić tryb korzystając z przycisku w górnym prawym narożniku panelu.



Rys. Klasyczny panel właściwości elementów wizualizacyjnych.

Tzw. panel klasyczny pokazuje wszystkie właściwości, zdarzenia, parametry w jednej długiej liście z możliwością zwiżania poszczególnych fragmentów listy. Panel jest wąszy i pozwala na jednoczesne oglądanie elementów definicji tego samego typu.

Bardziej rozbudowany, panel kompaktowy pokazuje tylko elementy definicji wybrane przy pomocy zakładek i list wyboru. Panel jest szerszy, ale pozwala łatwiej wyszukiwać żądane elementy, szczególnie w przypadku rozbudowanych obiektów wielostanowych. Dodatkowo w lewym dolnym narożniku panelu, znajduje się pole podglądu wartości wynikowej dla wybranej właściwości.



Rys. Kompaktowy panel właściwości elementów wizualizacyjnych.

Właściwości elementu wizualizacyjnego są odpowiedzialne za jego wygląd i sposób działania. Typowym przykładem są właściwości służące do określenia koloru użytego do wyświetlenia elementu. Zdarzenia służą natomiast do wykonania akcji definiowanych przez projektanta w reakcji na zajście pewnych zdarzeń w trakcie wykonania aplikacji. Typowy przykład to zdarzenie kliknięcia klawiszem myszki w obszarze obiektu.

Zarówno właściwości jak i zdarzenia mogą być sparametryzowane w taki sposób, że ich wartość zmienia się w trakcie wykonywania aplikacji. Zmiany te mogą zależeć od zmian w stanie kontrolowanego procesu, od sposobu otwarcia elementu lub od uprawnień zalogowanego użytkownika i jego działań.

Definicja właściwości lub zdarzenia jest tekstem, który jest interpretowany w zależności od pierwszego znaku definicji i od kontekstu użycia. Definicje mogą być wprowadzane ręcznie lub przy pomocy okienek udostępnianych przez panel *Właściwości*.

2.1 Prefiksy definicji właściwości

Poniższa tabelka opisuje sposób interpretacji definicji właściwości w zależności od użytego znaku prefiksu.

| Prefiks | Znaczenie | Opis |
|------------|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ~ lub brak | Wartość bezpośrednia | <p>Wartość właściwości jest podana bezpośrednio i nie ulega zmianie w trakcie używania elementu. Sposób podania wartości zależy od typu właściwości.</p> <p>Przykłady:</p> <p><i>Red</i> Właściwości określające kolor akceptują nazwę koloru.</p> <p><i>Tak</i> Właściwości typu logicznego akceptują teksty <i>Tak</i> i <i>Nie</i>.</p> <p><i>Płaski</i> Specjalizowane właściwości mogą mieć własny zestaw dopuszczalnych wartości.</p> <p><i>10</i> Duża część właściwości akceptuje wartości liczbowe.</p> |
| # | Zmienna główna | <p>W zależności od kontekstu użycia, notacja z prefiksem # pozwala odwołać się do nazwy lub wartości zmiennej głównej. Nazwa zmiennej głównej jest definiowana we właściwości Główna zmienna obiektu. Odwołania do zmiennej głównej można stosować tylko w kontekście obiektu.</p> <p>W przypadku właściwości, które reprezentują nazwę zmiennej (np. Zmienna sterująca) składnia notacji jest następująca:</p> <p><i>#[sufiks]</i></p> <p>Użycie tylko znaku # oznacza wprost nazwę zmiennej głównej. Dodanie sufiksu pozwala utworzyć nazwę utworzoną z połączenia nazwy i zmiennej głównej i sufiksu.</p> <p>Przykład:</p> <p><i>#_limit</i></p> <p>Jeżeli zmienna główna posiada nazwę <i>Z1</i>, to zostanie utworzona nazwa <i>Z1_limit</i>.</p> <p>Dla właściwości, które nie mają charakteru nazwy zmiennej, prefiks # oznacza odwołanie do wartości zmiennej. W tym przypadku składnia notacji jest następująca:</p> |

| | | |
|---|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <p><code>#[sufiks][.numer_bitu[r]]</code></p> <p>Jeżeli sufiks nie jest podany, to pobierana jest bieżąca wartość zmiennej głównej obiektu. Jeżeli sufiks jest podany, to pobierana jest wartość zmiennej procesowej o nazwie złożonej z nazwy zmiennej głównej i podanego sufiksu.</p> <p>Opcjonalne użycie klauzuli numeru bitu pozwala zamiast wartości zmiennej, zwrócić wartość wybranego bitu jako wartość typu logicznego <i>true/false</i>. Znacznik <i>r</i> pozwala odwrócić logikę: wartość <i>true</i> jest zwracana, gdy wartość bitu jest równa 0.</p> <p>Wartość zwracana przez wyrażenie z prefiksem <i>#</i> musi być kompatybilna z typem właściwości, w kontekście której zostało użyte.</p> <p>Przykłady:</p> <p><i>#</i> Zwraca wartość zmiennej głównej. Użycie we właściwości <i>Tekst</i> obiektu <i>Tekst</i> spowodowałoby wyświetlenie wartości zmiennej skonwertowanej na postać tekstową.</p> <p><i>#_limit</i> Jeżeli zmienna główna posiada nazwę <i>Z1</i>, to pobrana zostanie wartość zmiennej <i>Z1_limit</i>.</p> <p><i>#.0</i> Zwraca wartość logiczną wyliczaną na podstawie stanu najmłodszego bitu wartości zmiennej głównej.</p> |
| & | Zmienna procesowa | <p>Pobranie wartości dowolnej zmiennej procesowej. Nazwę zmiennej należy podać po znaku prefiksu. Pełna składnia notacji jest następująca:</p> <p><code>&nazwa_zmiennej[.numer_bitu[r]]</code></p> <p>Opcjonalne użycie klauzuli numeru bitu pozwala zamiast wartości zmiennej, zwrócić wartość wybranego bitu jako wartość typu logicznego <i>true/false</i>. Znacznik <i>r</i> pozwala odwrócić logikę: wartość <i>true</i> jest zwracana, gdy wartość bitu jest równa 0.</p> <p>Wartość zwracana przez wyrażenie z prefiksem <i>#</i> musi być kompatybilna z typem właściwości, w kontekście której zostało użyte.</p> <p>Przykład:</p> <p><code>&z1</code> Pobranie wartości zmiennej <i>z1</i>.</p> |
| @ | Atrybut zmiennej | <p>Pobranie wartości atrybutu zmiennej procesowej z bazy</p> |

| | | |
|----|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <p>definicji zmiennych. Nazwę zmiennej procesowej i atrybutu należy podać po znaku prefiksu. Jeżeli nazwa zmiennej nie jest podana, to używana jest zmienna główna obiektu. Jeżeli nazwa atrybutu nie jest podana, to pobierany jest atrybut domyślny właściwości.</p> <p>Wartość atrybutu musi być kompatybilna z typem właściwości, w kontekście której została użyta.</p> <p>Przykład:</p> <p>@</p> <p>Pobiera atrybut domyślny zmiennej głównej. Można użyć tej formy tylko wtedy, gdy właściwość ma zdefiniowany atrybut domyślny.</p> <p>@Description</p> <p>Pobiera opis zmiennej głównej obiektu.</p> <p>@Z1.Description</p> <p>Pobiera opis zmiennej Z1.</p> |
| % | Parametr | <p>Pobranie wartości parametru wzorca lub diagramu. Nazwę parametru należy podać po znaku prefiksu. Parametry są mechanizmem pozwalającym na tworzenie wzorców i diagramów, których działanie zależy od parametrów dostarczanych dopiero w momencie użycia elementu. Wartość parametru musi być kompatybilna z typem właściwości, w kontekście której została użyta.</p> <p>Przykład:</p> <p>%zmienna</p> <p>Pobranie wartości parametru diagramu lub wzorca o nazwie <i>zmienna</i>.</p> |
| \$ | Napis wielojęzyczny | <p>Odwołanie do tekstu wielojęzycznego zdefiniowanego w panelu roboczym <i>Napisy aplikacji</i> (otwieranym za pośrednictwem panelu eksploratora aplikacji). Identyfikator napisu należy podać po znaku prefiksu.</p> <p>Prefiks \$ można stosować tylko we właściwościach o znaczeniu tekstowym. W trakcie działania aplikacji, w miejscu zastosowania zostanie wyświetlony tekst w aktualnym języku pracy aplikacji.</p> |
| ! | Właściwość globalna | <p>Pobranie wartości właściwości globalnej. Nazwę właściwości należy podać po znaku prefiksu.</p> <p>Wartość właściwości globalnej musi być kompatybilna z typem właściwości, w kontekście której została użyta.</p> <p>Przykład:</p> <p>!KolorLowLow</p> <p>Pobranie wartości właściwości globalnej o nazwie</p> |

| | | |
|---|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <i>KolorLowLow.</i> |
| = | Wyrażenie | <p>Wyliczenie wartości wyrażenia arytmetycznego. Treść wyrażenia należy podać po znaku prefiksu.</p> <p>Wynik wyrażenia może być zależny od wielu czynników (np. wartości zmiennych procesowych). Jest wyliczany na bieżąco w trakcie działania aplikacji.</p> <p>Zobacz <i>Wyrażenia</i>.</p> <p>Wynik obliczenia wyrażenia musi być kompatybilny z typem właściwości, w kontekście której wyrażenie zostało użyte.</p> <p>Przykład:</p> <p>= "Temperatura = " + Variable(t1)</p> <p>Wynikiem wyrażenia jest tekst łączący napis „<i>Temperatura =</i>” z wartością zmiennej <i>t1</i>.</p> |
| ^ | Akcja operatorska | <p>Definiuje treść akcji operatorskiej, która zostanie wykonana w momencie zajścia zdarzenia. Treść akcji należy podać po znaku prefiksu.</p> <p>W parametrach akcji mogą być wewnętrznie użyte wyrażenia. Stwarza to możliwość wykonywania akcji, których działanie zależy od czynników typu wartość zmiennej procesowej itp..</p> <p>Zobacz <i>Akcje operatorskie</i>.</p> <p>Przykład:</p> <p>^SetVariable("#",Variable()+1)</p> <p>Akcja spowoduje zwiększenie bieżącej wartości zmiennej głównej obiektu o 1.</p> |

UWAGA:

Nie każdy prefiks jest dopuszczalny w każdym kontekście użycia. W definicji zdarzeń można używać wyłącznie prefiks akcji operatorskich. Niektóre właściwości mogą być wprowadzane tylko w sposób bezpośredni – dotyczy to przede wszystkim właściwości okien i paneli.

3 Okna i panele

Okno jest podstawowym elementem aplikacji, w ramach którego wyświetlane są diagramy służące do prezentacji stanu kontrolowanego procesu. Większość właściwości okna służy do określenia jego wyglądu i dostępnych funkcji systemowych.

Okno może posiadać własne menu, którego nazwa jest podawane we właściwości *Menu*. W zależności od sposobu zdefiniowania menu, może się ono wyświetlić wzdłuż dowolnej krawędzi okna.

Każde okno składa się z paneli, w których w trakcie pracy umieszczane są diagramy. Określenie, które diagramy mają zostać wyświetlone w poszczególnych panelach, odbywa się:

- na etapie definiowania panelu, we właściwości *Domyślny diagram*.
- w trakcie pracy aplikacji w treści akcji operatorskiej *OpenWindow*

W trakcie definiowania paneli istotne jest określenie ich rozmiarów, szczególnie w przypadku, gdy projektowane okno ma zezwolenie na zmianę rozmiaru. Panele mogą dostosowywać swój rozmiar do zmienianego rozmiaru okna. Służą do tego właściwości paneli *Dokowanie* i *Zakotwiczenie*. Sposób działania tych właściwości najlepiej sprawdzić eksperymentalnie. W przypadku paneli dokowanych istotne znaczenie ma kolejność definiowania paneli.

Okno może zostać otwarte poprzez:

- zaznaczenie okna jako okna startowego w ustawieniach stanowiska – w tym przypadku okno jest otwierane w trakcie startu aplikacji.
- użycie akcji *OpenWindow*
- użycie akcji *OpenDiagram* – w tym przypadku tworzone jest tymczasowe okno złożone z pojedynczego panelu wypełniającego całe okno, o rozmiarze dostosowanym do rozmiaru diagramu

3.1 Właściwości okien

| Nazwa | Typ wartości | Opis |
|------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nazwa okna | Nazwa | Nazwa identyfikująca okno. Pierwszy znak musi być literą, a następne literą, cyfrą lub znakiem <code>_</code> . |
| Tytuł okna | Tekst | Opis zawartości okna wyświetlany na belce tytułowej |
| Kolor tła | Kolor | Kolor używany do wyświetlenia tła okna. Ma znaczenie w trybie edycji lub wtedy, gdy obszar okna nie jest całkowicie przykryty panelami. |
| X | Liczba | Współrzędna X otwarcia okna w trybie wykonawczym wyrażona w pikselach. Współrzędna otwarcia może być zmieniona w akcji operatorskiej otwierającej okno lub wynikać z trybu otwarcia podanego we właściwości <i>Stan początkowy okna</i> . |
| Y | Liczba | Współrzędna Y otwarcia okna w trybie |

| | | |
|---------------------------|---------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | wykonawczym wyrażona w pikselach. Współrzędna otwarcia może być zmieniona w akcji operatorskiej otwierającej okno lub wynikać z trybu otwarcia podanego we właściwości <i>Stan początkowy okna</i> . |
| Szerokość | Liczba | Szerokość okna wyrażona w pikselach. |
| Wysokość | Liczba | Wysokość okna wyrażona w pikselach. |
| Menu | Nazwa menu | Opcjonalna nazwa menu używanego w oknie. |
| Pokazuj belkę tytułu | Logiczna | Określa, czy belka tytułowa okna ma być wyświetlana. |
| Pokazuj ramkę | Logiczna | Określa, czy ramka wokół okna ma być wyświetlana. Niektóre inne ustawienia mogą również wymusić wyświetlenie ramki. |
| Stała pozycja | Logiczna | Określa, czy użytkownik może zmienić pozycję okna. |
| Stałe rozmiary | Logiczna | Określa, czy użytkownik może zmienić rozmiar okna. Nawet jeżeli rozmiar jest zablokowany, to można go zmienić przy pomocy akcji operatorskiej <i>SetWindowSize</i> . |
| Okno narzędziowe | Logiczna | Zmienia wygląd belki tytułowej i jednocześnie blokuje wyświetlanie przycisków systemowych. |
| Pokazuj przyciski | Logiczna | Blokuje wyświetlanie przycisków systemowych na belce tytułowej. |
| Przycisk maksymalizacji | Logiczna | Określa, czy na belce tytułowej ma być wyświetlany przycisk maksymalizacji. |
| Przycisk minimalizacji | Logiczna | Określa, czy na belce tytułowej ma być wyświetlany przycisk minimalizacji. |
| Zawsze pod spodem | Logiczna | Określa, czy okno jest wyświetlane zawsze pod spodem innych okien (aplikacji Asix.Evo i innych). Zazwyczaj używane dla dużych okien startowych aplikacji. Pozwala ograniczyć używanie trybu <i>zawsze na wierzchu</i> . |
| Zawsze na wierzchu | Logiczna | Określa, czy okno jest wyświetlane zawsze nad innymi oknami. Powoduje to, że okno nieaktywne nie będzie przesłaniane. Tryb ten nie należy stosować dla dużych okien bo mogą spowodować niepożądane całkowite zasłonięcie innych okien. Alternatywną metodą jest stosowanie trybu <i>zawsze pod spodem</i> dla okien startowych. |
| Stan początkowy okna | Typ wyliczeniowy: Zwykły, Zminimalizowane, Zmaksymalizowane, Wyśrodkowane | Określa tryb otwarcia okna. Tryb <i>zwykły</i> powoduje otwarcie zgodnie z właściwościami <i>X</i> , <i>Y</i> , <i>Szerokość</i> i <i>Wysokość</i> . Tryb <i>zminimalizowany</i> minimalizuje okno, a <i>zmaksymalizowany</i> powiększa (lub pomniejsza) okno do rozmiaru ekranu. Opcja ta jest szczególnie przydatna, gdy aplikacja powinna się automatycznie dopasować do różnych rozdzielczości ekranu. Opcja <i>wyśrodkowanie</i> wyświetla okno na środku ekranu z zachowaniem oryginalnych rozmiarów. |
| Autodopasowanie do ekranu | Logiczna | Właściwość wykorzystywana jest w przypadku uruchamiania aplikacji na różnych komputerach z monitorami o różnych rozmiarach. Okno z ustawionym autodopasowaniem, automatycznie zmniejszy swoje rozmiary, jeżeli zostanie otwarte na |

| | | |
|-------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | monitorze , na którym nie zmieści się w całości. |
| Zachowanie proporcji | Logiczna | Wartość <i>Tak</i> oznacza, że w trakcie zmieniania wielkości okna przez użytkownika, zachowane zostaną oryginalne proporcje rozmiarów |
| Przezroczystość | Liczba <0,1> | Określa stopień przezroczystości okna w zakresie od 0 do 1. Wartość 1 oznacza okno nieprzezroczyste, wartość 0 oznacza okno całkowicie przezroczyste. |
| Pokazuj na pasku zadań | Logiczna | Określa, czy informacja o otwartym oknie ma być wyświetlana na pasku zadań. |
| Zezwolenie na zamykanie | Logiczna | Blokuje użytkownikowi możliwość zamknięcia okna. Okna zablokowane można zamknąć tylko przy pomocy akcji <i>CloseWindow</i> . |

3.2 Właściwości paneli

| Nazwa | Typ wartości | Opis |
|-----------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nazwa panelu | Nazwa | Nazwa identyfikująca panel. Pierwszy znak musi być literą, a następne literą, cyfrą lub znakiem <code>_</code> . |
| Domyślny diagram | Nazwa diagramu | Nazwa diagramu, który należy automatycznie otworzyć w panelu. Używana, gdy okno jest oknem startowym aplikacji, lub gdy w akcji <i>OpenWindow</i> nie podano nazwy diagramu. |
| Domyślne parametry diagramu | Tekst | Parametry domyślnie otwieranego diagramu. Zestaw dozwolonych parametrów zależy od definicji diagramu. Parametry są podawane jako sekwencja par <i>nazwa_parametru=wartość_parametru</i> rozdzielonych znakiem średnika, np. Kolor=Red;Zmienna=v1 |
| Kolor tła | Kolor | Kolor używany do wyświetlenia tła panelu. Ma znaczenie głównie w trybie edycji. W trybie wykonawczym używany tylko wtedy, gdy do panelu nie został podłączony diagram. |
| Dokowanie | Typ wyliczeniowy: Brak, Wypełnienie, Do góry, Do dołu, Do lewej, Do Prawej | Określa tryb dokowania panelu. Tryb <i>wypełnienie</i> oznacza zajęcie całego wolnego miejsca. Tryb dokowania do krawędzi oznaczają zajęcia miejsca przy wybranej krawędzi – wielkość panelu określona jest wtedy przez właściwość <i>Szerokość</i> lub <i>Wysokość</i> . |
| Zakotwiczenie | Typ wyliczeniowy wielokrotny: Brak, Do góry, Do dołu, Do lewej, Do prawej | Określa sposób wyliczania rozmiaru panelu, gdy zmieniany jest rozmiar okna. Zakotwiczenie do wybranej krawędzi oznacza, że utrzymywana jest stała odległość okna od tej krawędzi. Można zakotwiczyć panel do kilku krawędzi jednocześnie. |
| X | Liczba | Współrzędna X pozycji panelu wyrażona w pikselach, liczona względem lewego górnego narożnika okna. Nie ma znaczenia w przypadku paneli dokowanych. |
| Y | Liczba | Współrzędna Y pozycji panelu wyrażona w pikselach, liczona względem lewego górnego narożnika okna. Nie |

| | | |
|-----------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | ma znaczenia w przypadku paneli dokowanych. |
| Szerokość | Liczba | Początkowa szerokość panelu wyrażona w pikselach. Może ulec zmianie w wyniku zmiany rozmiaru okna, gdy stosowane są tryby dokowania lub zakotwiczenia. |
| Wysokość | Liczba | Początkowa wysokość panelu wyrażona w pikselach. Może ulec zmianie w wyniku zmiany rozmiaru okna, gdy stosowane są tryby dokowania lub zakotwiczenia. |

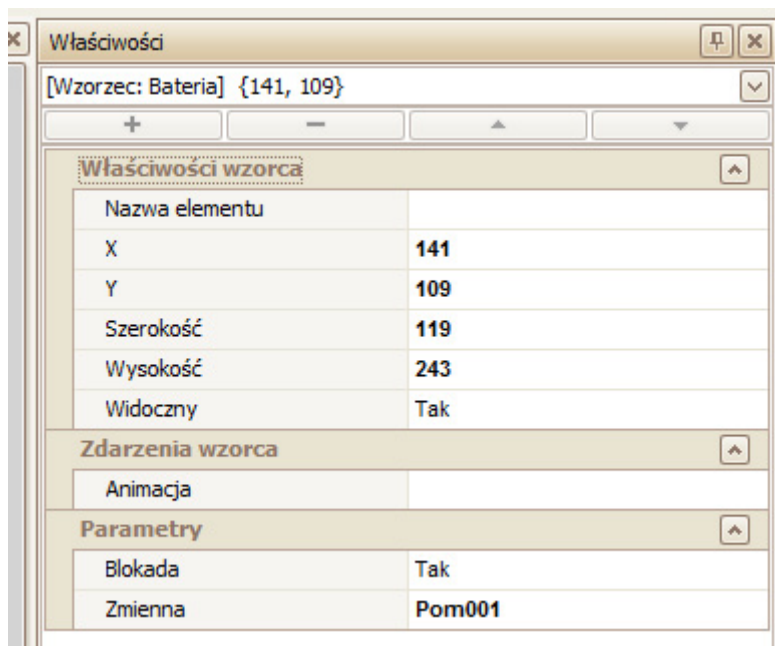
4 Diagramy

Diagram to główny element wizualizacyjny aplikacji służący do prezentacji stanu i obsługi kontrolowanego procesu. Diagramy są zawsze wyświetlane wewnątrz panelu okna. Okno może być otwarte poprzez:

- zadeklarowanie diagramu jako diagram domyślny panelu – w tym przypadku w momencie otwarcia okna (jeżeli nie będzie innych dyspozycji), diagram zostanie automatycznie wyświetlony.
- użycie nazwy diagramu w treści akcji *OpenWindow* – diagram zostanie otwarty we wskazanym panelu okna, akcja może dotyczyć nowo otwieranego okna lub okna otwartego wcześniej (wymiana diagramu).
- użycie nazwy diagramu w treści akcji *OpenDiagram* – diagram zostanie otwarty w automatycznie utworzonym oknie tymczasowym.

Istotnym aspektem definicji diagramu jest określenie, jak rozmiar diagramu dopasowywany jest do rozmiaru panelu. Służy do tego właściwość *Stałe rozmiary*. Jeżeli jest ona ustawiona na *Nie*, to rozmiar diagramu jest zawsze identyczny z rozmiarem panelu – w przypadku zmiany rozmiaru panelu położenie i wymiary wszystkich elementów diagramu jest przeskalowywane. Jeżeli tryb stałych rozmiarów został wybrany, to diagram nie jest skalowany, wyświetlany jest w oryginalnych rozmiarach, a w razie konieczności do panelu dodawane są suwaki pozwalające na wybór części widzialnej diagramu. Tryb stałych rozmiarów nie wyklucza jednak ręcznego skalowania diagramu. Ustawienie właściwości *Zezwalaj na powiększanie* na wartość *Tak* powoduje, że użytkownik może przeskalować wielkość diagramu przy pomocy rolki myszki (przy wciśniętym przycisku *Control*), ale w dalszym ciągu zmiana rozmiaru panelu nie ma wpływu na sposób wyświetlenia okna.

Diagram składa się z osadzonych w nim obiektów, grup obiektów i osadzonych wzorców. Grupy obiektów pozwalają na połączenie razem kilku obiektów znajdujących się na diagramie. Ułatwia to edycję obiektów (w szczególności selekcję, przesuwanie i kopiowanie obiektów grupy). Dodatkowo ułatwiona jest animacja ruchu grupy obiektów poprzez obsługę zdarzenia *Animate* grupy. Poza tym obiekty wchodzące w skład grupy zachowują się identycznie jako obiekty niezgrupowane. Inna sytuacja jest w przypadku osadzonego wzorca. Wzorzec podobnie jak grupa składa się z zestawu obiektów. W tym przypadku definicje tych obiektów nie znajdują się jednak wewnątrz definicji diagramu. Wzorce stanowią osobno definiowany element aplikacji. W diagramie przechowywana jest tylko informacja o sposobie osadzenia wzorca. Informacja ta składa się z właściwości określających rozmiar i położenie wzorca oraz wartości parametrów, które zostały użyte we wzorcu.



Rys. Panel właściwości wzorca.

Podobnie jak grupa, wzorzec posiada zdarzenie *Animacja*, które pozwala na animację ruchu wzorca.

Diagram może być diagramem parametryzowanym. Pozwala to używać ten sam diagram do wyświetlania informacji z różnych źródeł lub modyfikację wyglądu bez zmiany definicji diagramu.

Poprawne użycie parametrów składa się z dwóch kroków:

- Na etapie definicji diagramu definiowane są parametry diagramu. Definicja składa się z nazwy parametru oraz jego wartości domyślnej. Wartość domyślna używana jest w trakcie edycji diagramu oraz wtedy, gdy w akcji otwarcia diagramu nie zostały podane inne wartości parametru. Po zdefiniowaniu parametru można się do niego odwoływać we właściwościach obiektów poprzez notację sufiksową *%nazwa* lub wywołanie funkcji *Parameter*.
- W trakcie wykonywania aplikacji w akcjach otwarcia diagramu *OpenWindow* i *OpenDiagram* należy podać wartości parametrów. Parametry są podawane jako sekwencja par *nazwa_parametru=wartość_parametru* rozdzielonych znakiem średnika, np. *Kolor=Red;Zmienna=v1*.

Diagram posiada zdarzenia, które pozwalają na wykonanie dowolnych akcji operatorskich w momencie otwierania i zamykania diagramu.

4.1 Właściwości diagramów

| Nazwa | Typ wartości | Opis |
|-----------------------------|---------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nazwa diagramu | Nazwa | Nazwa identyfikująca diagram. Pierwszy znak musi być literą, a następne literą, cyfrą lub znakiem <code>_</code> . |
| Opis | Tekst | Opis diagramu o znaczeniu jedynie informacyjnym. |
| Kolor tła | Kolor | Kolor używany do wypełnienia tła diagramu |
| Obrazek tła | Nazwa obrazka | Nazwa obrazka, który zostanie użyty jako tło diagramu. |
| Tryb wypełniania obrazu tła | Typ wyliczeniowy: Oryginalny, Kafelki, Wyśrodkowany, Rozciąganie | Sposób wyświetlenia obrazka tła. Tryb oryginalny oznacza wyświetlenie obrazka w oryginalnych rozmiarach w lewym, górnym narożniku diagramu. W trybie kafelki obrazek jest wyświetlany wielokrotnie, tak aby wypełnić cały obszar diagramu. W trybie wyśrodkowania obrazek jest wyświetlany w oryginalnych rozmiarach na środku diagramu. W trybie rozciągania rozmiar obrazka jest dopasowywany do rozmiaru diagramu. |
| Stała szerokość | Liczba | Początkowa szerokość diagramu wyrażona w pikselach w trybie <i>stałych rozmiarów</i> . |
| Stała wysokość | Liczba | Początkowa wysokość diagramu wyrażona w pikselach w trybie <i>stałych rozmiarów</i> . |
| Stałe rozmiary | Logiczna | Określa, czy rozmiary diagramu są niezależne od rozmiaru panelu, w którym został wyświetlony. Wartość <i>Nie</i> oznacza, że diagram jest zawsze rozmiaru panelu – zmiana rozmiaru panelu powoduje przeskalowanie diagramu. Wartość <i>Tak</i> powoduje, że diagram jest wyświetlany zgodnie z rozmiarami podanymi we właściwościach <i>Stała wysokość</i> i <i>Stała szerokość</i> . Jeżeli rozmiar panelu jest mniejszy od rozmiaru diagramu, to zostaną użyte suwaki. |
| Zezwalaj na powiększanie | Logiczna | Określa, czy w trybie <i>stałych rozmiarów</i> użytkownik może przeskalować wielkość diagramu przy pomocy rolki myszki (przy wciśniętym przycisku <i>Control</i>). |

4.2 Zdarzenia diagramów

| Nazwa | Opis |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Otwarcie diagramu | Zdarzenie uruchamiane w momencie otwarcia diagramu. Typowy przykład użycia to utworzenie połączenia do AsBase'a przy pomocy akcji <i>AsbaseOpen</i> . |
| Zamknięcie diagramu | Zdarzenie uruchamiane w momencie zamknięcia diagramu. Typowy przykład użycia to zamknięcie połączenia do AsBase'a przy pomocy akcji <i>AsbaseClose</i> . |

4.3 Właściwości grup

| Nazwa | Typ wartości | Opis |
|----------------|--------------|----------------------------------------------------------------------------------------------------------------|
| Nazwa elementu | Nazwa | Nazwa identyfikująca grupę obiektów. |
| X | Liczba | Współrzędna X pozycji grupy obiektów wyrażona w pikselach, liczona względem lewego górnego narożnika diagramu. |
| Y | Liczba | Współrzędna Y pozycji grupy obiektów wyrażona w pikselach, liczona względem lewego górnego narożnika diagramu. |
| Widoczny | Logiczna | Określa, czy grupa obiektów jest widoczna. |

4.4 Zdarzenia grup

| Nazwa | Opis |
|----------|---------------------------------------------------------------------------------------|
| Animacja | Zdarzenie uruchamiane okresowe. Przeznaczone dla celów animacji ruchu grupy obiektów. |

4.5 Właściwości osadzonych wzorców

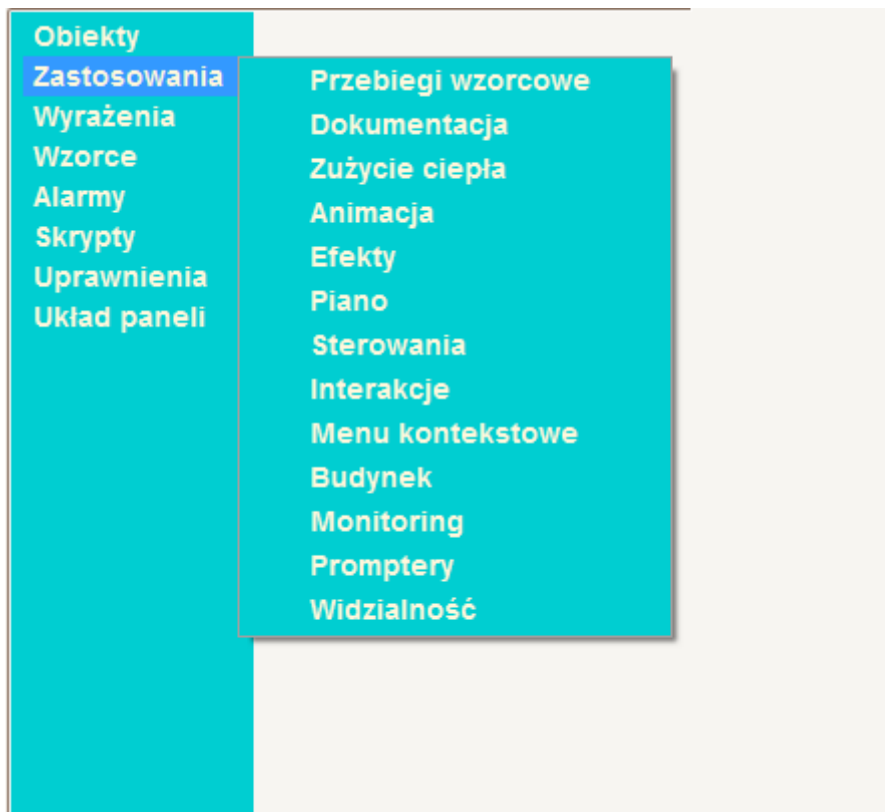
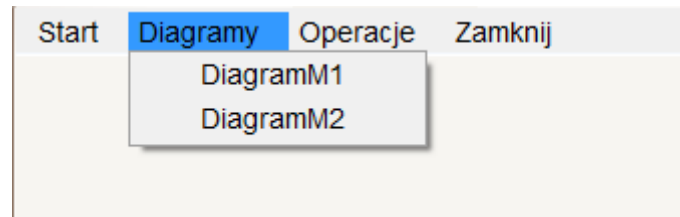
| Nazwa | Typ wartości | Opis |
|----------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nazwa elementu | Nazwa | Nazwa identyfikująca osadzony wzorec. |
| X | Liczba | Współrzędna X pozycji wzorca obiektów wyrażona w pikselach, liczona względem lewego górnego narożnika diagramu. |
| Y | Liczba | Współrzędna Y pozycji wzorca wyrażona w pikselach, liczona względem lewego górnego narożnika diagramu. |
| Szerokość | Liczba | Szerokość obszaru osadzenia wzorca |
| Wysokość | Liczba | Wysokość obszaru osadzenia wzorca |
| Warstwa | Liczba | Numer warstwy bazowej, względem której wyliczany jest numer warstwy obiektów składowych wzorca – numer warstwy wzorca jest dodawany do numeru warstwy obiektu. Obiekty o większym numerze warstwy przesłaniają obiekty o numerze mniejszym. Sposób wyświetlania obiektów o tym samym numerze warstwy wynika z kolejności ich zdefiniowania. |
| Widoczny | Logiczna | Określa czy grupa obiektów jest widoczna. |

4.6 Zdarzenia osadzonych wzorców

| Nazwa | Opis |
|----------|-------------------------------------------------------------------------------|
| Animacja | Zdarzenie uruchamiane okresowe. Przeznaczone dla celów animacji ruchu wzorca. |

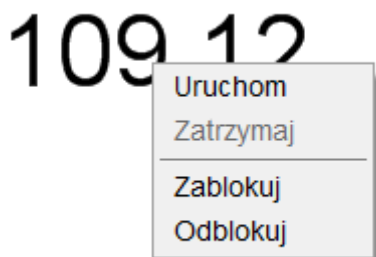
5 Menu

W aplikacjach Asix.Evo można stosować dwa rodzaje menu. Pierwszy rodzaj to menu powiązane z oknem. Można swobodnie definiować strukturę i wygląd menu. Menu podłączone jest do okna poprzez właściwość *Menu* okna. Właściwość *Dokowanie* określa sposób podłączenia menu do okna.



Rys. Przykład Menu powiązanego z oknem aplikacji.

Drugi rodzaj menu, to menu kontekstowe, z reguły podłączone do obiektów. Menu kontekstowe otwierane jest wynikiem wykonania akcji operatorskiej *ShowMenu*.



Rys. Przykład Menu kontekstowego powiązanego z obiektem.

5.1 Parametry i kontrola elementów menu

Menu kontekstowe może być menu parametryzowanym. Pozwala to używać tego samego menu w wielu miejscach i uzależniać jego działanie od parametrów przekazanych do menu. Ogólne zasady użycia parametrów są identyczne jak dla parametrów diagramów.

W przypadku menu kontekstowego, parametry są do niego przekazywane w momencie wykonania akcji otwarcia menu *ShowMenu*.

W przypadku menu powiązanego z oknem, przekazanie parametrów odbywa się w momencie każdorazowego wykonania akcji otwarcia diagramu w oknie *OpenWindow*. Parametry podane w akcji są w pierwszym kroku przekazywane do diagramu, ale jeżeli zachodzi zgodność nazwy dowolnego parametru z nazwą parametru menu powiązanego z oknem, to jest on również przekazywany do menu.

Wygląd menu może być uzależniany od stanu aplikacji. W szczególności właściwość *Widoczny* elementu menu pozwala na ukrywanie zbędnych elementów, a właściwość *Aktywny* kontroluje możliwość wykonania akcji powiązanej z elementem menu. Działanie menu można uzależnić od uprawnień użytkownika:

`HasRole(superoperator)`

lub od stanu procesu:

`Variable(v1)==0`

5.2 Właściwości menu

| Nazwa | Typ wartości | Opis |
|------------------------------|-----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| Nazwa menu | Nazwa | Nazwa identyfikująca menu. Pierwszy znak musi być literą, a następne literą, cyfrą lub znakiem <code>_</code> . |
| Kolor czcionki | Kolor | Kolor używany do wyświetlania tekstów w elementach menu. |
| Kolor tła | Kolor | Kolor tła elementów menu. |
| Kolor krawędzi | Kolor | Kolor obwódki wokół elementów menu. |
| Kolor tła podświetlenia | Kolor | Kolor tła elementów menu w momencie ich wybrania. |
| Kolor krawędzi podświetlenia | Kolor | Kolor obwódki wokół elementów menu w momencie ich wybrania. |
| Czcionka | Nazwa czcionki | Systemowa nazwa czcionki używanej do wyświetlania elementów menu. |
| Styl czcionki | Typ wyliczeniowy wielokrotny: Zwykły, Pogrubienie, Kursywa, Podkreślenie, Przekreślenie | Określa styl używanej czcionki. Można stosować wiele stylów jednocześnie. |
| Rozmiar czcionki | Liczba | Określa rozmiar używanej czcionki. |
| Dokowanie | Typ wyliczeniowy: Brak, Do góry, Do dołu, Do lewej, Do prawej | Określa przy której krawędzi okna należy umieścić menu. W przypadku menu kontekstowego obiektów należy stosować tryb <i>Brak</i> . |

5.3 Właściwości elementów menu

| Nazwa | Typ wartości | Opis |
|------------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tekst | Tekst | Tekst wyświetlany w pozycji menu |
| Nazwa obrazu | Nazwa obrazka | Nazwa obrazka wyświetlanego obok tekstu w pozycji menu. |
| Umieść po prawej | Logiczna | Określa, czy pozycja menu ma być wyświetlona po prawej stronie menu (u dołu dla menu dokowanych do pionowej krawędzi okna). Właściwość ma znaczenie tylko dla elementów pierwszego poziomu w menu powiązanych z oknem. |
| Widoczny | Logiczna | Określa, czy element menu ma być widoczny. Z reguły opcja jest wyliczana wyrażeniem uwzględniającym uprawnienia zalogowanego użytkownika lub wartości zmiennych procesowych. |
| Aktywny | Logiczna | Określa, czy element menu ma być aktywny. Elementy nieaktywne są wyświetlane wyszarzonym kolorem i nie wykonują akcji operatorskich. Z reguły opcja jest wyliczana wyrażeniem uwzględniającym uprawnienia zalogowanego użytkownika lub wartości zmiennych procesowych. |
| Separator | Logiczna | Określa, czy pozycja menu jest separatorem. Jeżeli tak, |

| | | |
|-------|-------------------|----------------------------------------------------------------------------------------|
| | | to w miejscu pozycji wyświetlana jest tylko linia separująca pozostałe pozycje w menu. |
| Akcja | Akcja operatorska | Akcja operatorska wykonywana w reakcji na wybranie elementu menu przez użytkownika. |

6 Obiekty

Obiekty są podstawowym elementem wizualizacyjnym. Są umieszczane na diagramach i we wzorcach. Wygląd i funkcje obiektów zależą od klasy obiektów. Dostępne są poniższe klasy obiektów:

- Dziennik zdarzeń – wyświetla tabela komunikatów systemowych
- Kłapa – wyświetlenie symbolu kłapy z sygnalizacją stopnia otwarcia
- Klawiatura – klawiatura ekranowa
- Komunikaty – wyświetlenie tekstu pobranego z pliku definicji komunikatów
- Kontroler wykresu – obiekt służący do sterowania pracą obiektów klasy *Wykres*
- Kształt – wyświetlenie kształtów na bazie prostokąta i elipsy
- Linia – wyświetlenie linii, również złożonych z wielu odcinków
- Miernik – wyświetlenie miernika wskazówkowego
- Obraz – wyświetlenie obrazka graficznego
- Pole autoryzacji – kontrolka do logowania użytkownika
- Przeglądarka – wyświetlenie stron www i zawartości dokumentów
- Przycisk – wyświetlenie przycisków w różnych stylach
- Punkt pracy – pokazuje położenie punktu w funkcji dwóch zmiennych określających współrzędne X i Y
- Rurociąg – obiekt w kształcie rurociągu
- Słupek – wyświetla dowolną wartość w postaci słupka o opcjonalnym suwakiem sterującym
- Tabela alarmów aktywnych – wyświetla tabelę pokazującego aktywne alarmy
- Tabela alarmów historycznych – wyświetla tabelę do przeglądania alarmów historycznych
- Tabela zmiennych – wyświetla wartości zmiennych procesowych w postaci tabelarycznej
- Taśmociąg – obiekt w kształcie taśmociągu
- Tekst – wyświetla informacje w postaci tekstowej
- Wielokąt – wyświetla kształt dowolnego wielokąta

- Wykres – wyświetla przebiegi zarchiwizowanych zmiennych lub krzywych wzorcowych
- Zbiornik – obiekt w kształcie zbiornika

Istnieje możliwość dodania do aplikacji własnych klas obiektów poprzez dołączenie odpowiednio napisanych modułów programowych.

Podstawowa funkcjonalność obiektu wynika z jego klasy. Projektant aplikacji ma jednak dużą swobodę dotyczącą szczegółów działania obiektów. Zdecydowana większość właściwości obiektów może być dynamicznie zmienna w trakcie pracy aplikacji. Pozwala to na uzyskiwanie praktycznie dowolnych efektów animacyjnych.

Niektóre klasy obiektów posiadają wbudowane funkcje interaktywne (np. obiekty *Tekst*, *Słupki*, *Przycisk*, tabele alarmów). W przypadku obiektów nieinteraktywnych (np. *Linia*, *Kształt*, *Rurociąg*) projektant może dodać takie funkcje poprzez obsługę zdarzeń. Na przykład, w każdym obiekcie można obsłużyć zdarzenie kliknięcia myszką i wykonać w takim momencie dowolną akcję operatorską.

Klasy obiektów dzielą się na dwa typy:

- Graficzne – typ podstawowy
- Kontrolkowe – obiekty oparte na wykorzystaniu standardowych kontroltek systemowych. Obiektami kontrolkowymi są *Dziennik Zdarzeń*, *Kontroler wykresu*, *Pole autoryzacji*, *Przeglądarka*, *Tabela alarmów aktywnych*, *Tabela alarmów historycznych*, *Tabela zmiennych*, *Wykres*.

Podstawowym ograniczeniem obiektów kontrolkowych jest brak możliwości nakładania się obiektów na siebie. Obiekty graficzne mogą być umieszczane w tym samym obszarze z kontrolą wzajemnego przesłaniania.

Obiekty są parametryzowane poprzez zestawy właściwości i zdarzeń. Istnieją zbiory tzw. standardowych właściwości i zdarzeń, które są obecne praktycznie we wszystkich klasach. Każda klasa może wprowadzać dodatkowe właściwości i zdarzenia dla niej specyficzne.

6.1 Grupy właściwości

Właściwości obiektów są zawsze umieszczone w jakiejś grupie. Istnieją trzy typy grup właściwości opisane w poniższych rozdziałach.

6.1.1 Właściwości podstawowe

Grupa właściwości występujących jednokrotnie. Najważniejsza właściwość w tej grupie to *Główna zmienna*. Ustala ona kontekst obiektu. Przy poprawnej definicji obiektu (użyciu funkcji w wersjach kontekstowych, np. *Variable()*, lub notacji sufiksowej nazw zmiennych) właściwość pozwala na tworzenie obiektów, które po skopiowaniu można przełączyć na pokazywanie innych danych poprzez zmianę tylko nazwy zmiennej głównej.

Wartości właściwości mogą ulegać dynamicznej zmianie poprzez użycie w ich definicjach wyrażeń i odwołań do elementów, których wartość ulega zmianie.

Przykład:

Właściwość: *Aktywny*

Definicja: = HasRole(DIR)

Status aktywności obiektu zmienia się w zależności od zalogowanego użytkownika.

Właściwość: *Dymek*

Definicja: @Description

Tekst wyświetlany w dymku będzie opisem zmiennej głównej obiektu

Właściwość: *Wartość obiektu Miernik*

Definicja: #

Położenie wskazówki obiektu *Miernik* będzie zależało od wartości zmiennej głównej obiektu.

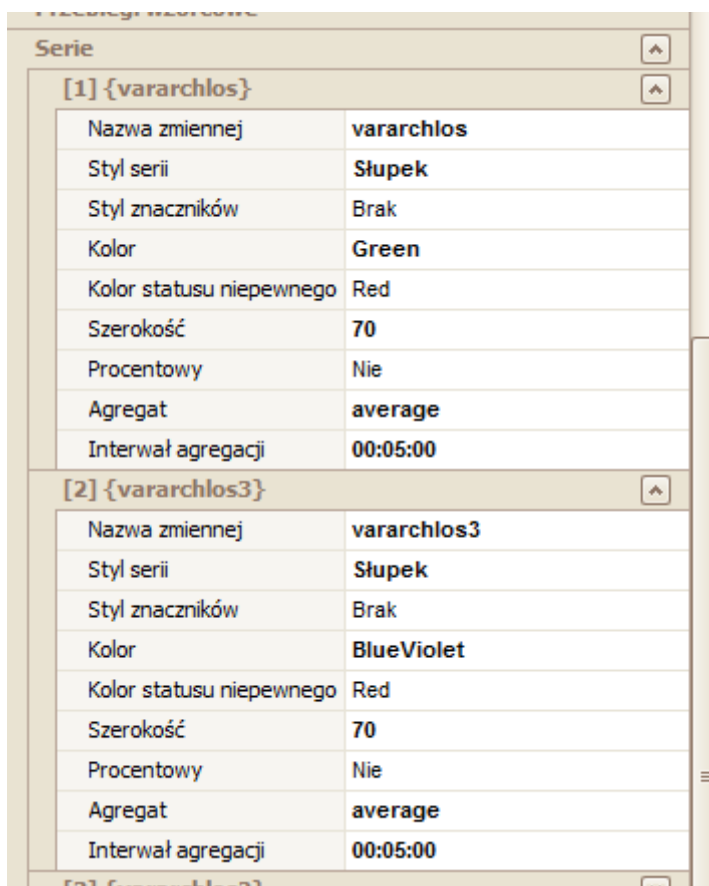
Właściwość: *Warstwa*

Definicja: =Variable(r1)>0?10:0

Jeżeli wartość zmiennej *r1* jest większa od 0, to numer warstwy ustawiany jest na 10. W przeciwnym wypadku jest równy 0. Powyższe wyrażenie pozwala dynamicznie kontrolować przesłanianie się obiektów.

6.1.2 Grupy wielokrotne

Grupy wielokrotne właściwości występują tylko w niektórych klasach obiektów, np. *Wykres*, *Tabela zmiennych*. Używane są wszędzie tam, gdzie w obiekcie tworzone są pewne powtarzalne elementy. Klasycznym przykładem są definicje wyświetlanych krzywych w obiekcie *Wykres*.



Rys. Grupy wielokrotne właściwości obiektu.

Projektant może utworzyć tylko tyle grup właściwości, ile jest potrzebnych. Każda grupa jest niezależna od pozostałych. Zasady wyliczania wartości są identyczne jak dla właściwości z grupy podstawowej.

6.1.3 Właściwości stanowe

Właściwości stanowe to specjalny rodzaj właściwości pozwalający na łatwe tworzenie obiektów zmieniających swój wygląd w zależności od stanu aplikacji i kontrolowanego procesu. Możliwe jest uzyskanie podobnych efektów bez wykorzystania grup stanowych, ale wymaga to użycia znacznie bardziej skomplikowanych wyrażeń.

Idea działania właściwości stanowych polega na tworzeniu wielu grup identycznych właściwości. W każdej z tych grup, właściwości mogą być definiowane w różny sposób. Oprócz tego grupa posiada dodatkową właściwość *Warunek stanu* typu logicznego, która określa, czy grupa brana jest pod uwagę przy wyliczaniu efektywnej wartości właściwości użytej do wyświetlenia obiektu.

Algorytm wyliczania wartości właściwości jest następujący:

- Wyszukiwana jest grupa, której warunek stanu ma wartość *Tak*, a definicja właściwości nie jest pusta.
- Jeżeli jest tylko jedna taka grupa, to wartość właściwości wynika z definicji użytej w tej grupie.
- Jeżeli jest więcej takich grup, to wartość właściwości pobierana jest z definicji ostatniej takiej grupy.
- Jeżeli żadna grupa nie jest aktywna, lub w żadnej z aktywnych grup właściwość nie jest zdefiniowana, to wartość określana jest na podstawie definicji w grupie podstawowej (pierwszej).

Przykład:

(verte)

| Właściwości stanowe | |
|----------------------------|------------------------|
| Widoczny | Tak |
| Nazwa obrazu | bt2_rad_red |
| Kolor przekreślenia | Transparent |
| Obrót | Brak |
| Odbicie lustrzane | Brak |
| Barwa | 0 |
| Nasylenie | 0 |
| Jasność | 0 |
| Przezroczystość | 1 |
| Stan [1] {=Variable()==1} | |
| Warunek stanu | =Variable()==1 |
| Widoczny | |
| Nazwa obrazu | bt2_rad_green |
| Kolor przekreślenia | |
| Obrót | 180 |
| Odbicie lustrzane | |
| Barwa | |
| Nasylenie | |
| Jasność | 0,1 |
| Przezroczystość | |
| Stan [2] {=VarIsNotGood()} | |
| Warunek stanu | =VarIsNotGood() |
| Widoczny | |
| Nazwa obrazu | |
| Kolor przekreślenia | Blue |
| Obrót | |
| Odbicie lustrzane | |
| Barwa | |
| Nasylenie | |
| Jasność | 0,3 |
| Przezroczystość | |

Rys. Przykład właściwości stanowych.

W pokazanym fragmencie definicji obiektu klasy *Obraz* pokazane jest, w jaki sposób wygląd obiektu uzależniony jest od wartości zmiennej oraz jej statusu. Grupa stanowa numer 1 aktywna jest wtedy, gdy wartość zmiennej głównej obiektu jest równa 0. Grupa stanowa numer 2 aktywna jest wtedy, gdy status zmiennej głównej jest niepoprawny. Sposób określenia wartości właściwości jest następujący:

| | |
|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Wartość zmiennej głównej jest różna od 1, a status zmiennej jest poprawny | Wartości wszystkich właściwości są pobierane z grupy podstawowej. |
| Wartość zmiennej głównej jest równa 1, status zmiennej jest | Wyświetlany obraz to bt2_rad_green , obraz jest obrócony o 180 stopni i ma zmienioną jasność ze współczynnikiem 0,1. Pozostałe |

| | |
|----------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| poprawny | ustawienia są pobierane z grupy podstawowej. |
| Wartość zmiennej głównej jest różna od 1, status zmiennej jest niepoprawny | Obrazek jest przekreślony i ma zmienioną jasność ze współczynnikiem 0,3. Pozostałe ustawienia są pobierane z grupy podstawowej, w szczególności wyświetlony obraz to <code>bt2_rad_red</code> . |
| Wartość zmiennej głównej jest równa 1, status zmiennej jest niepoprawny | Wyświetlany obraz to <code>bt2_rad_green</code> , obraz jest obrócony o 180 stopni, przekreślony i ma zmienioną jasność ze współczynnikiem 0,3. Współczynnik jasności pochodzi z grupy kontrolującej status zmiennej ponieważ jest ona zdefiniowana później. |

6.2 Standardowe właściwości obiektów

| Nazwa | Typ wartości | Opis |
|---------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nazwa elementu | Nazwa | Nazwa osadzonego obiektu. Pozwala zidentyfikować obiekt i zmieniać jego właściwości przy pomocy akcji <i>SetProperty</i> . |
| X | Liczba | Współrzędna X pozycji obiektu wyrażona w pikselach, liczona względem lewego górnego narożnika diagramu. |
| Y | Liczba | Współrzędna Y pozycji obiektu wyrażona w pikselach, liczona względem lewego górnego narożnika diagramu. |
| Szerokość | Liczba | Szerokość obszaru osadzenia obiektu wyrażona w pikselach. |
| Wysokość | Liczba | Wysokość obszaru osadzenia obiektu wyrażona w pikselach. |
| Warstwa | Liczba | Numer warstwy, na której wyświetlany jest obiekt. Obiekty o większym numerze warstwy przesłaniają obiekty o numerze mniejszym. Sposób wyświetlania obiektów o tym samym numerze warstwy wynika z kolejności ich zdefiniowania. |
| Aktywny | Logiczna | Określa, czy obiekt jest aktywny. W przypadku obiektów z wbudowanymi funkcjami sterującymi wartość <i>False</i> blokuje tę funkcjonalność. Dla obiektów bez standardowych funkcji interaktywnych (np. <i>Kształt</i>) wartość właściwości nie ma bezpośredniego znaczenia, może być jednak kontrolowana w akcjach i funkcjach użytych we właściwościach i zdarzeniach obiektu – stan aktywności można pobrać funkcją <i>IsActive</i> . |
| Tło obiektu aktywne | Logiczna | Określa w jaki sposób obiekt reaguje na kliknięcia myszką w trybie działania. Jeżeli wartość jest równa <i>Nie</i> , to kliknięcia we fragmentach pustych (wyświetlanych kolorem <i>Transparent</i>) są ignorowane. Pozwala to na przykład, na użycie częściowo przezroczystego obrazka, który reaguje na kliknięcia tylko w obszarze nieprzezroczystym. Właściwość nie ma znaczenia dla obiektów typu |

| | | |
|----------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | kontrolkowego, np. <i>Przeglądarka</i> . |
| Minimalna szerokość widoczności | Liczba | Minimalna szerokość obiektu w pikselach, przy której obiekt jest wyświetlany. Jeżeli w efekcie przeskalowania diagramu szerokość obiektu spadnie poniżej wartości granicznej, to obiekt jest automatycznie ukrywany. |
| Maksymalna szerokość widoczności | Liczba | Minimalna wysokość obiektu w pikselach, przy której obiekt jest wyświetlany. Jeżeli w efekcie przeskalowania diagramu wysokość obiektu spadnie poniżej wartości granicznej, to obiekt jest automatycznie ukrywany. |
| Główna zmienna | Nazwa zmiennej | Nazwa zmiennej głównej obiektu. Właściwość pozwalająca na centralne podanie w obiekcie nazwy zmiennej, od wartości i definicji której uzależnione jest działanie obiektu. W innych właściwościach obiektu można odwoływać się do zmiennej głównej przez kontekstowe funkcje typu <i>Variable</i> i <i>Attribute</i> , skrótowe odwołania do nazwy i wartości zmiennej głównej w postaci #, lub konstruować pochodne nazwy zmiennych stosując notację sufiksową # <i>sufiks</i> . |
| Zmienna sterowana | Nazwa zmiennej | Nazwa zmiennej, która jest zapisywana w efekcie użycia funkcji sterujących obiektu (dotyczy obiektów z wbudowanymi funkcjami sterującymi, np. <i>Tekst</i>). Przykład: # Zmienna sterowana jest identyczna ze zmienną główną obiektu. Jest to jedyny przypadek, gdy użycie notacji # oznacza nazwę zmiennej głównej, a nie jej wartość. #_s Nazwa zmiennej sterowanej powstaje przez połączenie nazwy zmiennej głównej i sufiksu _s. @ControlVariable Nazwa zmiennej sterowanej pobierana z atrybutu zmiennej głównej. |
| Dymek | Tekst | Definicja tekstu wyświetlanego w dymku podpowiedzi. Treść dymku może zależeć od wartości atrybutów z bazy zmiennych lub wartości zmiennych procesowych. Przykład: @Description Wyświetla opis zmiennej głównej obiektu =Format("{0} = {1}", Attribute(Name), Variable()) Wyświetla nazwę i wartość zmiennej głównej obiektu. |
| Typ kursora | Nazwa kursora | Systemowa nazwa typu kursora używanego w momencie najechania kursorem myszki na obiekt. |

| | | |
|-------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | Można w ten sposób sygnalizować gotowość obiektu na wykonanie operacji interaktywnych, np. wykonanie akcji po kliknięciu klawiszem myszki. |
| Blokada położenia | Logiczna | Określa, czy możliwa jest zmiana pozycji obiektu. Dotyczy tylko trybu edycji diagramu. |
| Blokada rozmiaru | Logiczna | Określa, czy możliwa jest zmiana rozmiaru obiektu. Dotyczy tylko trybu edycji diagramu. |
| Warunek stanu | Logiczna | Właściwość w grupie właściwości stanowych decydująca o tym, czy ustawienia grupy należy uwzględniać przy wyliczaniu efektywnych wartości właściwości stanowych. |
| Opis stanu | Tekst | Określa tekstowy opis grupy właściwości stanowych. |
| Widoczny | Logiczna | Określa, czy obiekt ma być wyświetlany. W trybie edycyjnym wartość właściwości jest ignorowana – obiekt jest zawsze widoczny. |
| Przezroczystość | Liczba <0,1> | Określa stopień przezroczystości obiektu w zakresie od 0 do 1. Wartość 1 oznacza obiekt nieprzezroczysty, wartość 0 oznacza obiekt całkowicie przezroczysty. |

6.3 Standardowe zdarzenia obiektów

| Nazwa | Opis |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Animacja | Zdarzenie uruchamiane okresowe. Przeznaczone dla celów animacji ruchu obiektów. |
| Wysłanie sterowania | Zdarzenie uruchomiane w momencie wykonania akcji operatorskiej <i>SendControls</i> . Dotyczy tylko obiektów aktywnych. |
| Anulowanie sterowania | Zdarzenie uruchomiane w momencie wykonania akcji operatorskiej <i>CancelControls</i> . Dotyczy tylko obiektów aktywnych. |
| Wciśnięcie lewego przycisku | Zdarzenie uruchomiane w momencie wciśnięcie lewego klawisza myszki, gdy kursor znajduje się w obszarze obiektu. |
| Wciśnięcie prawego przycisku | Zdarzenie uruchomiane w momencie wciśnięcie prawego klawisza myszki, gdy kursor znajduje się w obszarze obiektu. |
| Wciśnięcie środkowego przycisku | Zdarzenie uruchomiane w momencie wciśnięcie środkowego klawisza myszki, gdy kursor znajduje się w obszarze obiektu. |
| Zwolnienie lewego przycisku | Zdarzenie uruchomiane w momencie zwolnienia lewego klawisza myszki, gdy kursor znajduje się w obszarze obiektu. |
| Zwolnienie prawego przycisku | Zdarzenie uruchomiane w momencie zwolnienia prawego klawisza myszki, gdy kursor znajduje się w obszarze obiektu. |
| Zwolnienie środkowego przycisku | Zdarzenie uruchomiane w momencie zwolnienia środkowego klawisza myszki, gdy kursor znajduje się w obszarze obiektu. |
| Kliknięcie lewym przyciskiem | Zdarzenie uruchomiane w momencie kliknięcia (przyciśnięcie i zwolnienie) lewym klawiszem myszki, gdy kursor znajduje się w obszarze obiektu. |
| Kliknięcie prawym przyciskiem | Zdarzenie uruchomiane w momencie kliknięcia prawym klawiszem myszki, gdy kursor znajduje się w obszarze obiektu. |
| Kliknięcie środkowym przyciskiem | Zdarzenie uruchomiane w momencie kliknięcia środkowym klawiszem myszki, gdy kursor znajduje się w obszarze obiektu. |
| Podwójne kliknięcie lewym przyciskiem | Zdarzenie uruchomiane w momencie podwójnego kliknięcia lewym klawiszem myszki, gdy kursor znajduje się w obszarze obiektu. |
| Podwójne kliknięcie | Zdarzenie uruchomiane w momencie podwójnego kliknięcia prawym |

| | |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| prawym przyciskiem | klawiszem myszki, gdy kursor znajduje się w obszarze obiektu. |
| Podwójne kliknięcie środkowym przyciskiem | Zdarzenie uruchamiane w momencie podwójnego kliknięcia środkowym klawiszem myszki, gdy kursor znajduje się w obszarze obiektu. |
| Trzymanie lewego przycisku | Zdarzenie uruchamiane okresowo, gdy kursor myszki znajduje się w obszarze obiektu i przyciśnięty jest lewy klawisz myszki. |
| Trzymanie prawego przycisku | Zdarzenie uruchamiane okresowo, gdy kursor myszki znajduje się w obszarze obiektu i przyciśnięty jest prawy klawisz myszki. |
| Trzymanie środkowego przycisku | Zdarzenie uruchamiane okresowo, gdy kursor myszki znajduje się w obszarze obiektu i przyciśnięty jest środkowy klawisz myszki. |
| Najechanie kursorem | Zdarzenie uruchamiane, gdy kursor myszki został przesunięty w obszar obiektu. |
| Wyjechanie kursorem | Zdarzenie uruchamiane, gdy kursor myszki został przesunięty poza obszar obiektu. |
| Ruch kursora | Zdarzenie uruchamiane, gdy kursor myszki przemieszcza się w ramach obszaru obiektu. |
| Wciśnięcie klawisza klawiatury | Zdarzenie uruchamiane, gdy zostanie przyciśnięty przycisk na klawiaturze i obiekt jest wyselekcjonowany. Jaki klawisz został przyciśnięty można sprawdzić przy pomocy funkcji <i>LastKeyPressed</i> . Przykład: <code>Perform (LastKeyPressed()==q, SetVariable (v1, 1), Nothing())</code> Zmienna v1 jest ustawiana na 1, jeżeli został przyciśnięty klawisz q. |
| Wyselekcjonowanie obiektu | Zdarzenie uruchamiane w momencie wyselekcjonowania obiektu. Dotyczy tylko obiektów aktywnych. |

7 Wzorce

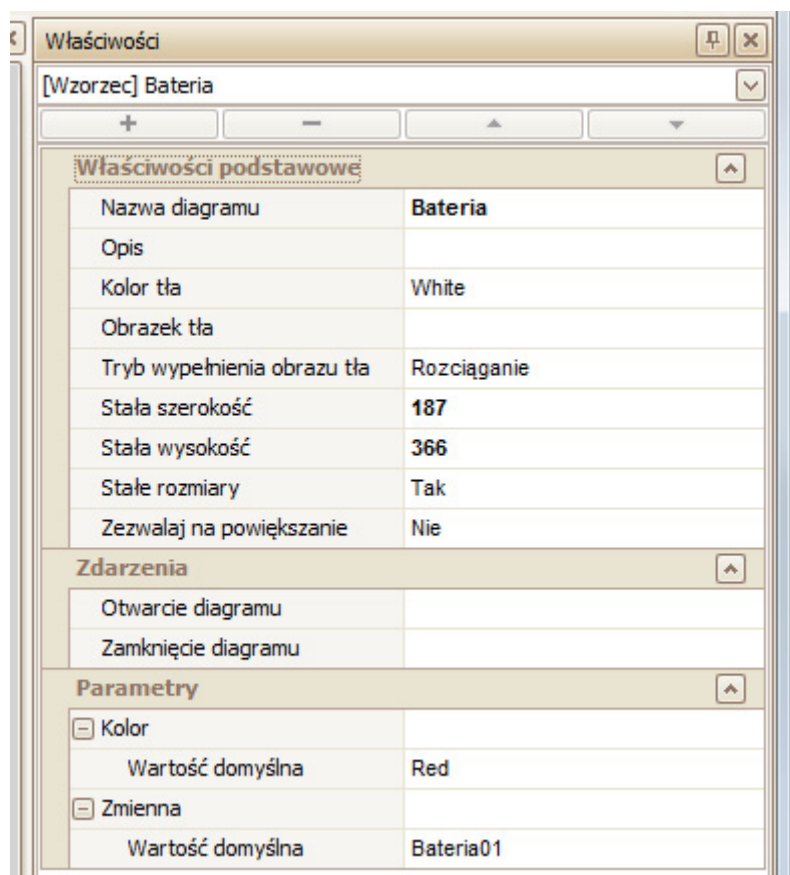
Wzorce to wstępnie sparametryzowane grupy obiektów. Wzorce identyfikowane są przez nazwę. Po utworzeniu wzorzec może być wielokrotnie osadzany na diagramach. Jeżeli definicja dowolnego elementu wzorca ulegnie zmianie, to automatycznie wpływa to na wszystkie diagramy używające wzorca.

Diagram może składać się także z pojedynczego obiektu. Pozwala to przygotować obiekty, które realizują pewien schemat działania. Zmiana tego schematu nie wymaga zmiany diagramów.

Osadzony wzorzec może zostać przekształcony na zwykłe obiekty. Powoduje to utworzenie na diagramie kopii wszystkich obiektów wzorca. Od tego momentu obiekty pochodzące z wzorca traktowane są jak każdy inny obiekt diagramu.

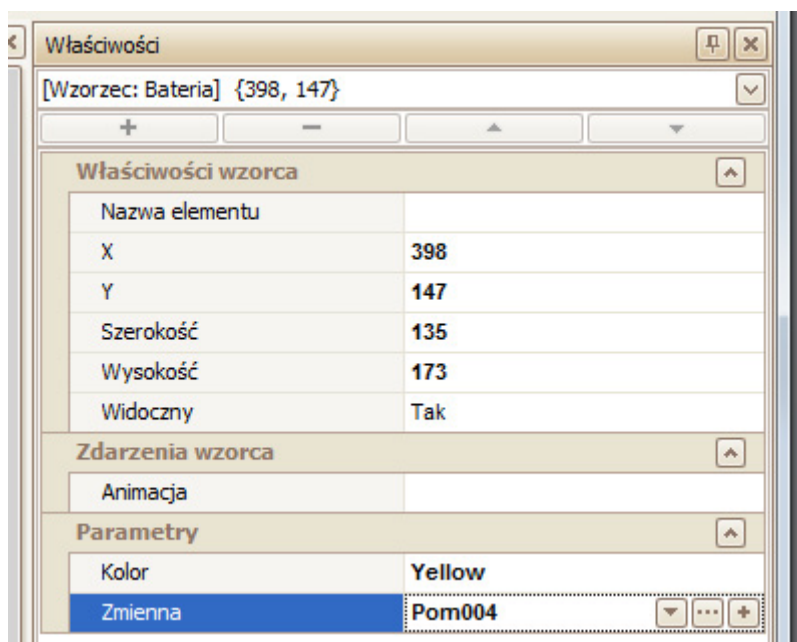
Kluczowym elementem definicji wzorca są parametry. Pozwala to wielokrotnie osadzać wzorzec z podaniem danych, na których wzorzec będzie pracować, w szczególności nazwy zmiennych procesowych.

Na etapie definicji wzorca definiowane są parametry wzorca. Definicja składa się z nazwy parametru oraz jego wartości domyślnej. Wartość domyślna używana jest w trakcie edycji wzorca oraz wtedy, gdy w miejscu osadzenia wzorca nie zostaną podane inne wartości parametru. Po zdefiniowaniu parametru można się do niego odwoływać we właściwościach obiektów składowych wzorca poprzez notację sufiksową *%nazwa* lub wywołanie funkcji *Parameter*.



Rys. Podstawowe właściwości wzorca.

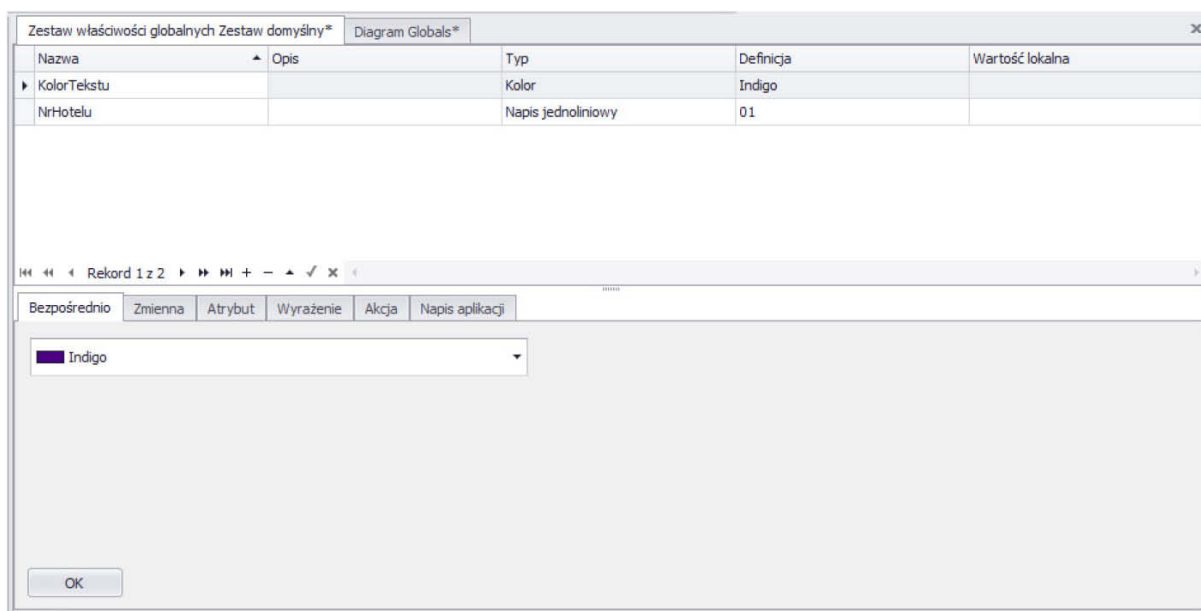
W miejscu osadzenia wzorca należy podać właściwe wartości parametrów.

*Rys. Właściwości wzorca.*

8 Właściwości globalne

Właściwości globalne to centralnie definiowane właściwości niepowiązane z żadnym konkretnym elementem aplikacji. Projektant aplikacji może odwołać się wielokrotnie do właściwości globalnej w definicjach różnych elementów, np. we właściwościach obiektów. Stwarza to możliwość łatwej modyfikacji wyglądu lub zachowania aplikacji bez konieczności zmiany definicji właściwości wielu elementów składowych. Wartości właściwości globalnych mogą być modyfikowane w trybie wykonywania aplikacji – pozwala to na stworzenie w aplikacji mechanizmów pozwalających użytkownikom na zmianę sposobu jej działania.

Właściwości globalne definiowane są w panelu właściwości.



Rys. Panel właściwości globalnych.

Właściwość globalna jest definiowana poprzez jej nazwę, opcjonalny opis, typ właściwości i definicję wartości. Typ ma znaczenie tylko pomocnicze i służy do wyboru odpowiedniego edytora trybu bezpośredniego. Rzeczywista interpretacja definicji wartości właściwości zależy od miejsca jej użycia. Jeżeli przykładowa właściwość *KolorTekstu* zostanie użyta w obiekcie *Tekst* we właściwości *Kolor*, to spowoduje zmianę koloru wyświetlanego napisu. Jeżeli zostanie użyta we właściwości *Tekst*, to spowoduje wyświetlenie tekstu *Indigo*.

Kolumna *Wartość lokalna* jest kolumną tylko do odczytu. Pokazuje ona aktualną wartość właściwości wtedy, gdy została ona zmieniona w trybie wykonywania aplikacji przy pomocy akcji *SetGlobalProperty*.

Odwołania do właściwości globalnej wykonywane są przy pomocy funkcji *Property* lub notacji prefiksowej *!nazwa_właściwości*.

Definicja wartości właściwości najczęściej podawana jest w sposób bezpośredni. Można użyć innych metod, np. odwołań do atrybutów zmiennych czy wyrażeń. W tych przypadkach wyliczenie wartości właściwości odbywa się w miejscu i kontekście jej użycia.

Przykład:

Właściwość *Limit* jest zdefiniowana jako:

`=Attribute(LimitHi)*1.1`

Jeżeli właściwość *Limit* zostanie użyta w obiekcie *Tekst* we właściwości *Tekst*, poprzez odwołanie `!Limit`, to zostanie wyświetlona wartość limitu górnego zmiennej głównej obiektu pomnożona przez 1,1.

Należy zwrócić uwagę, że w tym przypadku użycie wyrażenia `=Property(Limit)` nie jest równoważne. Zostałaby wyświetlona wartość właściwości *Limit* w postaci tekstowej `Attribute(LimitHi)*1.1`. Poprawne użycie to wyrażenie postaci `=Evaluate(Property(Limit))`.

8.1 Zmiana wartości właściwości w trybie wykonywania aplikacji

Dostępne są dwa mechanizmy zmiany właściwości globalnej w trybie wykonywania aplikacji. Pierwszy z nich polega na zmianie definicji właściwości w komendzie startowej programu. Mechanizm ten pozwala np. uzależnić działanie aplikacji od miejsca jej uruchomienia.

Zmiana właściwości w komendzie startowej polega na umieszczeniu w jej treści deklaracji w postaci:

`&nazwa_właściwości=definicja`

Przykład:

`AsixEvo.exe &Department=KJ NrHotelu=04 -as=Terminal -run "c:\asixapp\app1"`

W powyższym przykładzie uruchamiana jest aplikacja na stanowisku *Terminal*. Dodatkowo właściwość globalna *Department* jest ustawiana na wartość *KJ*.

Drugim mechanizmem zmiany wartości właściwości jest wykorzystanie akcji operatorskiej *SetGlobalProperty*. Wszystkie zmiany wartości wykonane przy jej pomocy są zapamiętywane. Po restarcie aplikacji, wartość właściwości zostanie automatycznie ustawiona na podstawie ostatniego wykonania akcji *SetGlobalProperty*. Powrót do oryginalnej wartości właściwości zdefiniowanej w aplikacji wymaga wykonania akcji *ResetGlobalProperty*. Zmiany wykonane akcją *SetGlobalProperty* są ograniczone tylko do stanowiska, na którym akcja została wykonana.