

Opis przetwarzania Diagramów Stanów UML do CPNTools

Wczytywanie UML

1. Odczytanie StateMachine i Regionów

Diagram stanów z pliku UML odczytuje się jako packedElement o typie StateMachine. State Machine posiada regiony, a te z kolei posiadają pola m.in. miejsc, stanów i tranzycji.

Pierwszym krokiem jest pętla po wszystkich elementach w StateMachine, po regionach w nich znajdujących się, a następnie po poszczególnych polach.

```
<packagedElement xmi:type="uml:StateMachine" xmi:id="_7msDoPsiEet4uq1OaFxuW" name="StateMachine1">
  <region xmi:type="uml:Region" xmi:id="_8FKvYPsiEet4uq1OaFxuW" name="Region1">
    <transition xmi:type="uml:Transition" xmi:id="_-0oqEAo7EeiuuqFePlcUhg" name="Trans4" source="_BhPv
    <transition xmi:type="uml:Transition" xmi:id="_swcrMatUEiXC4mW10fKqA" name="Trans1" source="_8qg4
```

Rysunek 1. StateMachine i regiony w UML

2. Odczytanie tranzycji

Każda tranzycja jest oznaczona tagiem „transition” i posiada źródło oraz cel. Dla każdego regionu została utworzona lista tranzycji z odpowiednimi parametrami, nie korzystając z id elementów źródłowych oraz docelowych, lecz z nazw. Wynika to z tego, że każda nazwa w CPN Tools musi być unikatowa;

```
<region xmi:type="uml:Region" xmi:id="_8FKvYPsiEet4uq1OaFxuW" name="Region1">
  <transition xmi:type="uml:Transition" xmi:id="_-0oqEAo7EeiuuqFePlcUhg" name="Trans4" source="_BhPWgPsjEet4uq1OaFxuW" target="_9RuCEPsiEet4uq1OaFxuW"/>
  <transition xmi:type="uml:Transition" xmi:id="_swcrMatUEiXC4mW10fKqA" name="Trans1" source="_8qg4IPsiEet4uq1OaFxuW" target="_BhPWgPsjEet4uq1OaFxuW"/>
```

Rysunek 2. Tranzycje w UML

3. Odczytanie pól Subvertex

Każde pole jest oznaczone Tagiem „subvertex”. Dla każdego regionu została utworzona lista wszystkich pól. Wyjątkiem jest subvertex o typie „State”.

```
<subvertex xmi:type="uml:Pseudostate" xmi:id="_8qg4IPsiEet4uq1OaFxuW" name="Initial1"/>
<subvertex xmi:type="uml:FinalState" xmi:id="_9RuCEPsiEet4uq1OaFxuW" name="FinalState1"/>
<subvertex xmi:type="uml:State" xmi:id="_BhPWgPsjEet4uq1OaFxuW" name="State1">
  <region xmi:type="uml:Region" xmi:id="_CRmmUAEfBeiG_Ymvf7uI1Q" name="Region1">
```

Rysunek 3. Subvertex w UML

4. Odczyt stanów

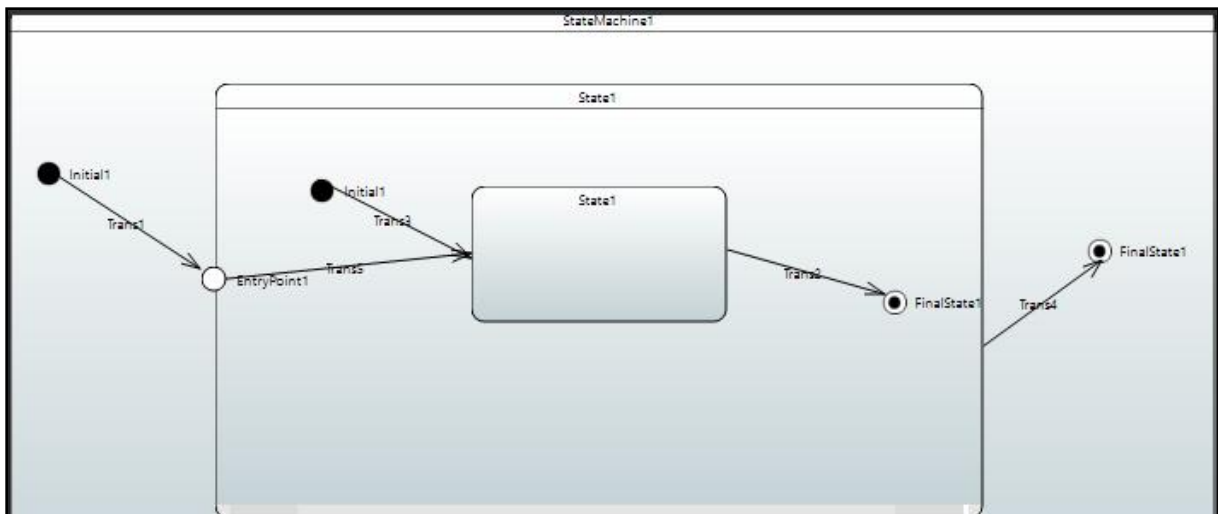
Każdy stan jest utworzony mniej więcej w ten sam sposób co w krokach 1, 2, 3 z tym, że została dodana obsługa pustych stanów oraz wejść i wyjść do danego stanu oznaczonych osobnym tagiem „connectionPoint”, umieszczonych poza regionami.

```
<subvertex xmi:type="uml:State" xmi:id="_BhPWgPsjEet4uq1OaFxiw" name="State1">
<connectionPoint xmi:type="uml:Pseudostate" xmi:id="_rAT4YAtWEiXC4mW10fKqA" name="EntryPoint1" kind="entryPoint"/>
<region xmi:type="uml:Region" xmi:id="_CRmmUAEfEeiG_Ymvf7uI1Q" name="Region1">
  <transition xmi:type="uml:Transition" xmi:id="_iPrsgAEfEeiG_Ymvf7uI1Q" name="Trans2" source="_Ct5gwAEfEeiG_Ymvf7uI1Q" target="_Dvij8AEfEeiG_Ymvf7uI1Q" />
  <transition xmi:type="uml:Transition" xmi:id="_uvlRYAtUEiXC4mW10fKqA" name="Trans3" source="_t91EUAtUEiXC4mW10fKqA" target="_Ct5gwAEfEeiG_Ymvf7uI1Q" />
  <subvertex xmi:type="uml:State" xmi:id="_Ct5gwAEfEeiG_Ymvf7uI1Q" name="State1"/>
  <subvertex xmi:type="uml:FinalState" xmi:id="_Dvij8AEfEeiG_Ymvf7uI1Q" name="FinalState1" visibility="package"/>
  <subvertex xmi:type="uml:Pseudostate" xmi:id="_t91EUAtUEiXC4mW10fKqA" name="Initial1"/>
</region>
```

Rysunek 4. ConnectionPoint w UML

5. Obsługa wyjątku

Okazuje się, że tranzycja z elementu wejściowego („entryPoint”) w stanie do obiektu w środku tego samego stanu, w UML umieszczona jest w stanie nadrzędnym. Przedstawiony problem został pokazany na rysunku (5) oraz (6). Została dodana obsługa tego wyjątku, przenosząc tranzycję ze stanu nadrzędnego do stanu, do którego powinien należeć. Zostało to wykonane podczas odczytywania UML.



Rysunek 5. Diagram Stanów w Eclipse, który zawiera tranzycję z EntryPoint1 do State1

```
<region xmi:type="uml:Region" xmi:id="_8FKvYPsiEet4uq1OaFxiw" name="Region1">
  <transition xmi:type="uml:Transition" xmi:id="_-0oqEAc7EeiunqFePlcUhg" name="Trans4" source="_BhPWgPsjEet4uq1OaFxiw" target="_8qg4IPsiEet4uq1OaFxiw" />
  <transition xmi:type="uml:Transition" xmi:id="_swcrMATUEiXC4mW10fKqA" name="Trans1" source="_8qg4IPsiEet4uq1OaFxiw" target="_9RuCEPsiEet4uq1OaFxiw" />
  <transition xmi:type="uml:Transition" xmi:id="_s2PfQAtWEiXC4mW10fKqA" name="Trans5" source="_rAT4YAtWEiXC4mW10fKqA" target="_Ct5gwAEfEeiG_Ymvf7uI1Q" />
  <subvertex xmi:type="uml:Pseudostate" xmi:id="_8qg4IPsiEet4uq1OaFxiw" name="Initial1"/>
  <subvertex xmi:type="uml:FinalState" xmi:id="_9RuCEPsiEet4uq1OaFxiw" name="FinalState1"/>
  <subvertex xmi:type="uml:State" xmi:id="_BhPWgPsjEet4uq1OaFxiw" name="State1">
    <connectionPoint xmi:type="uml:Pseudostate" xmi:id="_rAT4YAtWEiXC4mW10fKqA" name="EntryPoint1" kind="entryPoint"/>
    <region xmi:type="uml:Region" xmi:id="_CRmmUAEfEeiG_Ymvf7uI1Q" name="Region1">
      <transition xmi:type="uml:Transition" xmi:id="_iPrsgAEfEeiG_Ymvf7uI1Q" name="Trans2" source="_Ct5gwAEfEeiG_Ymvf7uI1Q" target="_Dvij8AEfEeiG_Ymvf7uI1Q" />
      <transition xmi:type="uml:Transition" xmi:id="_uvlRYAtUEiXC4mW10fKqA" name="Trans3" source="_t91EUAtUEiXC4mW10fKqA" target="_Ct5gwAEfEeiG_Ymvf7uI1Q" />
      <subvertex xmi:type="uml:State" xmi:id="_Ct5gwAEfEeiG_Ymvf7uI1Q" name="State1"/>
      <subvertex xmi:type="uml:FinalState" xmi:id="_Dvij8AEfEeiG_Ymvf7uI1Q" name="FinalState1" visibility="package"/>
      <subvertex xmi:type="uml:Pseudostate" xmi:id="_t91EUAtUEiXC4mW10fKqA" name="Initial1"/>
    </region>
  </subvertex>
</region>
```

Rysunek 6. Tranzycja Trans5 z Rysunku 1 znajduje się w stanie nadrzędnym w dokumencie UML.

Przetwarzanie otrzymanej struktury z UML do CPN Tools

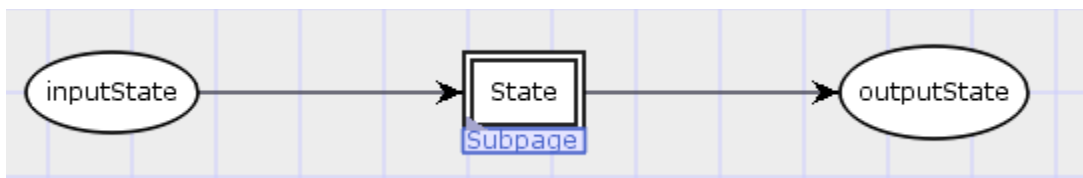
1. Wczytanie szablonu pustego projektu w CPN Tools.
2. Przetworzenie otrzymanej struktury wczytanej z UML

3. **Dodanie miejsc**

W pierwszej kolejności do dokumentu CPN dodawane są miejsca dla każdego typu „subvertex” wykorzystując klasę PlaceCreator. Wyjątkiem jest „state”, który nie powinien być miejscem lecz tranzycją/podgrupą w CPN Tools:

- 1) **Dodanie wejść i wyjść do stanu**

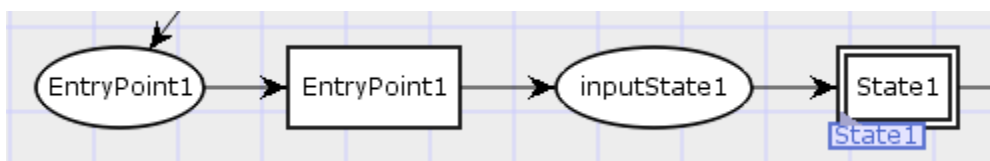
Każdy podgrupa w CPN Tools musi posiadać wejście („Input”) i wyjście („Output”). Biorąc pod uwagę to, że zostało założone, że każdy Stan jest tranzycją, należy w stanie nadrzędnym dostarczyć mu miejsca będące wejściem i wyjściem. Zostały one oznaczone nazwą „input <state_name>” oraz „output <state_name>”



Rysunek 7. Utworzone miejsca inputState i outputState

- 2) **Dodanie EntryPoints**

W UML mogą być wejścia i wyjścia do stanów nie jako Initial czy FinalState tylko jako Entry Points lub Exit Point. Dla każdego z tych elementów w stanie nadrzędnym również zostało utworzone miejsce.



Rysunek 8. Utworzone miejsce EntryPoint

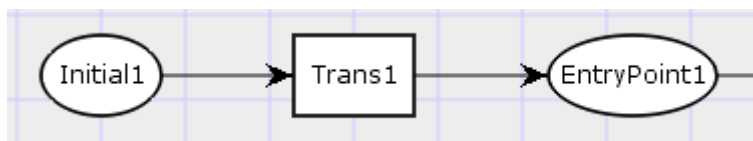
4. **Dodanie tranzycji**

Kolejnym krokiem jest dodanie tranzycji do dokumentu CPN korzystając z klasy TransCreator.

- 1) **Dodanie tranzycji z diagramu UML**

Najpierw odbywa się dodanie tranzycji dla każdej tranzycji pochodzącej z UML. Następnie dodane zostają połączenia (tzw. „arc” w CPN Tools) pomiędzy tranzycjami. W UML obszary stanów są podzielone na regiony, ale w CPN nie ma takiego podziału. Regiony zostały więc obsłużone w ten sposób, aby nie miało znaczenia gdzie znajduje się dany obiekt. Podczas dodawania połączeń pomiędzy

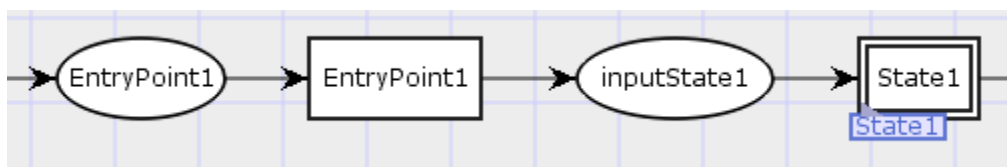
obiektami należało wziąć pod uwagę, że w diagramie UML połączone obiekty mogą znajdować się w różnych regionach.



Rysunek 9. Przykładowa tranzycja

2) Dodanie tranzycji dla EntryPoint oraz Tranzycja Stanu

Następnie odbywa się dodanie tranzycji dla każdego stanu. Najpierw dodawany jest sam obiekt tranzycji, a następnie połączenia, pomiędzy wejściem do stanu i wyjściem ze stanu (opisane w punkcie 3). Stan również może mieć połączenia „ConnectionPoint” więc jeżeli takie istnieją to są tworzone tranzycje w stanie nadrzędnym z punktów połączeń do wejścia lub wyjścia ze stanu. Kolejnym krokiem jest tworzenie podstanu.



Rysunek 10. Tranzycja EntryPoint1 połączona z wejściem do stanu, a miejscem EntryPoint1

5. Dodanie Stanu

Dodawanie podstanu obsługane zostało jako stworzenie podgrupy w CPN Tools. Składa się ono z następujących kroków:

1) Utworzenie nowej strony

W pierwszej kolejności stworzona jest nowa strona w CPN Tools. Została utworzona i wykorzystana klasa PageCreator, która dodaje stronę do dokumentu CPN Tools pod tagiem „cpnnet”.

```
<page id="ID1412977915">
  <pageattr name="State1"/>
  <constraints/>
```

Rysunek 11. Strona w CPN

2) Dodanie instancji

Samo dodanie strony powoduje błędy podczas uruchomienia projektu CPN. Najpierw należało utworzyć instancję strony w dokumencie CPN pod tagiem „instances”, w odpowiedniej instancji strony nadrzędnej. Nowa podstrona nie posiada już domyślnych atrybutów instancji czyli „id” oraz „page” lecz „id” oraz „trans” czyli tranzycję, pod którą została utworzona podstrona.



Rysunek 12. Dodanie instancji

3) Dodanie Cpn sheet

Po utworzeniu instancji należy umieścić ją w CPN Sheets pod tagiem „sheets”, a następnie pod „cpnsheets” w dokumencie CPN,

```
<cpnsheet id="ID1412977917" instance="ID1412977916" panx="0.000000" pany="0.000000" zoom="1.000000">
  <zorder>
    <position value="0"/>
  </zorder>
</cpnsheet>
```

Rysunek 13. Cpn Sheet

4) Dodanie wejścia i wyjścia ze stanu

Po utworzeniu podstrony, instancji oraz CPN Sheets można przejść do tworzenia wewnętrznego diagramu stanów. Należało rozpocząć od utworzenia miejsca wejściowego oraz wyjściowego, który został nazwany „input” oraz „output”. Nazwa ta jest w związku z tym zarezerwowana i stworzenie innych elementów w diagramie UML o takich nazwach spowoduje błędy w postaci złego połączenia obiektów w CPN.



Rysunek 14. Wejście i wyjście ze stanu

5) Dodanie portu

Dla wejścia oraz wyjścia został dodany również niezbędny Tag o nazwie „port”, który wskazuje, że jest to miejsce wyjściowe lub wejściowe z danej podgrupy.

```
<port id="ID1412977874" type="In"/>
```

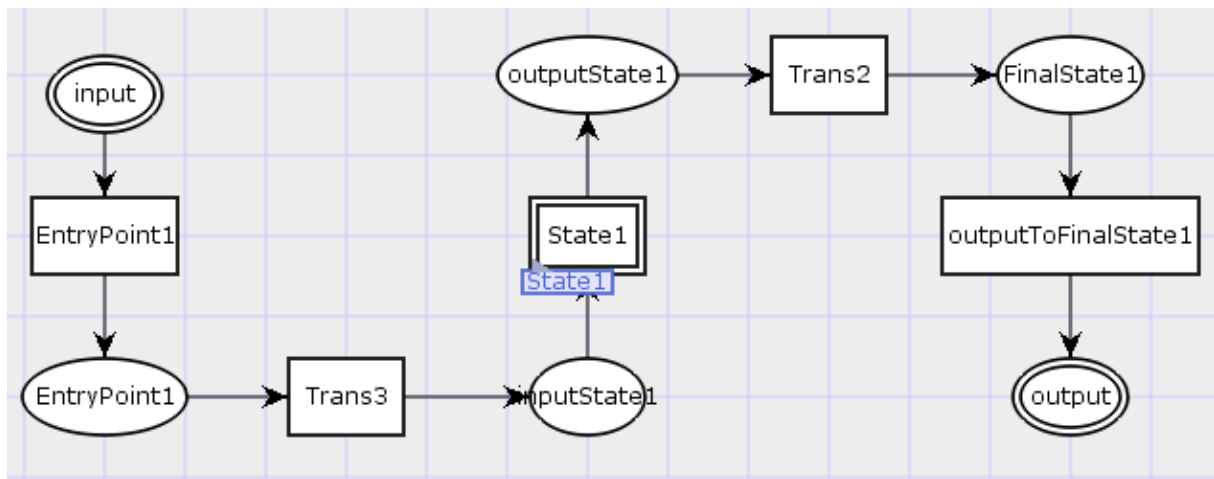
Rysunek 15. Tag port w miejscu w dokumencie cpn

6) Dodanie obiektów w stanie

Następnie został utworzony diagram stanów czyli:

- Utworzenie miejsc, również dla połączeń wejściowych i wyjściowych do stanu nadrzędnego
- Utworzenie tranzycji z Tranzycji pochodzących z UML

- Utworzenie tranzycji dla Stanów wraz z ewentualnymi połączeniami wejściowymi do nich i wyjściowymi z nich
- Utworzenie tranzycji od wejścia do stanu („input”), do wszystkich dostępnych punktów połączeń lub stanów początkowych („initial”)
- Utworzenie tranzycji do wyjścia ze stanu („output”), od wszystkich dostępnych punktów połączeń lub stanów końcowych („FinalState”)
- Utworzenie połączeń pomiędzy wszystkimi tranzycjami i miejscami
- Utworzenie podgrup stanów, co zostało zrealizowane poprzez rekurencyjne wywołanie funkcji odpowiedzialnej za wykonanie punktu 5, aż do momentu gdy nic nie znajdzie się w środku stanu podrzędnego.
- W przypadku gdy już nic się nie znajdzie w środku stanu, zostaje utworzona jedna tranzycja w środku stanu pomiędzy miejscami „input” i „output”, aby została zachowana ciągłość diagramu.



Rysunek 16. Przykładowe utworzenie diagramu stanu podrzędnego

7) Dodanie PortSock

Przed ostatnim krokiem została dodana informacja o podstronie w tranzycji, w którym znajduje się stan, w stanie nadrzędnym, poprzez dodanie Tagu „subst” do tranzycji, a w nim atrybut „id” oraz „portsock”, który posiada następującą strukturę:

(<id_miejsca_input_w_podstronie>,<id_miejsca_input_ze_strony_nadrzędnej>)(<id_miejsca_output_w_podstronie>,<id_miejsca_output_ze_strony_nadrzędnej>)

```
<subst portsock="(ID1412977918,ID1412977876) (ID1412977921,ID1412977879)" subpage="ID1412977915"/>|
</trans>
```

Rysunek 17. Portsock w tranzycji

Nowe klasy i funkcjonalności

Nowe typy obiektów umieszczone w parser.Entities :

- TransitionType
- SubvertexType
- StateType
- StateMachineType
- RegionType
- ConnectinoPointType

Nowe klasy do tworzenia obiektów pod parser.CPN.CPNCreators:

- PageCreator
- TransCreator (znaczne rozszerzenie o nowe funkcjonalności)
- PlaceCreator (znaczne rozszerzenie o nowe funkcjonalności)
- ArcCreator (rozszerzenie o nowe funkcjonalności)

Dodano diagram UML do projektu:

- State_machine.di
- State_machine.uml
- State_machine.notation

Inne rozszerzone klasy:

- UMLReader
- CPNParser