

pracę z modułem procesowym, początkujący użytkownik może napotkać trudności związane z kolejnością łączenia poszczególnych poleceń w dłuższe sekwencje zadań logicznych. Zaprezentowane w dalszej części przykłady pozwolą wypracować właściwe podejście, które przyspieszy pracę nad budową bardziej skomplikowanych i wymagających modeli.

Szczegółowy opis symboli dostępnych z modułu procesowego można znaleźć w instrukcji obsługi programu. Należy pamiętać, że w przypadku użycia modułu procesowego razem z modelem 3D moduł procesowy jest zawsze nadrzędny nad modelem. Moduł procesowy pracuje na wyższym poziomie i przejmuje kontrolę nad modelem, który najczęściej służy jedynie do wizualizacji badanego procesu. W celu lepszego poznania działania ogólnego modułu procesowego przeanalizowany zostanie od podstaw schemat blokowy realizujący najczęściej stosowane polecenia.

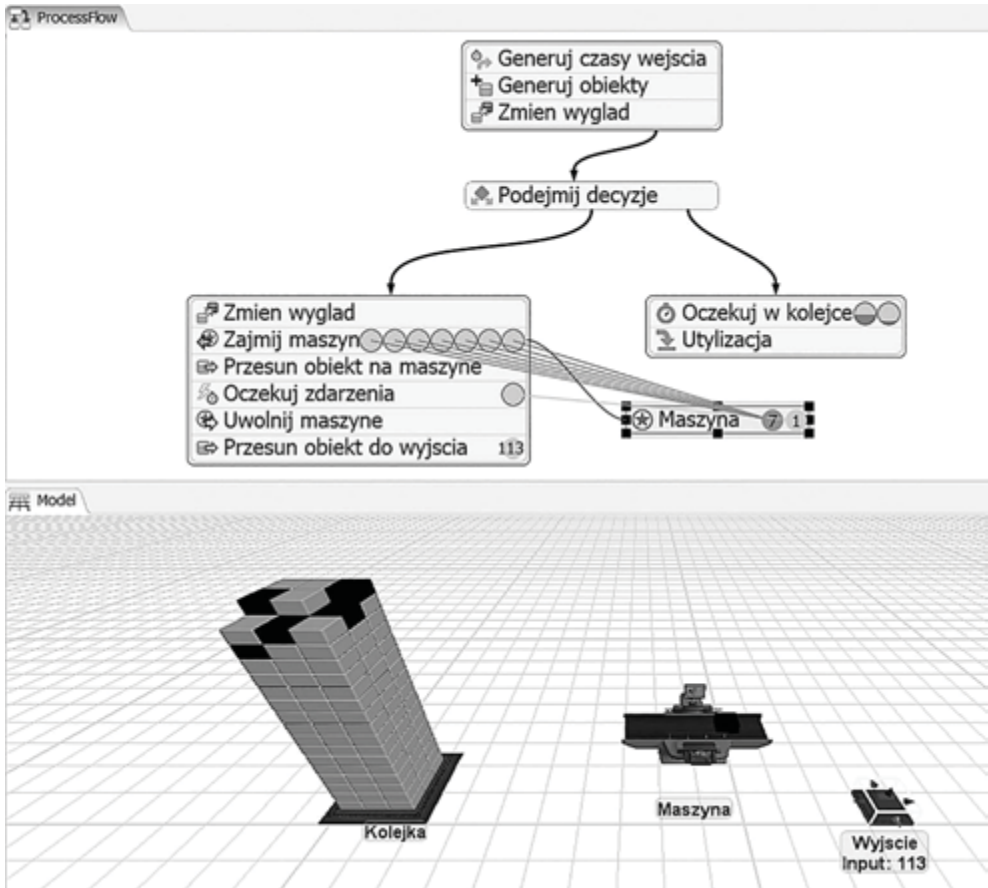
### **Charakterystyka podstawowych operacji służących do sterowania modelem**

Pierwsze zadanie polegać będzie na zapoznaniu się ze sposobem zarządzania modelem symulacyjnym z poziomu modułu procesowego. Głównym celem tego podrozdziału jest poznanie filozofii pracy z modułem procesowym. Ważnym celem tego przykładu jest poznanie zasady działania mechanizmu przejmowania kontroli i uwalniania zasobów współdzielonych, a także poznanie sposobu podejmowania decyzji w systemie. Należy pamiętać, że token jest pojęciem szerszym niż element przepływu, ale w tym zadaniu jest on z nim powiązany, więc reprezentuje tradycyjny element przepływu typu *Box*, jaki jest już znany czytelnikowi z poprzednich modeli 3D.

**Zadanie.** Należy zbudować schemat blokowy modelu procesu realizującego poniższe zadania oraz połączyć jego działanie z modelem 3D, który wizualizuje poszczególne operacje takie jak:

- a) generowanie dwóch typów tokenów (zdarzeń);
- b) powiązanie tokena z fizycznie obrabianym na linii detalem;
- c) przekazywanie tych elementów do wspólnej kolejki;
- d) zmienianie koloru detalu na linii produkcyjnej;
- e) przekazywanie detali za pomocą bloku decyzyjnego do maszyny lub pozostawianie ich w kolejce w stanie oczekiwania w zależności od typu elementu przepływu;
- f) ustalanie czasu oczekiwania elementu przepływu w kolejce z zastosowaniem bloku opóźnienia.

**Rozwiązanie.** Na rysunku 121 przedstawiono propozycję rozwiązania. Budując klasyczny model symulacyjny, który miałby wykonywać powyższe operacje od a) do f), należałoby użyć czterech elementów połączonych w szereg tj.: źródła, kolejki, maszyny, wyjścia z systemu. W przypadku sterowania z poziomu modułu procesowego cała logika systemu jest przedstawiona na schemacie blokowym, a do wizualizacji jego działania są potrzebne tylko trzy elementy: kolejka, maszyna i wyjście. W klasycznym podejściu model 3D w ogóle nie powinien działać, a jednak działa, ponieważ kontrolę nad nim przejmuje moduł procesowy. Nie przeszkadza brak połączeń między portami przepływu. Dla omawianego schematu blokowego model 3D jest jedynie wizualizacją procesu, który przebiega w module procesowym zgodnie z logiką schematu blokowego w górnej części rys. 121.



Rysunek 121

Omawianie działania schematu zaprojektowanego w module procesowym należy rozpocząć od pierwszego górnego bloku, który jest odpowiedzialny za generowanie zdarzeń. Użyto słowa zdarzenie jako ogólnego pojęcia, ponieważ może odnosić się ono do modelowania dowolnych zdarzeń. W klasycznym modelu były to elementy przepływu zwane flowitemami lub itemami, a w module procesowym to abstrakcyjne obiekty nazwane tokenami. Pierwszy blok schematu zawiera następujące aktywności tj.

1. *Generuj czasy wejścia.*
2. *Generuj obiekty.*
3. *Zmień wygląd.*

Rozwinięcie poszczególnych opcji znajduje się po prawej stronie obszaru roboczego w menu *Quick Properties*, rys. 122. W pierwszej aktywności *Generuj czasy wejścia* za częstość wejść zdarzeń odpowiada pole *Inter-Arrivaltime*, w którym wpisano rozkład wykładniczy z parametrem skali równym 5 s. Pole *Token Name* jest nieobowiązkowe. Nazwę *Produkty* wprowadzono tam jedynie w celu lepszego rozróżnienia. W drugiej aktywności *Generuj obiekty* dla reprezentacji wizualnej wybrano grupę elementów przepływu typu *Box*. Mają