Symulacja i Sterowanie Procesów Dyskretnych

Uogólniony model procesu produkcyjnego

Andrzej Brodzicki

Aleksander Pasiut

Michał Trojnarski

Mateusz Wąsala

Automatyka i Robotyka

5 kwietnia 2017

Spis treści

[1. Cel projektu 3](#_Toc483697835)

[2. Opis problemu 3](#_Toc483697836)

[3. Struktury danych 3](#_Toc483697837)

[3.1 Produkt 3](#_Toc483697838)

[3.2 Wpis w magazynie 3](#_Toc483697839)

[3.3 Parametry przetwarzania 4](#_Toc483697840)

[4. Klasy obiektów 5](#_Toc483697841)

[4.1 Stanowisko 5](#_Toc483697842)

[5. Koordynator zdarzeniowy 6](#_Toc483697843)

# 1. Cel projektu

Celem projektu jest uogólnione zamodelowanie procesu produkcyjnego uwzględniające przepływ i przetwarzanie elementów, a także analiza wydajności procesu w zależności od jego parametrów i zastosowanego sterowania.

# 2. Opis problemu

Uogólnione zamodelowanie procesu produkcyjnego oznacza stworzenie takiego modelu, który mógłby być łatwo przystosowany do opisania określonego procesu produkcyjnego. W szczególnym, uproszczonym przypadku, proces produkcyjny reprezentowany jest przez pojedynczą linię produkcyjną, gdzie półprodukty są kolejno przekazywane między stanowiskami, aż do osiągnięcia ostatniego stanowiska, które jest opuszczane przez gotowe produkty. W ogólnym przypadku proces produkcyjny nie jest jednak linią, a siecią, która łączy ze sobą wiele stanowisk na różne sposoby, gdzie półprodukty są przekazywane w znacznie bardziej skomplikowany sposób.

W celu dostosowania się do powyższego wymagania, podstawowym obiektem, z którego składa się model jest Stanowisko. W zależności od parametrów, obiekt ten może reprezentować stanowisko produkcyjne, stanowisko kontrolne lub magazyn. Stanowiska są między sobą połączone tworząc sieć zależności.

# 3. Struktury danych

## 3.1 Produkt

Reprezentuje typ produktu / półproduktu, czyli dowolnego materiału / elementu / składnika przekazywanego między stanowiskami. Składa się z dwóch pól: unikalnego identyfikatora i nazwy.

struct Product

{

int id;

string name;

};

## 3.2 Wpis w magazynie

Reprezentuje stan produktu w magazynie wewnętrznym lub zewnętrznym stanowiska. Składa się z pól: typ produktu (struktura danych: Produkt), aktualna ilość, maksymalna ilość, górny limit ostrzegawczy, dolny limit ostrzegawczy, skojarzone Stanowisko, czas wezwania, flaga wezwania i czas dostawy.

Dla wpisu w magazynie wejściowym skojarzone Stanowisko oznacza Stanowisko, z którego obiekty są pobierane, czas wezwania to czas Wezwania Dostawy, a czas dostawy jest nieużywany (równy 0).

Dla wpisu w magazynie wyjściowym skojarzone Stanowisko oznacza Stanowisko, do którego obiekty są wysyłane, czas wezwania to czas Wezwania Odbioru, a czas dostawy to czas wykonania dostawy.

W obu przypadkach flaga wezwania jest ustawiana po wysłaniu wezwania i resetowana przy otrzymaniu odpowiedzi (w postaci wezwania lub dostawy).

struct Warehouse\_Register

{

Station Reg\_station;

Product Reg\_product;

int Actual\_amount;

int Max\_amount;

int Min\_amount;

int Upper\_warning;

int Lower\_warning;

double Call\_time;

double Delivery\_time;

bool Call\_flag;

};

## 3.3 Parametry przetwarzania

Reprezentuje parametry przetwarzania dla danego stanowiska. Struktura ta składa się listy par: typ produktu (struktura danych: Produkt) i ilość pochłaniana w pojedynczym cyklu produkcyjnym, listy par: typ produktu (struktura danych: Produkt) i ilość wytwarzana w pojedynczym cyklu produkcyjnym oraz z wartości oznaczającej czas wykonywania pojedyczego cyklu produkcyjnego.

struct Product\_Amounts

{

Product product;

int process\_value;

};

struct Product\_Parameters

{

List<Product\_Amounts> Input\_pairs;

List<Product\_Amounts> Output\_pairs;

double cycle\_time;

};

# 4. Klasy obiektów

## 4.1 Stanowisko

Podstawową klasą obiektu jest Stanowisko. W zalezności od parametrów, obiekt ten może reprezentować stanowisko produkcyjne, stanowisko kontrolne lub magazyn. Składa się ono z trzech podstawowych elementów:

* wewnętrznego magazynu wejściowego,
* wewnętrznego magazynu wyjściowego,
* stanowiska właściwego.

Obecność powyższych elementów jest częściowo opcjonalna:

* jeżeli Stanowisko reprezentuje magazyn wydający obiekty, to zawiera ono tylko element: wewnętrzny magazyn wyjściowy,
* jeżeli Stanowisko reprezentuje magazyn przyjmujący obiekty, to zawiera ono tylko element: wewnętrzny magazyn wejściowy,
* jeżeli Stanowisko reprezentuje stanowisko produkcyjne lub kontrolne, to zawiera ono wszystkie podstawowe elementy i jego rolą jest wytwarzanie elementów, które są umieszczane w wewnętrznym magazynie wyjściowym, przy pomocy elementów pobieranych z wewnętrznego magazynu wejściowego.

Niezależnie od spełnianej roli stanowisko składa się z następujących pól:

* lista Wpisów w magazynie reprezentująca wewnętrzny magazyn wejściowy (rozmiar listy równy 0 oznacza jej brak)
* lista Wpisów w magazynie reprezentująca wewnętrzny magazyn wyjściowy, (rozmiar listy równy 0 oznacza jej brak)
* Parametry przetwarzania, opisujące zachowanie Stanowiska, w przypadku, gdy reprezentuje ono stanowisko produkcyjne lub kontrolne.

Dodatkowo, obiekt klasy Stanowisko posiada jeszcze dwie flagi: aktywności, która jest ustawiona, jeżeli Stanowisko jest aktywne (w przeciwnym wypadku Stanowisko nie działa) i przetwarzania, która jest ustawiana na czas wykonywania przez Stanowisko czynności przetwarzania.

class Station

{

List<Warehouse\_Register> input\_warehouse;

List<Warehouse\_Register> output\_warehouse;

Process\_Parameters station\_parameters;

bool Enabled;

bool Started;

};

Komunikacja między Stanowiskami odbywa się na dwa sposoby niezależnie od tego co każde z dwóch Stanowisk konkretnie reprezentuje. Pierwszy z nich to Wezwanie-Dostawa (DC-DA), a drugi to Wezwanie-Wezwanie-Dostawa (CC-DC-DA).

Komunikacja typu Wezwanie-Dostawa:

* Stanowisko A wzywa Stanowisko B i ustawia flagę wezwania
* po odpowiednim czasie wezwanie dociera do B
* B wysyła A żądaną liczbę elementów lub maksymalną możliwą jeżeli żądanie przekracza mozliwości B
* po odpowiednim czasie dostawa dociera do A
* A resetuje flagę wezwania

Komunikacja typu Wezwanie-Wezwanie-Dostawa:

* Stanwisko B wzywa A i ustawia flagę wezwania
* po odpowiednim czasie wezwanie dociera do A
* Stanowisko A wzywa Stanowisko B i ustawia flagę wezwania
* po odpowiednim czasie wezwanie dociera do B
* B wysyła A żądaną liczbę elementów lub maksymalną możliwą jeżeli żądanie przekracza mozliwości B
* B resetuje flagę wezwania
* po odpowiednim czasie dostawa dociera do A
* A resetuje flagę wezwania

W tym miejscu można umieścić kompletny diagram reprezentujący na jakiej zasadzie działa Stanowisko (de facto będzie to reverse engineering: z kodu c# na diagram) :D

# 5. Koordynator zdarzeniowy

W zaprezentowanym przez nas modelu kontrolowaniem i sterowaniem przebiegiem symulacji zajmuje się koordynator zdarzeniowy. Zgodnie z podejściem zdarzeniowym, koordynator ma dostęp do dwóch list: listy zdarzeń bieżących i listy zdarzeń przyszłych. Zdarzenia z listy zdarzeń przyszłych są przerzucane zgodnie z chronologią wynikającą z czasu symulacji do listy zdarzeń bieżących i wykonywane.

W opisywanym modelu dostępne są różne rodzaje zdarzeń, które jednak zawsze odnoszą się do określonego Stanowiska. Te zdarzenia to:

* Aktywacja - zdarzenie sygnalizujące uruchomienie Stanowiska; jest ono ściśle powiązane z flagą aktywności Stanowiska; jest także niezbędne na początku symulacji, w celu zapoczątkowania działania wszystkich Stanowisk;
* Dezaktywacja - zdarzenie sygnalizujące wyłączenie Stanowiska;
* Zakończenie Cyklu Produkcyjnego - zdarzenie, które jest dodawane do listy zdarzeń przyszłych w momencie rozpoczęcia cyklu produkcyjnego przez dane Stanowisko;
* Rozpoczęcie Cyklu Produkcyjnego - zdarzenie, które jest dodawane do listy zdarzeń przyzłych w momencie zakończenia cyklu produkcyjnego przez dane Stanowisko;
* Wezwanie Odbioru - zdarzenie, które jest odpowiedzią na ostrzegawczo dużą liczbę elementów w magazynie wyjściowym; sygnalizuje prośbę do Stanowiska odbiorcy o wezwanie dostawy;
* Wezwanie Dostawy - zdarzenie, które jest odpowiedzią na ostrzegawczo małą liczbę elementów w magazynie wejściowym lub na Wezwanie Odbioru; sygnalizuje prośbę do Stanowiska dostawcy o przysłanie dostawy;
* Dostawa - zdarzenie, które jest odpowiedzią na Wezwanie Dostawy; rerezentuje ono przybycie elementów / materiałów do magazynu wejściowego odpowiedniego stanowiska;

Każde ze zdarzeń ma następujące pola:

* identyfikator - niezbędny do rozróżnienia typu zdarzenia;
* Stanowisko nadawcę - zawiera Stanowisko, które jest źródłem tego zdarzenia lub nic, jeżeli zdarzenie zostało dodane do listy z zewnątrz jako część symulacji;
* Stanowisko adresata - zawiera Stanowisko, do którego odnosi się zdarzenie;
* czas zajścia zdarzenia - czas, w którym zdarzenie ma miejsce;

Odpowiednie zdarzenia oprócz powyższych pól zawierają jeszcze dodatkowe, które opisują szczegóły zdarzenia.

Najprościej powyższą koncepcję chyba będzie zaimplementować poprzez stworzenie klasy Zdarzenie z tymi 4 polami i zrobić dziedziczące po niej klasy, które będą zawierać ewentualnie dodatkowe pola.