

NetSim - Symulacja sieci (C++)

W ramach tego zadania zaprojektujesz system służący do modelowania i symulacji działania sieci – na przykładzie “linii produkcyjnych” w fabryce. Następnie wykorzystasz poznaną wiedzę z zakresu technik programowania obiektowego w C++, aby zaimplementować zaprojektowany system.

Model

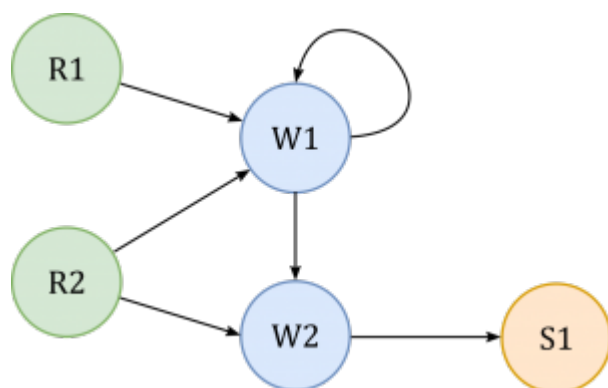
Linia produkcyjna w fabryce (**sieć**, ang. net) składa się z następujących elementów (**węzłów**, ang. nodes), połączonych w spójną całość:

- rampy rozładunkowe (ang. loading ramps) – dostarczają półprodukty do fabryki (stanowią **źródła** sieci, ang. sources)
- robotnicy (ang. workers) – przetwarzają półprodukty
- magazyny (ang. storehouses) – przechowują półprodukty po zakończeniu całego procesu technologicznego (stanowią **ujścia** sieci, ang. sinks)

Dopuszczalne są następujące **połączenia** (ang. links) w ramach procesu technologicznego:

- rampa rozładunkowa → robotnik (tj. dostarczenie półproduktów do pierwszego etapu produkcji)
- robotnik → robotnik (tj. dostarczenie półproduktu do dalszej obróbki; w szczególności dopuszczamy sytuację, gdy ten sam robotnik przekazuje efekt swojej pracy samemu sobie – do dalszej obróbki)
- robotnik → magazyn (tj. umieszczenie końcowego półproduktu w magazynie)

Każda rampa i każdy robotnik może przekazywać półprodukty tylko do pewnego podzbioru wszystkich możliwych odbiorców (robotników i magazynów) – podzbiór ten zdefiniowany jest indywidualnie dla każdej rampy/robotnika. W poniższym przykładzie dopuszczalni odbiorcy węzłów zostali oznaczeni strzałkami.



Przykładowa sieć (R – rampa, W – robotnik, S – magazyn)

Dodatkowe założenia:

- [P] Każdy półprodukt dostarczany do fabryki ma swój unikalny numer seryjny, który nie zmienia się pomiędzy etapami produkcji (numer ten to liczba całkowita dodatnia).
- [N] Dostawy półproduktów do ramp odbywają się z określoną częstotliwością (np. co drugą turę symulacji), począwszy od pierwszej tury. Jednorazowo dostarczana jest jedna jednostka półproduktu.

- [N] Robotnik przetwarza na raz (w danej turze) pojedynczą jednostkę półproduktu, przy czym każdemu robotnikowi proces ten zajmuje określoną liczbę tur (może się ona różnić pomiędzy robotnikami, lecz zawsze wynosi co najmniej jedną turę).
- [P, N] Każdy robotnik posiada swoje preferencje odnośnie kolejności przetwarzania dostarczanych mu półproduktów – np. jedni wolą przetwarzać je w kolejności napływania (kolejka *first in, first out*, **FIFO**), inni wolą zaczynać od półproduktu dostarczonego jako ostatni (kolejka *last in, first out*, **LIFO**, *stos*). Na chwilę obecną robotnicy przetwarzają produkty wyłącznie w trybie albo FIFO albo LIFO, jednak w przyszłości na pewno zechcą opracować nowe schematy organizacji swej pracy...
- [N] Robotnik nie przerywa pracy nad aktualnie przetwarzanym produktem, gdy napływają do niego nowe półprodukty – te są po prostu wstawiane do kolejki i oczekują na zakończenie wykonywania przez robotnika aktualnego zadania.
- [N] Na chwilę obecną każda rampa/robotnik przekazuje przetworzony półprodukt jednemu z przypisanych mu odbiorców nie faworyzując żadnego z nich (tj. odbiorca jest wybierany z puli n odbiorców z jednakowym prawdopodobieństwem – $1/n$). Jednak docelowo każda rampa/robotnik będzie mieć możliwość posiadania własnych preferencji w tym względzie (np. będzie przekazywać półprodukty robotnikowi A dwa razy częściej niż robotnikowi B) – należy to wymaganie uwzględnić na etapie projektowania, stosując odpowiednią strukturę danych do przechowywania informacji o preferencjach wyboru odbiorców.
- [N] Każdy magazyn ma możliwość przechowywania produktów w sposób niezależny od innych magazynów (tj. wewnętrzna realizacja funkcjonalności “kontenera półproduktów” może się różnić pomiędzy magazynami).

Symulacja

Zadania: N, F, S

Dla zadanego modelu sieci (wczytanego z pliku) chcemy mieć możliwość zasymulowania jej działania – np. aby sprawdzić, jaka jest potencjalna wydajność linii produkcyjnej.

Jednostką czasu w symulacji jest **tura** (ang. round). Każda tura składa się z **etapów** (ang. stages), rozpatrywanych w poniższej kolejności:

1. **Dostawa** półproduktów do ramp
Półprodukty “pojawiają się” na rampie.
2. **Przekazanie** półproduktów do odbiorców.
Półprodukt jest momentalnie przekazywany od nadawcy do odbiorcy – nie znajduje się nigdy “w drodze”.
3. **Przetworzenie** półproduktów przez robotników.
4. **Raportowanie** o stanie symulacji (jeśli dotyczy danej tury)

Przyjmij, że w ramach każdego etapu symulacji węzły rozpatrywane są sekwencyjnie, zgodnie z rosnącym ID (czyli np. najpierw pracę wykonuje robotnik #1, potem robotnik #2 itd.) – dzięki temu nie ma potrzeby korzystania z mechanizmu **wielowątkowości**.

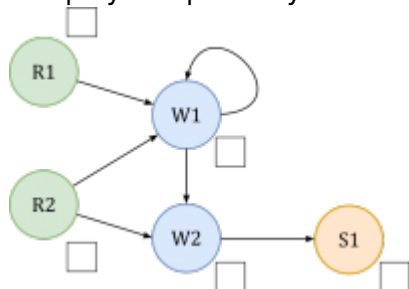
Czas symulacji zaczyna się od 1 (tj. pierwsza tura ma identyfikator 1).

Powinna istnieć możliwość generowania raportu o stanie symulacji w następujących trybach:

- dla zadanych tur (np. dla trzeciej i piątej tury, dla ostatniej tury)
- w zadanych odstępach czasu (np. co drugą turę)

Przykład

Oto przykład pierwszych dwóch tur symulacji dla poniższej sieci:

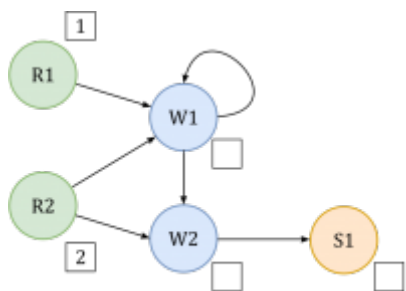


(Ramki oznaczają kolejki półproduktów związane z danymi węzłami.)

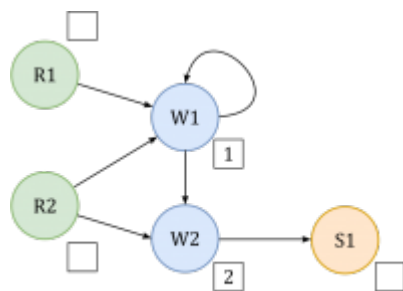
Parametry węzłów sieci:

- rampa #1: "częstotliwość" generowania półproduktów - 2 (tj. 1 półprodukt co 2 tury)
- rampa #2: "częstotliwość" generowania półproduktów - 1 (tj. 1 półprodukt co 1 turę)
- robotnik #1: czas przetwarzania półproduktów - 2 (tj. 2 tury)
- robotnik #2: czas przetwarzania półproduktów - 1 (tj. 1 turę)

Numery oznaczają ID półproduktów (u robotników na czerwono zaznaczono półprodukty w trakcie przetwarzania, na zielono - przetworzone i oczekujące na przekazanie dalej).



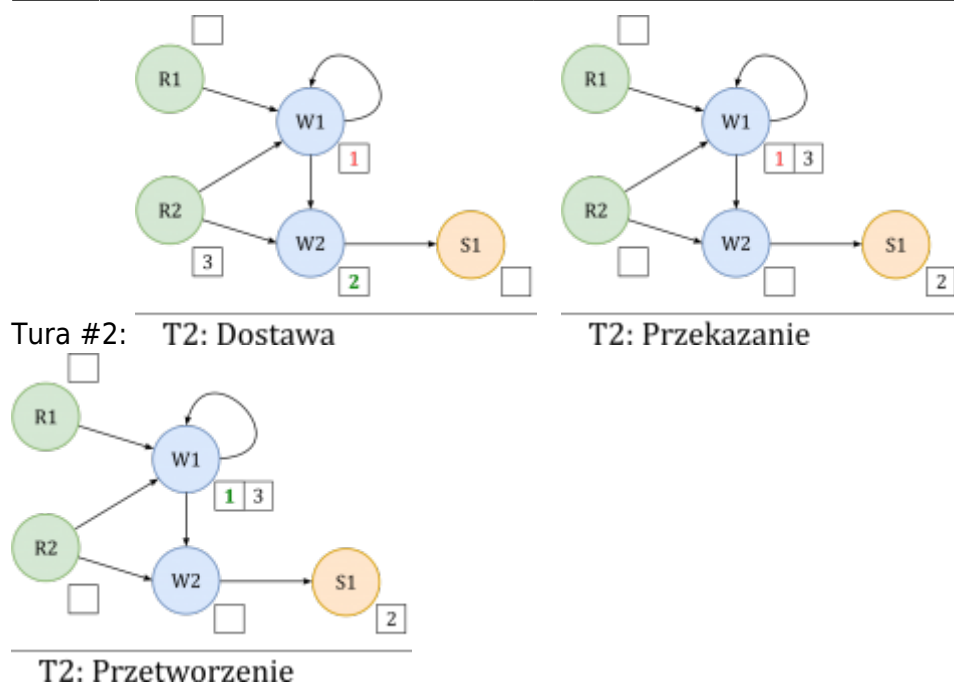
Tura #1: T1: Dostawa



T1: Przekazanie



T1: Przetworzenie



Uwagi

- Przed rozpoczęciem właściwej symulacji należy zweryfikować to, czy utworzona sieć jest spójna – tzn. czy każda jednostka półproduktu, rozładowana na obojętnie z której ramp, zostanie ostatecznie (po ewentualnym przetworzeniu) umieszczona w magazynie.
- W trakcie działania symulacji (czyli już po zweryfikowaniu poprawności sieci) nie dokonujemy już dalszych zmian w strukturze sieci (takich jak dodawanie/usuwanie węzłów bądź połączeń między węzłami).
- Symulacja kończy się po wykonaniu określonej liczby tur.

Raportowanie

Zadania: R

Tworzony system powinien umożliwiać dwie opcje raportowania:

- raport o strukturze sieci
- raport o stanie symulacji

Wygenerowane raporty mogą być zapisywane do pliku tekstowego lub na standardowe wyjście (w naszym przypadku – do konsoli).

Specyfikacje formatów raportów znajdują się [tutaj](#).

Zakres funkcjonalności aplikacji

Zaimplementowana przez Ciebie aplikacja do symulowania sieci powinna umożliwiać wykonanie kolejno następujących operacji:

1. Wczytanie struktury sieci z pliku.

Specyfikacja formatu pliku wejściowego znajduje się [tutaj](#).

Ewentualne błędy parsowania pliku powinny być sygnalizowane z użyciem mechanizmu wyjątków i obsługowane w głównej funkcji.

2. Dokonanie ewentualnych modyfikacji struktury sieci (z poziomu kodu programu) i zapis nowej struktury do pliku.**3. Weryfikacja poprawności wczytanej sieci.**

Ewentualne błędy parsowania pliku powinny być sygnalizowane z użyciem mechanizmu wyjątków i obsługowane w głównej funkcji.

4. Symulacja działania sieci wraz z raportowaniem.

Specyfikacje formatów raportów znajdują się [tutaj](#).

Zadania

Fragmenty systemu:

- Warstwa logiki biznesowej
 - [Półprodukty i ich przechowywanie](#) [P]
 - [Węzły sieci](#) [N]
 - [Fabryka](#) [F]
- [Warstwa odczytu i zapisu danych](#) [IO]
- [Warstwa prezentacji danych \(raporty\)](#) [R]
- [Symulacja](#) [S]

Etapy w ramach prac nad każdym z fragmentów systemu:

1. [Faza koncepcyjna](#)
2. [Faza projektowania interfejsów](#)
3. [Faza implementacji i testowania](#)

From:

<https://home.agh.edu.pl/~mdig/dokuwiki/> - **MVG Group**

Permanent link:

<https://home.agh.edu.pl/~mdig/dokuwiki/doku.php?id=teaching:programming:soft-dev:topics:net-simulation:start>



Last update: **2020/12/15 10:21**