http://xqtav.sourceforge.net

dr hab. Jerzy Tyszkiewicz
dr Andrzej Kierzek
mgr Jacek Sroka
Grzegorz Kaczor

XQTav - reprezentacja diagramów przepływu prac w formacie SCUFL przy pomocy XQuery

praca mgr pod kierunkiem
dr hab. Jerzego Tyszkiewicza

Grzegorz Kaczor, 181264

XQTav – reprezentacja diagramów Taverny w XQuery

Eksperymenty **in silico**:

- są stosowane w bioinformatyce
- obejmują przetwarzanie dużej ilości danych
- często wymagają dostępu do ogromnych baz danych
- korzystają z dostępnych przez Internet serwerów obliczeniowych

Tradycyjne sposoby wykonywania eksperymentów:

- jednorazowo – przeglądarka internetowa, formularze dostępne w Internecie (np. http://fasta.bioch.virginia.edu/)
- wielokrotnie – tworzenie skryptów łączących się z serwerami, tworzenie plików tymczasowych, ręczne tworzenie opisów wykonanych eksperymentów (np. do publikacji naukowych)

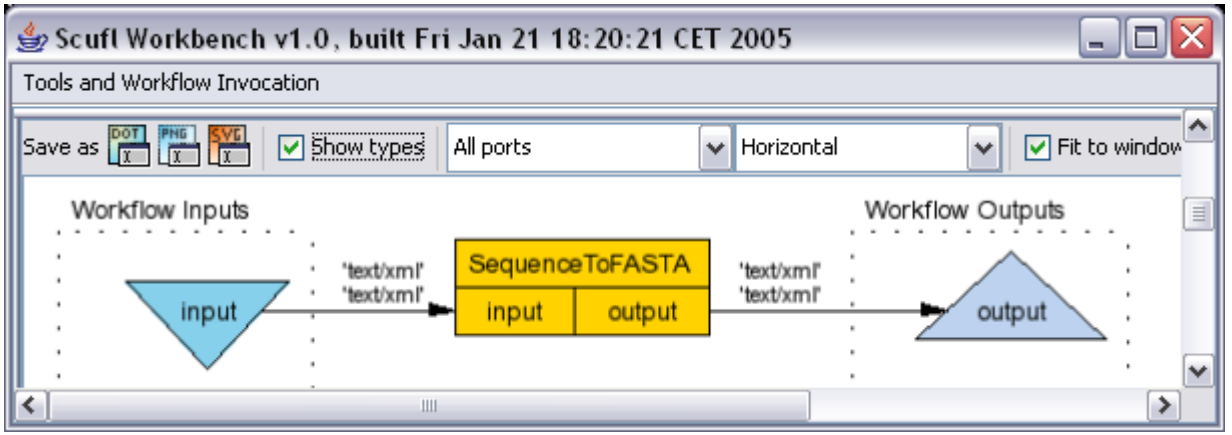
Wady takiego sposobu wykonywania eksperymentów

- wymaga od biologa **umiejętności programowania**
- konieczna jest **obsługa wszystkich** użytych **protokołów** sieciowych
- wyniki pośrednie trzeba przechowywać i czasem **konwertować** do innych formatów
- eksperymenty są **trudne do powtórzenia** na innych maszynach i przez inne osoby
- pisanie skryptów jest **pracochłonne** i **podatne na błędy**
- w publikacjach naukowych umieszczane są **tylko wyniki eksperymentu** i opis procedury – utrudnia to weryfikację lub modyfikacje przez inne osoby
- wielokrotnie implementuje się to samo w różnych eksperymentach

Taverna – wspomaganie tworzenia i wykonywania eksperymentów

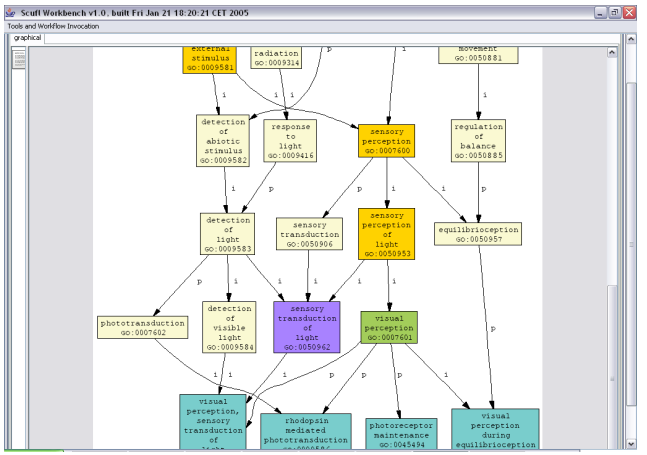
interfejs do MyGrid, http://taverna.sourceforge.net, inne systemy: BioKleisly, Kepler

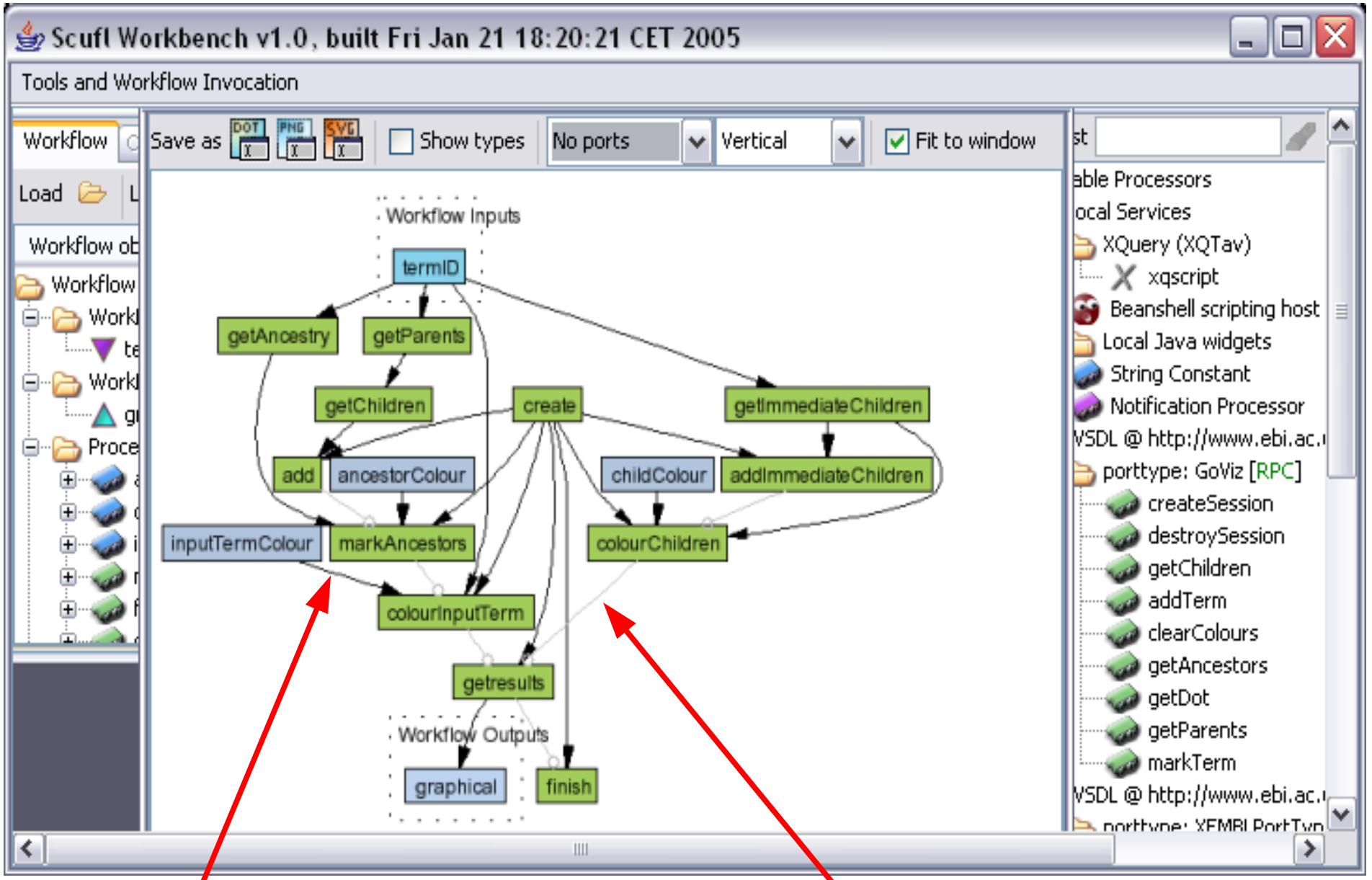
- wizualna reprezentacja eksperymentów, graf:
  - wierzchołki – **procesory** – aplikacje obliczeniowe (serwery, klasy lokalne)
  - krawędzie – **przekazywanie danych** lub **synchronizacja czasowa**
- **procesor** odpowiada jednej operacji logicznej – np. zastosowanie FASTY dla danych argumentów
- procesor ma **porty** wejściowe, na które przekazywane są argumenty, oraz wyjściowe, z których odczytuje się wyniki
- eksperyment można **uruchomić**



Taverna – zaawansowane funkcje

- możliwość definiowania **zagnieżdżonych diagramów** – wewnętrzny jest wtedy widoczny jako procesor
- **import/eksport danych** w postaci XML i surowej, a także XLS
- wyniki niekoniecznie w postaci tekstowej, także np. grafika (zwykle **dot**)
- mechanizmy kontroli błędów
  - **ponawianie prób** po ustalonym czasie oczekiwania
  - **procesory alternatywne** – uruchamiane w przypadku niedostępności procesora bazowego





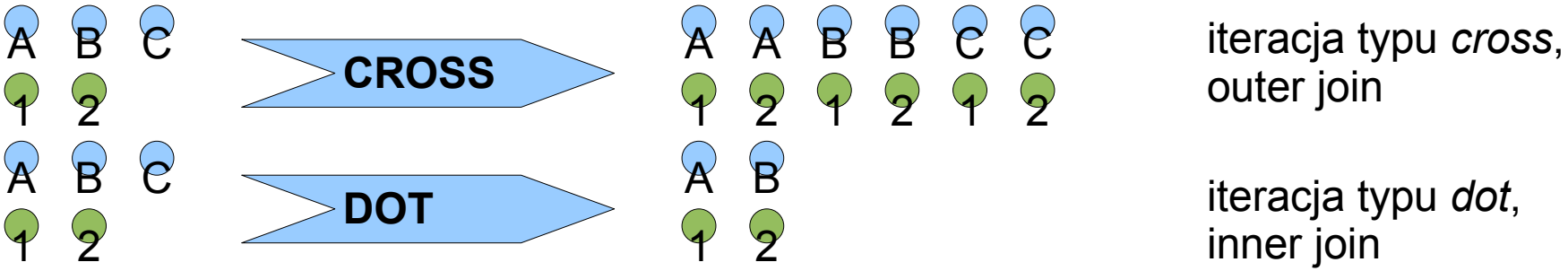
przekazywanie danych

synchronizacja czasowa

Wady Taverny

- wbudowane lokalne procesory zapewniają bardzo **mało użytecznej funkcjonalności**, bez programowania trudno jest dodać nową
- są problemy, które **w diagramach zapisuje się trudno**, np. odpowiednik zapytania SQL: *select * from T1, T2 where T1.v = 2*T2.v*
- część z tych problemów daje się łatwo wyrazić w XQuery
- Taverna ukrywa tzw. **strategie iteracji** w przypadku niedopasowania typów

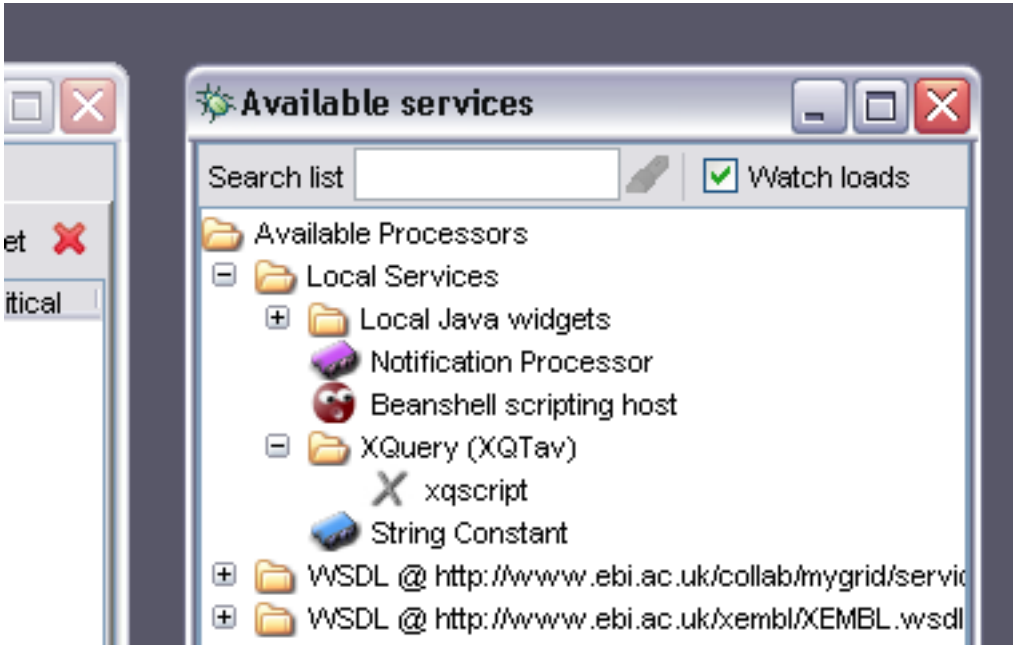
procesor, który oczekuje po jednym elemencie na port, a dostaje dwie listy



kolumna odpowiada jednemu wywołaniu procesora

Procesor umożliwiający wykonywanie XQuery

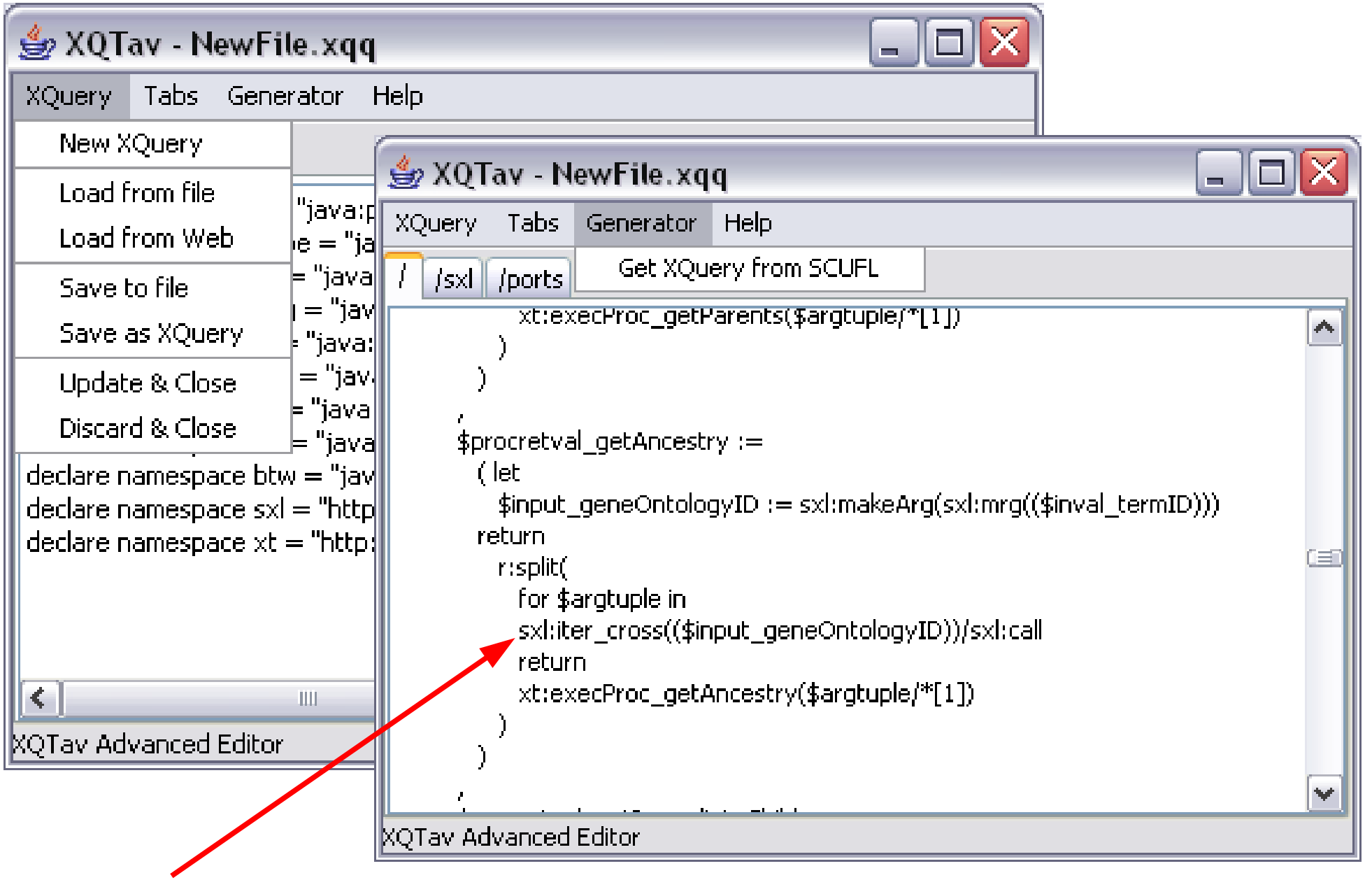
- integruje się z Taverną jako procesor
- umożliwia definiowanie portów wejściowych i odczytywanie z nich danych
- umożliwia wykonanie dowolnego XQuery i przekazanie wyników do Taverny
- można korzystać z napisanych przez siebie klas Javy do wykonywania bardziej skomplikowanych przekształceń
- wykorzystuje engine XQuery Saxon-B (http://saxon.sourceforge.net)



Generowanie XQuery z diagramów przepływu prac

- wygenerowany skrypt robi to, co diagram, z którego został wygenerowany
- wywołania procesorów zamieniane są na funkcje XQuery
- specjalny komentarz XQuery definiuje strukturę portów wejściowych i wyjściowych skryptu
- strategie iteracji zrealizowane przy pomocy pętli języka
- nie ma łącz synchronizacji czasowej
- dla czytelności skrypt podzielony na zakładki
- wbudowany edytor skryptu, umożliwiający także generowanie XQuery
- dane Taverny konwertowane do postaci XML, można je zmieniać w skrypcie

Edytor XQuery



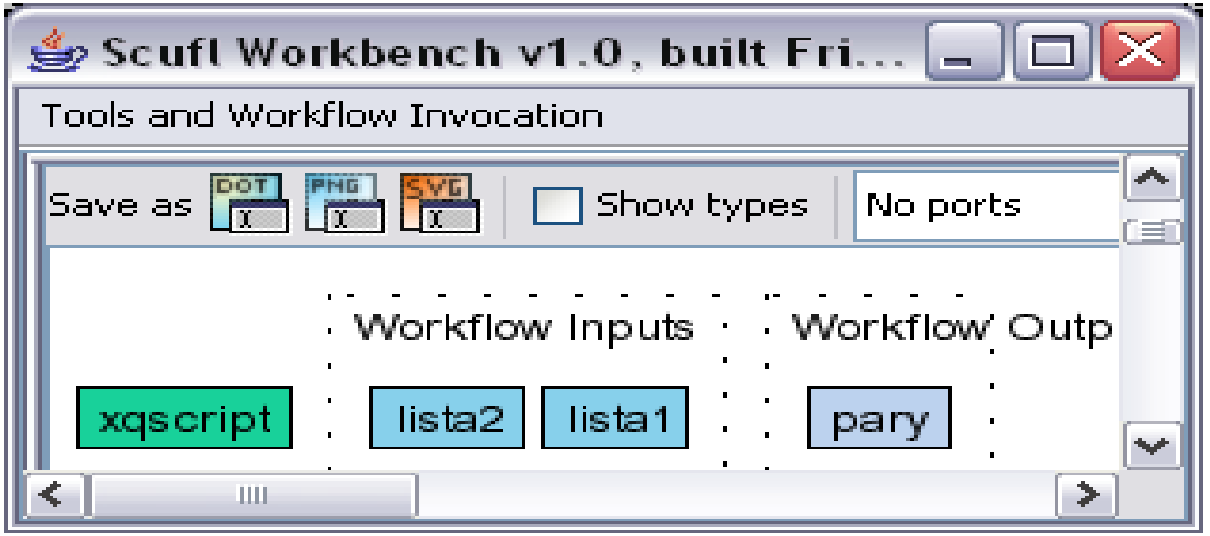
zastosowanie strategii iteracji cross

Rozwiązanie przykładowego problemu

- chcemy z dwóch list wybrać pary napisów o tej samej długości, jednak nie przekraczającej 5 znaków

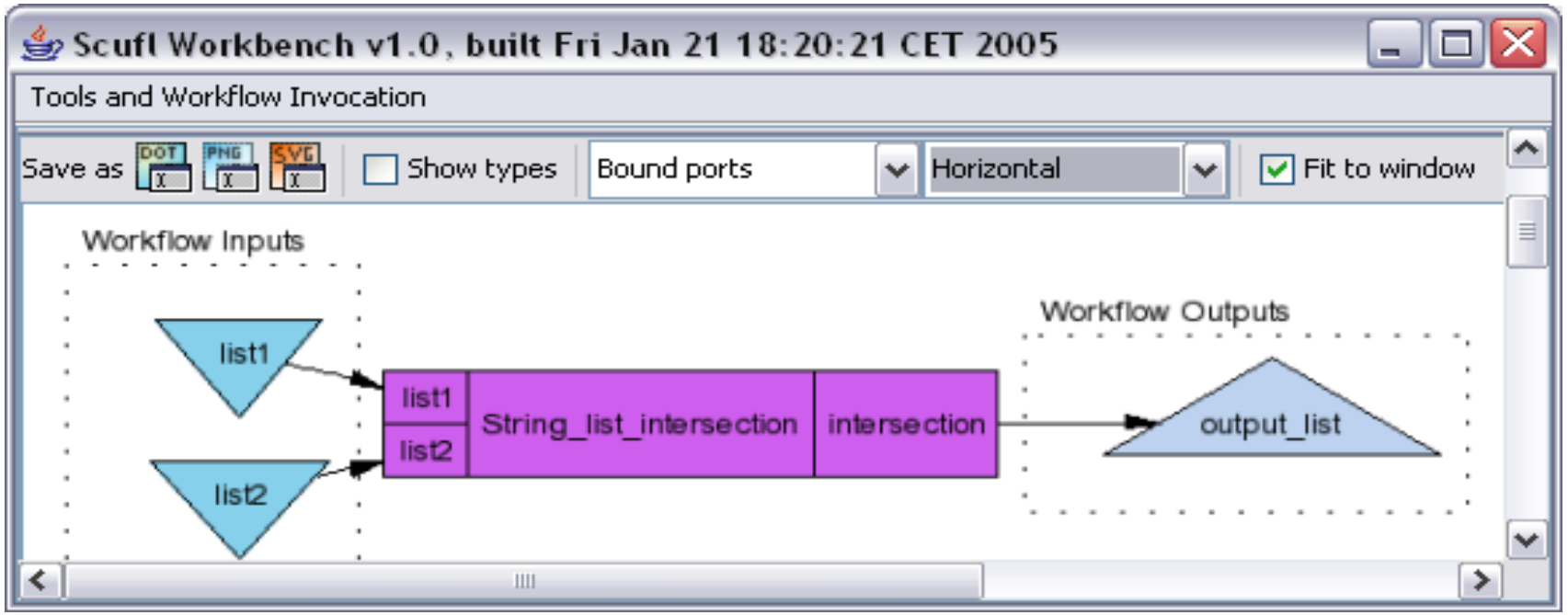
Krok 1 – tworzymy podstawowy diagram

- definiujemy dwa wejścia i jedno wyjście oraz procesor XQuery *xqscript*



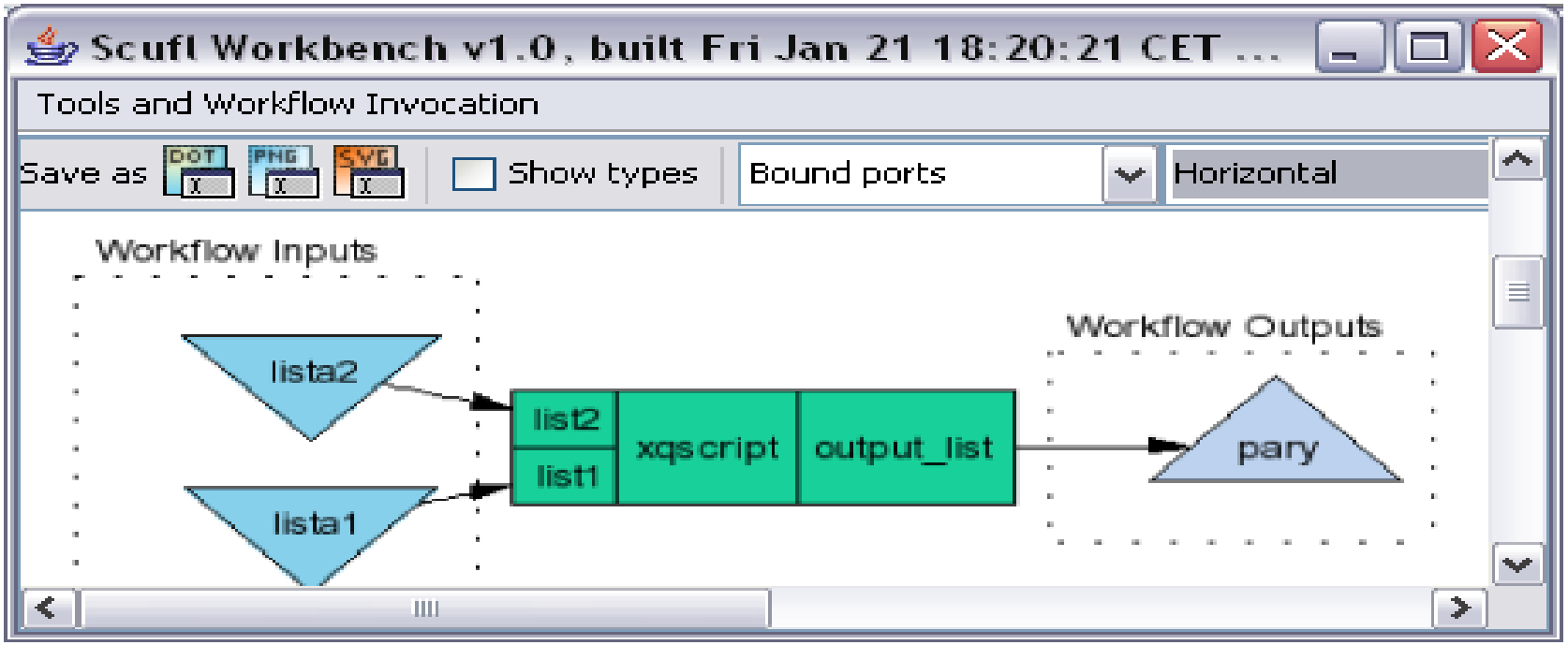
Krok 2 – tworzymy diagram pomocniczy

- powinien rozwiązywać problem jak najbardziej zbliżony do danego

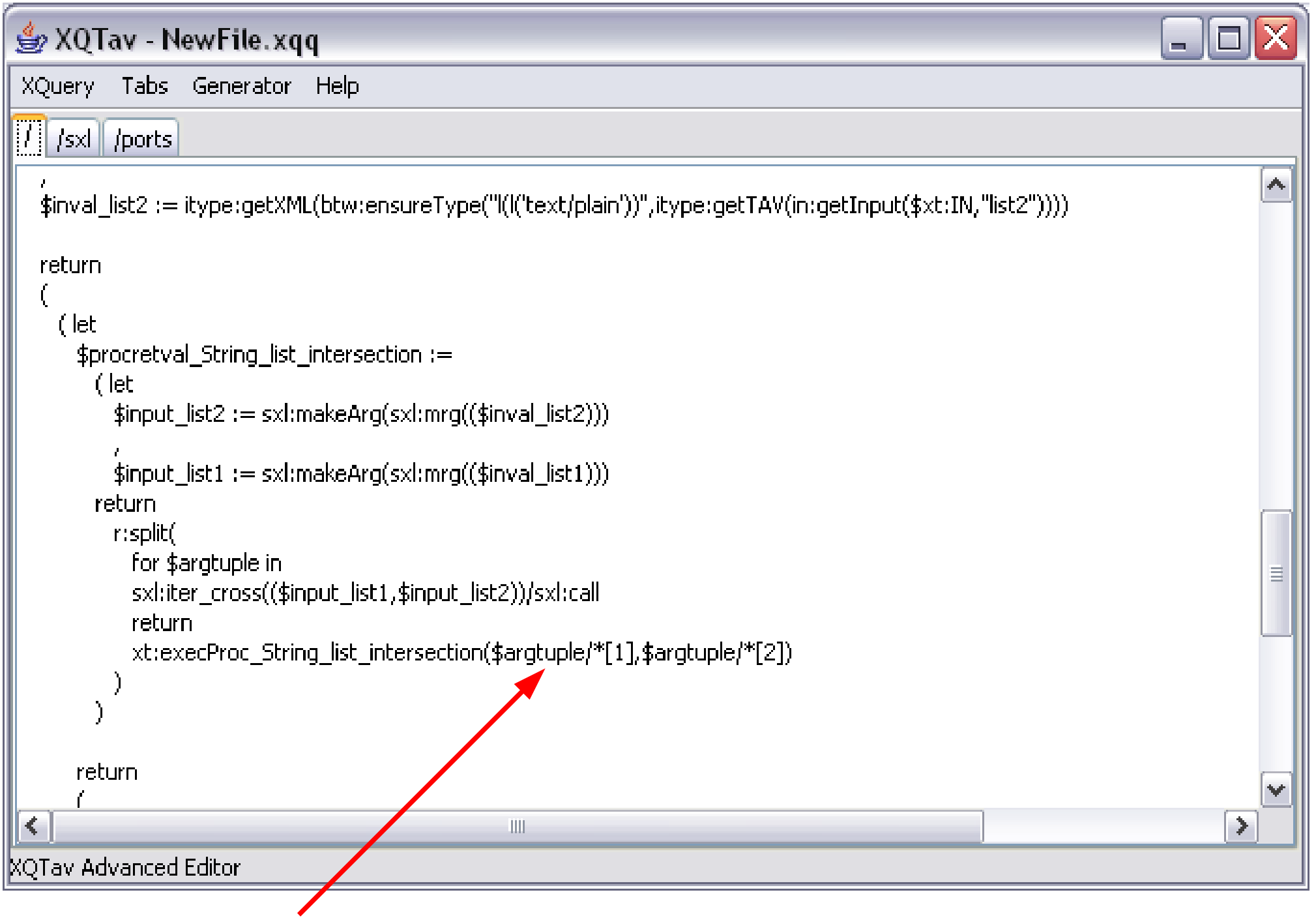


Krok 3 – generujemy XQuery z diagramu pomocniczego

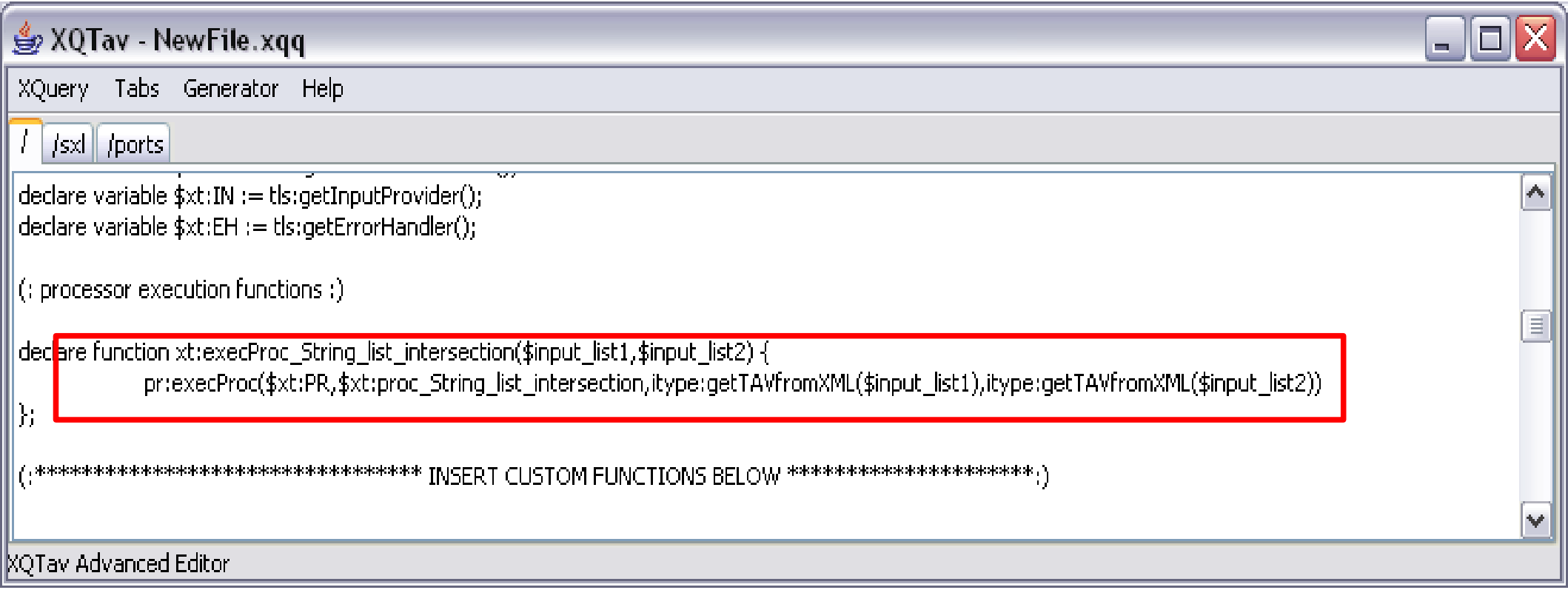
- w diagramie podstawowym uruchamiamy edytor XQuery i generujemy skrypt



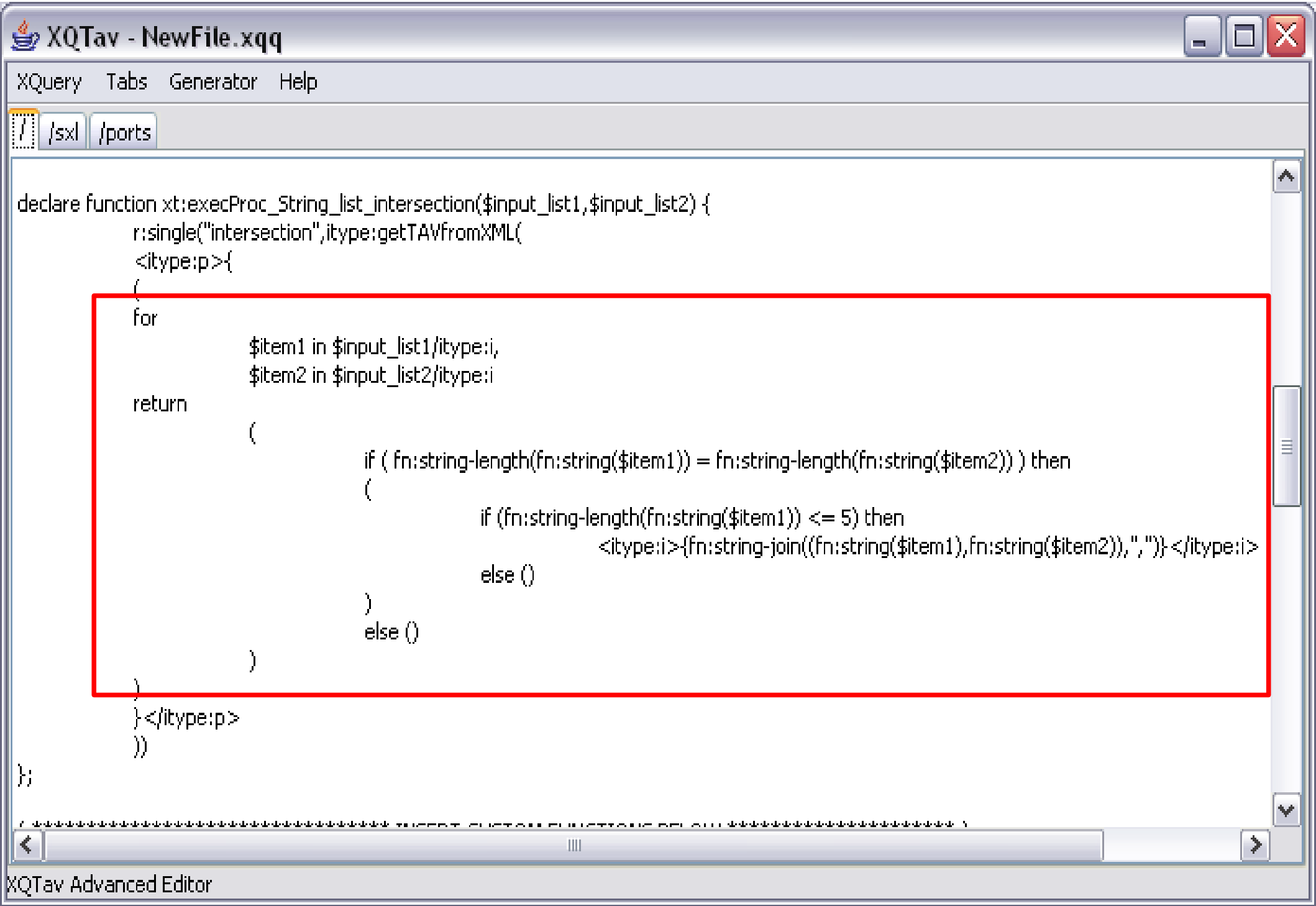
- łączymy wejścia i wyjście diagramu z odpowiednimi portami procesora, edytujemy XQuery procesora

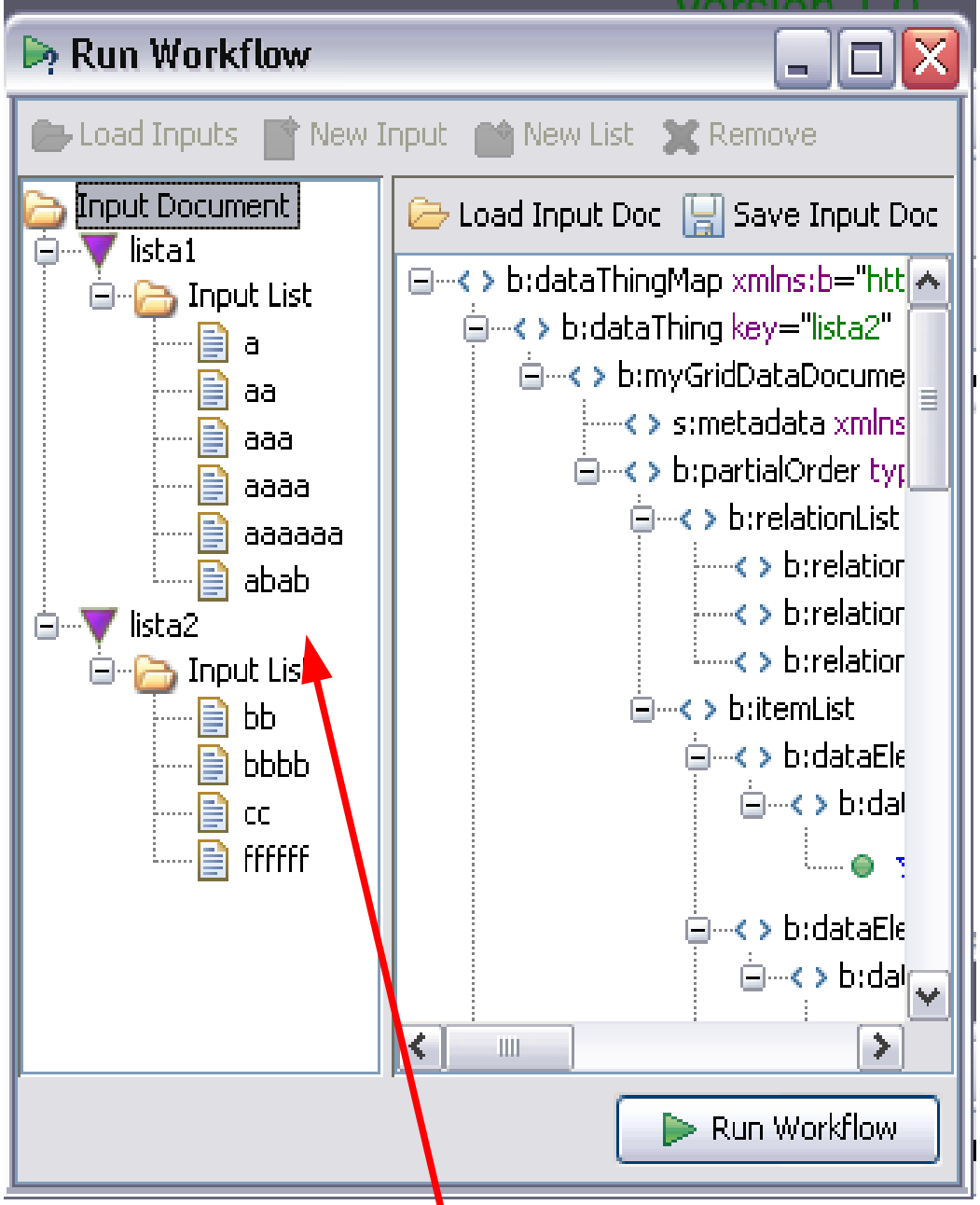


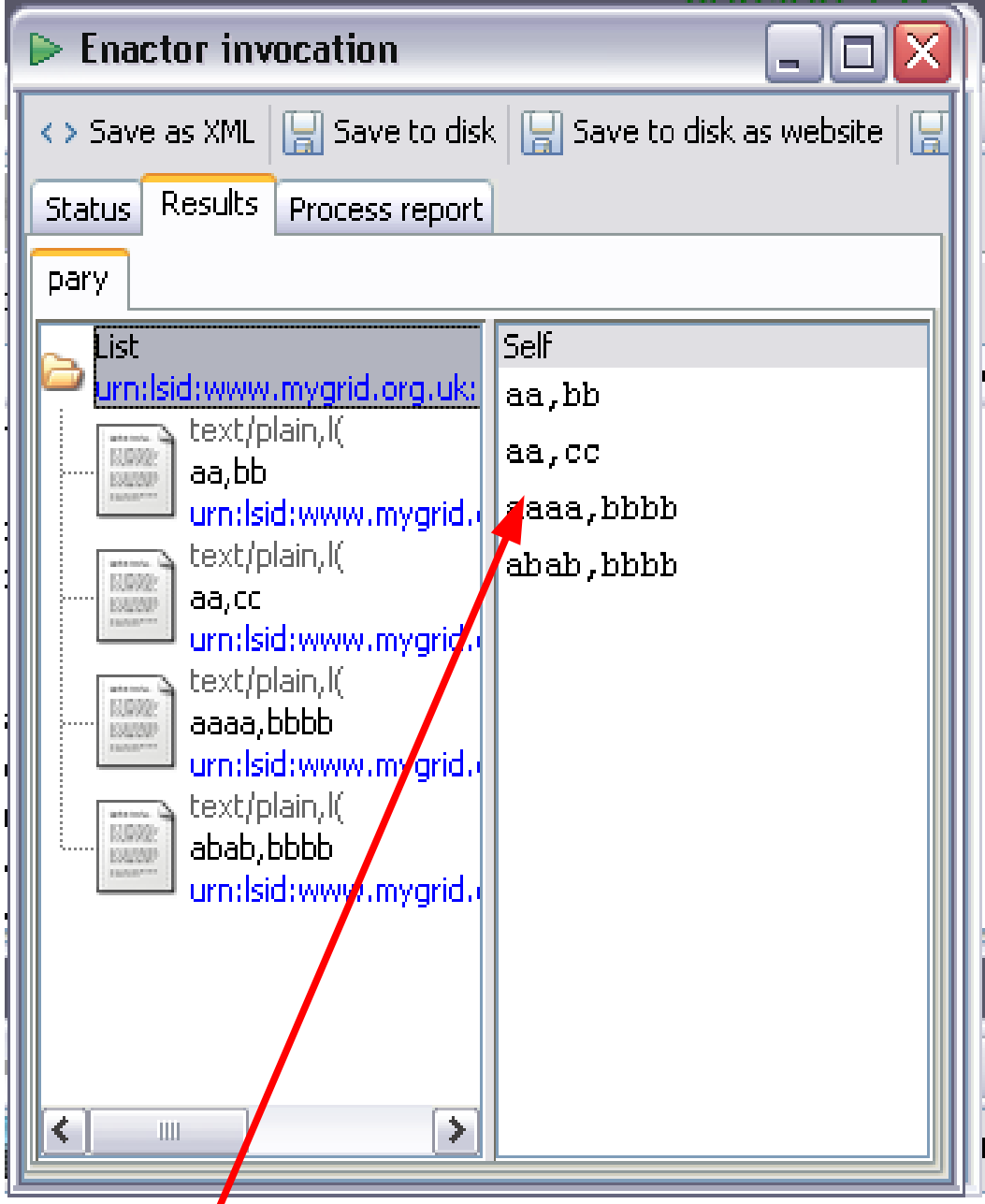
w tym miejscu wywoływany jest procesor *intersection*



- w tym miejscu wywoływany jest procesor Taverny
- chcemy ten kod zmodyfikować tak, żeby zamiast wywołania procesora wykonane zostało obliczenie wybierające pary napisów o tych samych długościach, nie większych jednak od 5


XQTav - NewFile.xqq
XQuery Tabs Generator Help
/ /sxl /ports
declare function xt:execProc_String_list_intersection($input_list1,$input_list2) {
r:single("intersection",itype:getTAVfromXML(
<itype:p>{
(
for
$item1 in $input_list1/itype:i,
$item2 in $input_list2/itype:i
return
(
if (fn:string-length(fn:string($item1)) = fn:string-length(fn:string($item2))) then
(
if (fn:string-length(fn:string($item1)) <= 5) then
<itype:i>{fn:string-join((fn:string($item1),fn:string($item2)),",")}</itype:i>
else ()
)
else ()
)
)
}</itype:p>
))
};
XQTav Advanced Editor





dane wejściowe

połączone listy

Podsumowanie

- jeśli problem jest prosty – należy używać standardowej Taverny, bo XQuery wciąż jest językiem dość trudnym
- jeśli problem nadaje się dobrze do wyrażenia w XQuery – można użyć XQTav; wspierane są także rozszerzenia SQL zaimplementowane w Saxonie
- jeśli problem jest naprawdę skomplikowany – można napisać klasę w Javie, a następnie użyć XQTav jako interfejsu między Taverną a Javą

W sieci

- MyGrid - http://www.mygrid.org.uk
- Taverna - http://taverna.sourceforge.net
- Saxon - http://saxon.sourceforge.net
- XQTav - http://xqtav.sourceforge.net