



INSTITUT PASTEUR



Conception et programmation d'un gestionnaire graphique de
processus bioinformatiques d'analyse de séquences
et
application à l'identification des résidus encodant la spécificité de
la reconnaissance de l'interleukine-2 humaine par ses récepteurs.

Franck VALENTIN
franck_valentin@yahoo.fr

29 septembre 2004

Préambule

- **Cadre**

- Pasteur, Unité d'Immunogénétique Cellulaire.
- Identification des résidus encodant la spécificité de la reconnaissance de l'IL-2 par ses récepteurs

- **Les besoins**

- chaîner intuitivement et visuellement des applications pour tester des hypothèses de travail.
- partager les procédures.
- faciliter l'apprentissage des outils de bioinformatique.

- **Workflow**

- "Application qui permet de séquencer des tâches suivant un modèle définissant en particulier comment ces tâches sont synchronisées".

Plan

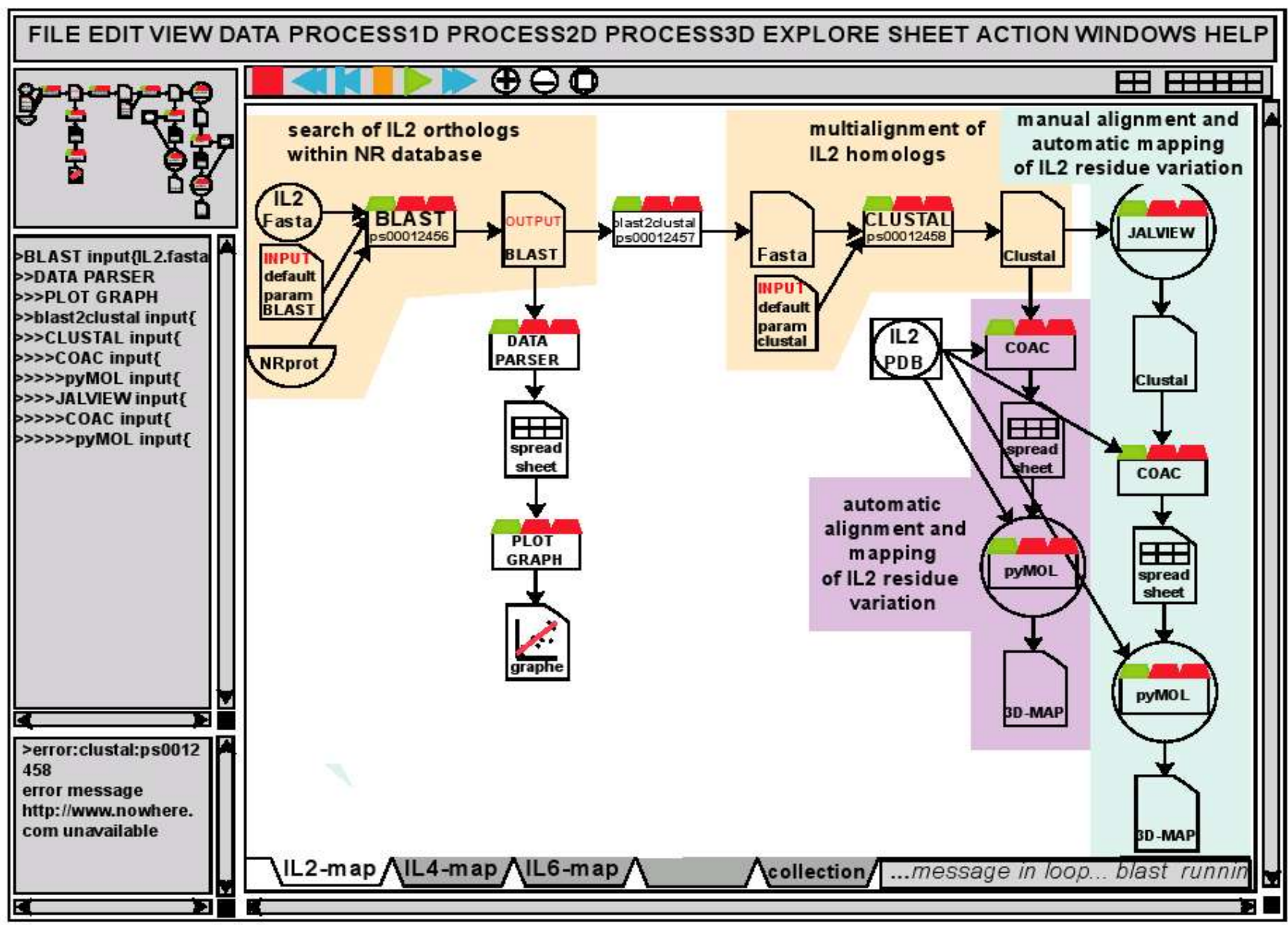
1. Nos besoins : cahier des charges
2. Les solutions disponibles
3. Choix retenu
4. Fonctionnalités développées
5. Application à la reconnaissance de l'IL-2
6. Futur et réflexions

Nos besoins

- Principalement

- Glisser/déposer des applications/données/contrôles dans un environnement de travail.
- Les relier entre elles pour définir le workflow.
- Exécutions séquentielles ou concurrentes.
- Exécuter un sous-ensemble du workflow.
- Pouvoir faire des exécutions pas à pas, ajouter des points d'arrêt, faire des reprises sur erreur.
- Suivre l'état de l'exécution.
- Pouvoir comparer des résultats entre plusieurs exécutions.
- Définir des opérateurs de comparaison, des boucles.

Maquette



... mais aussi

- Permettre à l'utilisateur de rajouter simplement de nouvelles applications (intégration de briques PISE par exemple).
- Vérifier la compatibilité des données, les rendre compatibles par l'ajout d'adaptateurs.
- Intégrer des services web.
- Permettre l'exécution en arrière plan sans interface graphique mais pouvoir ensuite visualiser à chaque moment l'état du workflow et le modifier.
- et beaucoup d'autres idées

Formalisons

• Elaboration d'un cahier des charges

P?9	Ajout de types de données
	<p>Des types doivent pouvoir être définis pour les données échangées dans un workflow. Ces types peuvent être primitifs (entiers, booléens,...), des types "biologiques" (séquence d'ADN ou protéique, structures ...), des collections de types (collections ordonnées ou non, avec doublons ou non...) ou encore une référence vers une donnée (ex. nom de fichier ou URN décrivant une séquence).</p> <p>Cette fonctionnalité permet d'une part d'attirer l'attention de l'utilisateur sur d'éventuelles incompatibilités (les connexions sont autorisées mais affichées différemment) et d'autre part de transformer automatiquement ces données pour les rendre compatibles par l'intermédiaire d'outils comme readseq ou squizz par exemple.</p> <p>Le typage des données permet de plus de guider l'utilisateur en lui proposant les applications disponibles pour un certain type de données.</p> <p>Cette fonctionnalité est à approfondir :</p> <ul style="list-style-type: none"> - l'utilisateur peut-il rajouter ses types de données, définir leurs compatibilités et les outils de transformations ? - comment peut-on utiliser les ontologies déjà définies (cf. BioMOBY, myGRID, GeneOntology, ...)

P10	Liens entre données et acteurs
P10.1	Le lien (entrée ou sortie) entre une donnée et le port d'entrée d'un acteur peut se faire même s'ils ne sont pas compatibles.
P10.2	Si la donnée et l'acteur ne sont pas compatibles, le lien est représenté de manière différente. L'utilisateur a la possibilité de demander à l'application d'ajouter un ou des acteurs intermédiaires (wrappers) pour les rendre compatibles. Si plusieurs possibilités existent une liste est proposée.
P10.3	Les acteurs intermédiaires ne font que de la traduction de données (adaptation de formats par exemple).

P?11	Support des services web
	<p>A définir en fonction des solutions retenues pour les types de données. En plus du service proprement dit, il serait intéressant d'utiliser des mécanismes de découverte de ces services (à voir en fonction des solutions disponibles).</p> <p>Remarque : Kepler implémente déjà des services web, voir comment les adapter à notre plateforme.</p>

P?12	Répartition de la charge d'exécution
	<p>L'application doit permettre une répartition de la charge si plusieurs ressources de calcul sont disponibles. Cette fonctionnalité est à approfondir !</p> <p>Nous pouvons a priori considérer au moins trois cas :</p> <ul style="list-style-type: none"> - les acteurs lancent des programmes locaux (création de processus). Un outil externe comme PBS peut être utilisé. - Les acteurs font partie intégrante de l'application (ils sont lancés sous forme de threads). - Les acteurs sont des services web ou font appel à des ogi.

P18	Exécution en arrière plan
P18.1	Les workflows créés par l'utilisateur doivent pouvoir être lancés en arrière plan, leur exécution ne sera alors pas interrompue par la fermeture du gestionnaire graphique.
P18.2	L'utilisateur pourra à partir du gestionnaire graphique rouvrir une instance à partir de la liste des workflows lancés en arrière plan (issu de cette session graphique ou d'une autre) par cet utilisateur.
P18.3	Le workflow ouvert affichera l'état en cours et sera modifiable comme un workflow classique.

P19	Données
	Les données en entrée (i.e. : qui ne sont pas issue d'un acteur) suivantes doivent être définies en priorité, cette fonction est cependant dépendante du choix retenu pour P?9 :
P19.1	Une séquence de protéines, son identificateur ou une liste de séquences de protéines.
P19.2	Une structure de protéines, son identificateur ou une liste de séquences de protéines.
P19.3	Une ou des listes de paires de séquences de protéines formant un réseau continu ou disjoint.
P19.4	Un arbre phylogénétique ou une liste d'arbres.

A1	Affichage des liens
	Lors de la création d'un lien entre une donnée et un acteur, le trait est continu si la compatibilité est confirmée et discontinu sinon.

A2	Zooms
A2.1	Zoom sur un acteur qui contient un workflow : les acteurs qui contiennent des workflows peuvent être visualisés sous leur forme condensée (un acteur) ou éclatée (éléments du workflow).
A2.2	Les éléments du workflow ont plusieurs représentations en fonction du zoom (avec plus ou moins de détails). A confirmer cependant, il semble plus judicieux de garder la même représentation quel que soit le zoom.
A2.3	Il doit être possible d'afficher un ensemble d'éléments sous la forme d'un seul (différent de la transformation d'un workflow en acteur) par exemple après sélection avec un menu (view as box) puis ensuite revenir à la vue éclatée.

A3	Représentation arborescente
A3.1	Les éléments du workflow (données, acteurs, liens) seront aussi représentés et modifiables sous forme d'arbres.

A4	Facilités de visualisation
A4.1 ✓	Un espace affichera une vue générale du workflow.
A?4.2	Une fenêtre "tip" affichera une aide sur l'acteur sélectionné dans le workflow ou dans la liste des applications (voir le workflow VIBE).

2 - Les solutions disponibles

Taverna

- European Bioinformatics Institute (EBI), IT Innovation and the Rosalind Franklin Centre for Genomic Research (RFCGR), New Castle Computer Science Faculty and for Life, Manchester Science Faculty, Nottingham University.
- Orienté services web et calcul distribué (myGRID, BioMOBY, Soaplab).
- Pas à proprement parler de manipulation directe (seulement une représentation graphique).

- Ajout de nouvelle brique pas élémentaire.

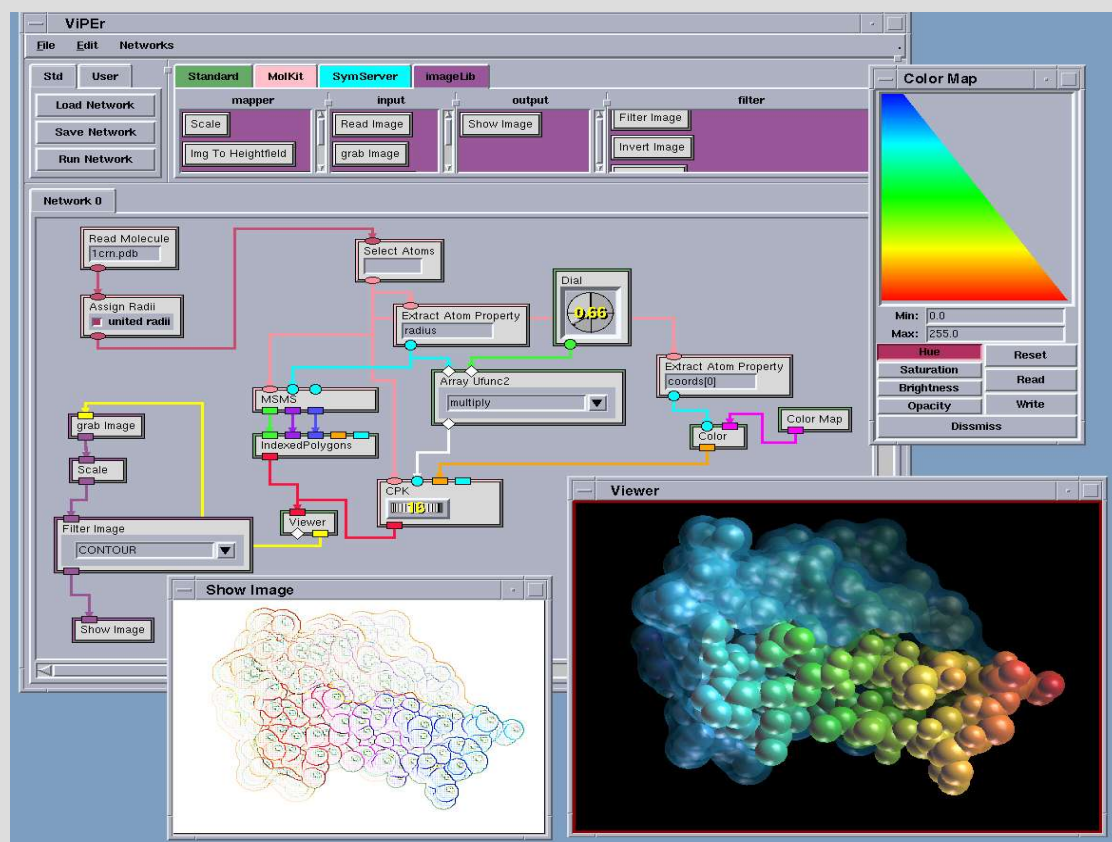
The screenshot displays the Taverna Scuff Workbench interface, which is used for managing and executing workflows. It consists of several main windows:

- Advanced model explorer:** This window shows a hierarchical view of the workflow object. It includes sections for 'Workflow inputs', 'Workflow outputs', and 'Processors'. The 'Processors' section lists various tasks such as 'example_sequenceID', 'blast_database', 'get_sequence', and 'blastx_ncbi', along with their respective parameters and values.
- Workflow diagram:** This window provides a visual representation of the workflow. It shows a flow from an input 'example_sequenceID' through a 'sequence_usa' processor to a 'query_sequence' processor, which then interacts with a 'blast_database' and a 'blastx_ncbi' processor to produce 'outfile_SeqretTest' and 'blast_result' outputs.
- Available services:** This window lists a variety of web services available for integration into workflows. Services include 'TEST - always fails', 'Notification Processor', 'String Constant', 'Soaplab', 'alignment_consensus', 'alignment_differences', 'alignment_global', 'alignment_local', 'matcher', 'seqmatchall', 'supermatcher', 'water', 'wordmatch', 'alignment_multiple', 'display', 'edit', 'graphics', 'nucleic_codon_usage', 'gowlab', 'interproscan', 'testing', 'Biomoby', 'prometheus', 'schematikon', 'www.hapmap.org', 'getAlleleFreq', 'getGenotypes', 'atidb.org', 'icapture', 'tropgenedb', 'heaven', 'VWSL', 'porttype: GoQuery', 'Soaplab', 'protein_structure', 'jess', 'Classic', and 'alignment'.

The interface also includes a taskbar at the bottom with various application icons and a system tray showing the time as 22:42.

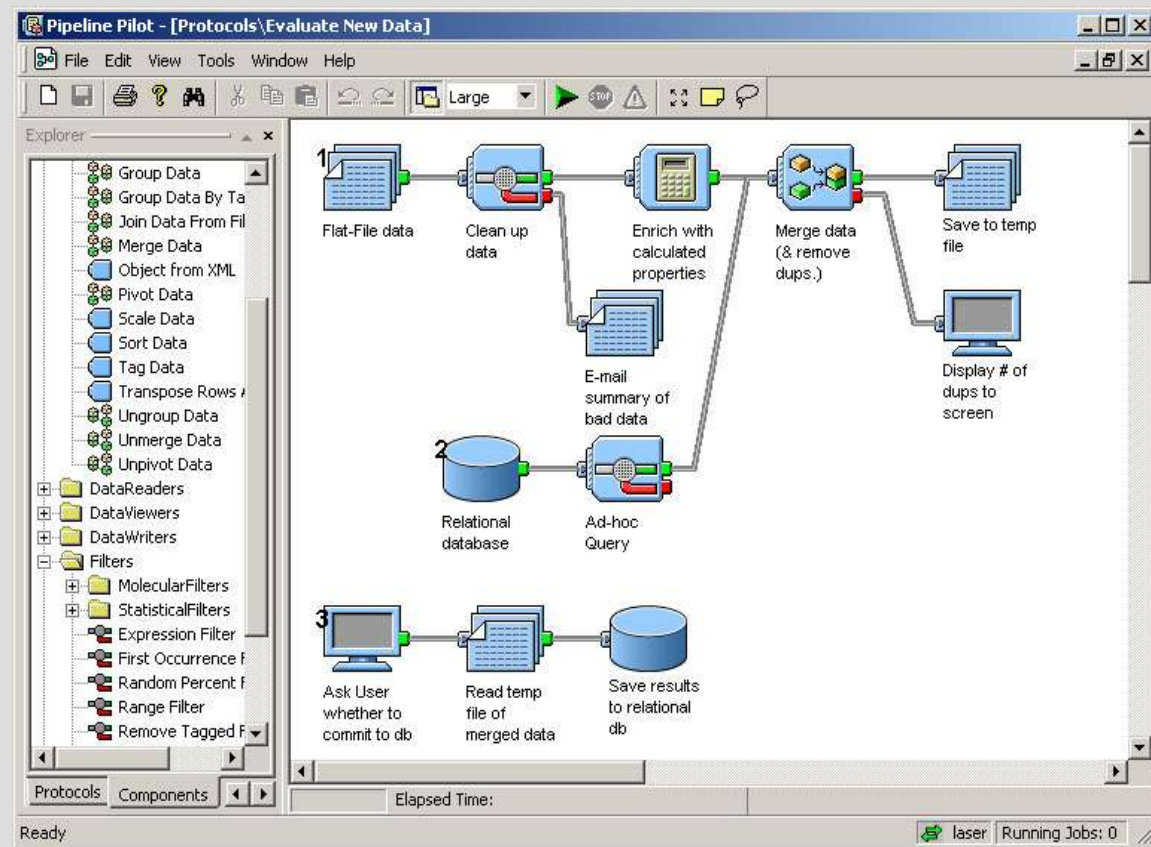
VipER

- Scripps research Institute, San Diego.
- Écrit en Python, workflow sauvé sous la forme de code Python.
- Essentiellement destiné à la biologie structurale.
- Plus proche du modèle Dataflow.



Pipeline Pilot

- Société Scitegic (San Diego)
- Industrie pharmaceutique
- Langage pour définir des briques
- Exécution à partir d'un formulaire web



VIBE

- Société Incogen (Williamsburg)
- Applications d'analyse de séquences
- Test de la validité et de la compatibilité des données échangées
- Plusieurs espaces de travail
- Aide pour chaque brique.
- Ajout de notes de l'utilisateur.

The screenshot displays the INCOGEN VIBE v2.4.1.6 software interface. The main window shows a workflow diagram with 10 numbered modules. A red box highlights a toolbar at the top containing icons for Smith-Waterman, HMMScan, BLASTX, FASTX, FASTY, BLASTP, HMMSearch, FASTA, and BLAT-AA. Another red box highlights a help window for HMMBUILD, which contains the following text:

HMMBUILD builds a hidden Markov model (HMM) based on a multiple sequence alignment generated from ClustalW. The model is saved in HMMER 2.0 format.

HMMs are statistical models of the primary structure consensus of a sequence family designed to improve the sensitivity and speed of database searching. HMMER 2.0, HMM software for biological sequence analysis, was developed by Sean Eddy from the Washington University School of Medicine. For more information on HMMs, see Sean Eddy's webpage: <http://hmm.wustl.edu>.

[COMMENT] Comments or descriptions of the type of search, data, etc.

[VERSION] The version of HMMER used to build the HMM

For more information on HMMER 2.0 and its options, see Sean Eddy's

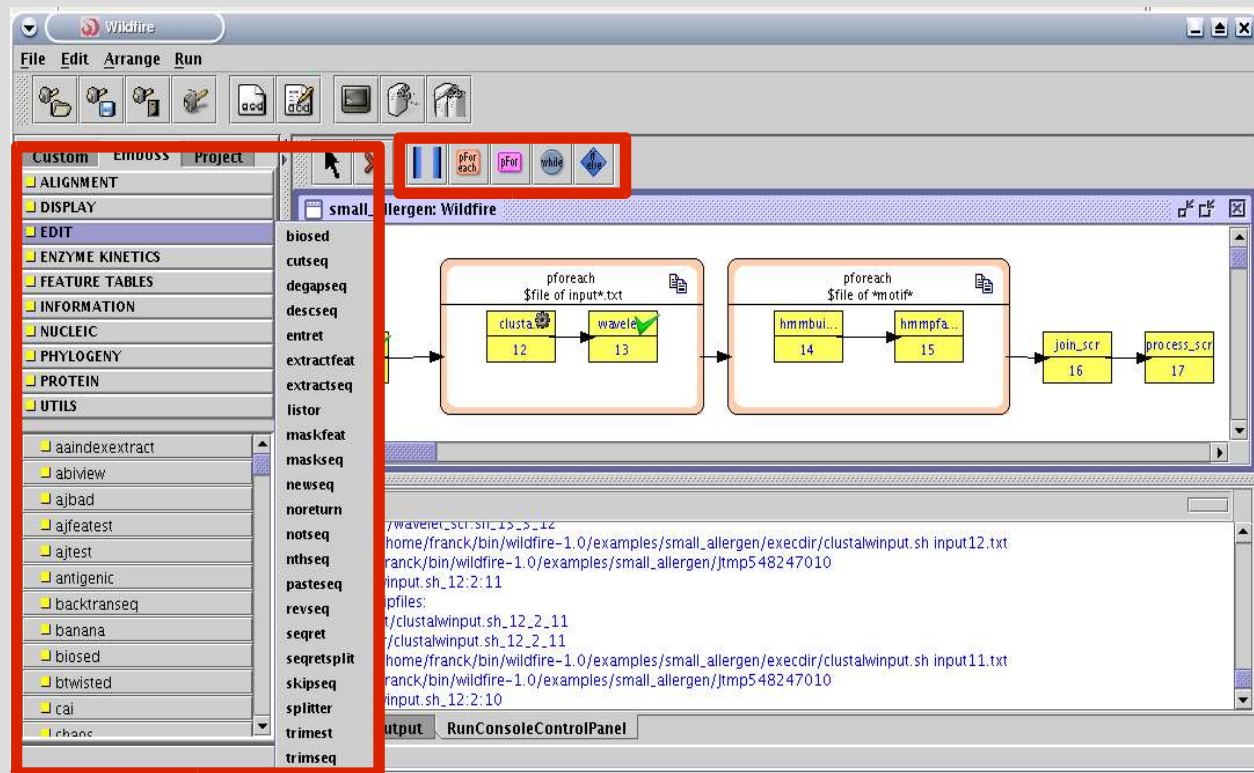
At the bottom of the interface, a terminal window shows the following output:

```
COMPLETE: fasta_nt [Module-2] (fasta_nt1-1044552275433)...
READY: SimViewer [Module-3] (SimViewer2-1044552287495)...
EXECUTING: SimViewer [Module-3] (SimViewer2-1044552287495)...
COMPLETE: SimViewer [Module-3] (SimViewer2-1044552287495)...
```

The status bar at the bottom indicates "Mouse Chr 1" and "Module 8 - running". The system memory usage is shown as "Memory free: 9420KB total: 16476KB".

Wildfire

- A-STAR (Agency for Science Technology and Research, Singapour)
- Utilisation d'itérations, boucles, conditions.
- Fortement lié à EMBOSS.
- Ajout de nouvelles applications par définition d'un fichier ACD.
- Plusieurs supports pour les exécutions concurrentes (langage GEL).
- Communication inter-tâches par fichiers.



Ptolemy II

- Département EECS (Electrical Engineering and Computer Science, université de Berkeley).
- Initialement pour modéliser des systèmes embarqués.
- Plusieurs modèles d'ordonnancement

The screenshot displays the Ptolemy II software interface. The main window shows a model diagram with the following components and connections:

- Signal Source**, **Carrier Source**, and **Noise Source** are connected to an **Expression2** block.
- The **Expression2** block is configured with the expression: $\text{signal} * \text{carrier} + \text{noise}$.
- The output of the **Expression2** block is connected to a **Spectrum** block.
- The **Spectrum** block is connected to a **Frequency Domain Display**.
- The output of the **Expression2** block is also connected to a **Time Domain Display**.

The interface includes a menu bar (File, View, Edit, Graph, Debug, Help) and a toolbar with various icons. A left-hand pane shows a hierarchical tree of components, including Decorative, Parameters, UnitSystems, Directors, Actors, Sources, Sinks, Array, Conversions, FlowControl, HigherOrderActors, IO, Logic, and Math. The Math section is expanded, showing blocks like AbsoluteValue, AddSubtract, Accumulator, Average, Counter, Differential, DotProduct, Expression, Limiter, and LookupTable.

Text annotations on the screenshot provide additional information:

- SDF Director:** This model shows a simple periodogram spectral estimate of a modulated sinusoid in noise. The top-level parameters control the carrier frequency, the signal frequency, and the noise level. Notice that the two peaks are centered at the carrier frequency, with their distance from the carrier given by the signal frequency. The sample rate is assumed to be 8kHz.
- Parameters:**
 - carrierFrequency: 2000.0
 - signalFrequency: 500.0
 - noiseStandardDeviation: 0.1
- Block Interaction:** The blocks with red outlines are hierarchical. Right click and select "Look Inside". These generate sinusoids, one for the signal and the other for the carrier.
- Expression Block:** The Expression block calculates a mathematical expression, as shown.
- Execution Instructions:** Select "Run Window" from the View menu to execute the model, or click on the red triangle in the toolbar. Try changing the parameters in the run window or on the diagram.

Author: Edward A. Lee

Ptolemy II - environnement

The screenshot displays the Ptolemy II environment with a simulation diagram. The diagram includes a 'Carrier Source' block, a 'Noise Source' block, an 'Expression2' block containing the mathematical expression $signal * carrier + noise$, and a 'Spectrum' block. A 'Domain Display' block is also present. The interface features a menu bar (File, View, Edit, Graph, Debug, Help), a toolbar with icons for search, zoom, and execution, and a left-hand pane with a hierarchical tree of components like 'Decorative', 'Parameters', 'UnitSystems', 'Directors', 'Actors', 'Sources', 'Sinks', 'Array', 'Conversions', 'FlowControl', 'HigherOrderActors', 'IO', 'Logic', and 'Math'. A parameter table is visible on the right side of the diagram, listing parameters such as 'carrierFrequency: 2000.0', 'signalFrequency: 500.0', and 'noiseStandardDeviation: 0.1'. The diagram is annotated with several yellow callout boxes containing text in French.

Sauvegarde au format XML

Modèle d'ordonnancement

- opération
- fonction mathématiques,
- conversions entre types,

Commentaires

Langage d'expressions

- textes et graphiques,
- écriture de données

Définition de paramètres

Affichage graphique 2D

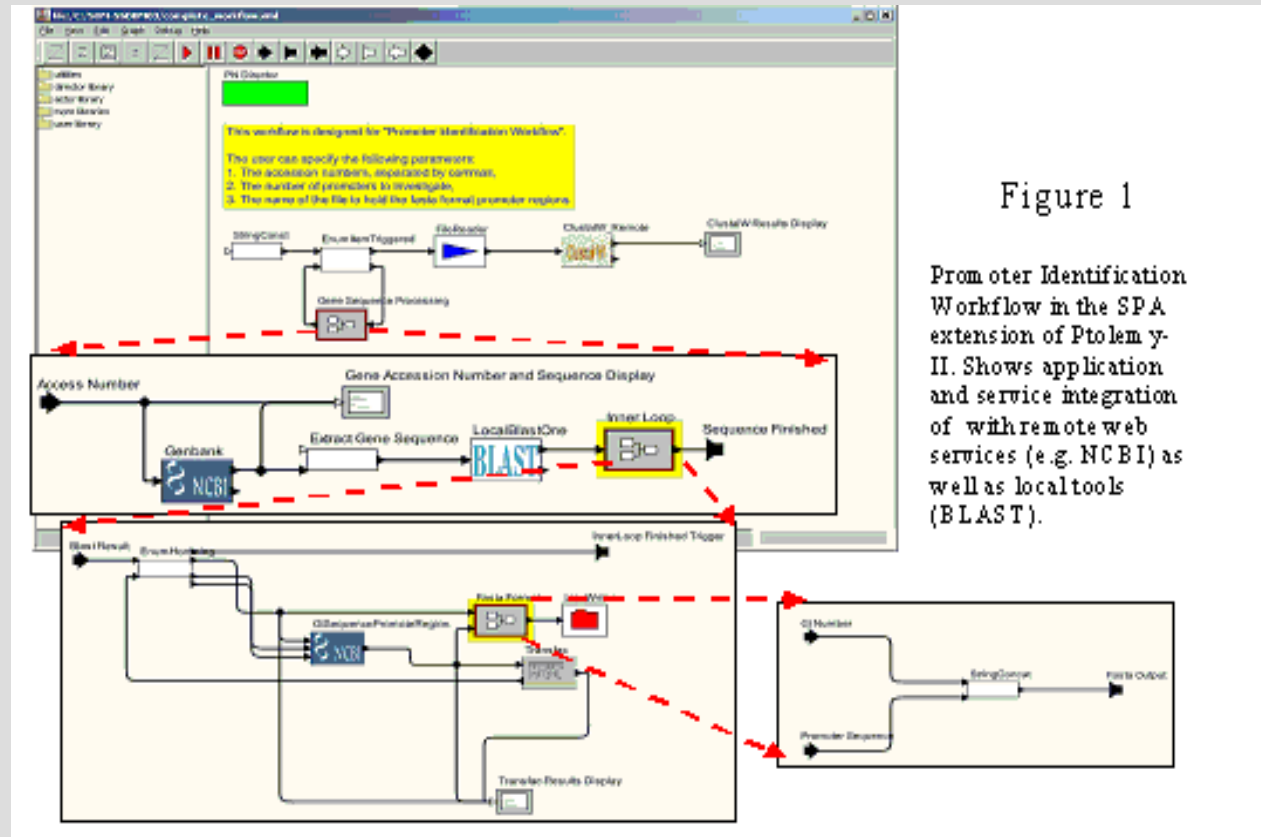
Inclusion d'un workflow

Vue d'ensemble

Kepler/SPA

- Collaboration entre
SEEK (Science Environment for Ecological Knowledge),
SPA (Scientific Process Automation),
GEON (Cyberinfrastructure for the Geosciences),
ROADNet (Real-time Observatories, Applications, and Data Management Network).
- Ajout de briques, essentiellement des services web.

- SPA



3 – *Choix retenu*

Choix retenu : Ptolemy II

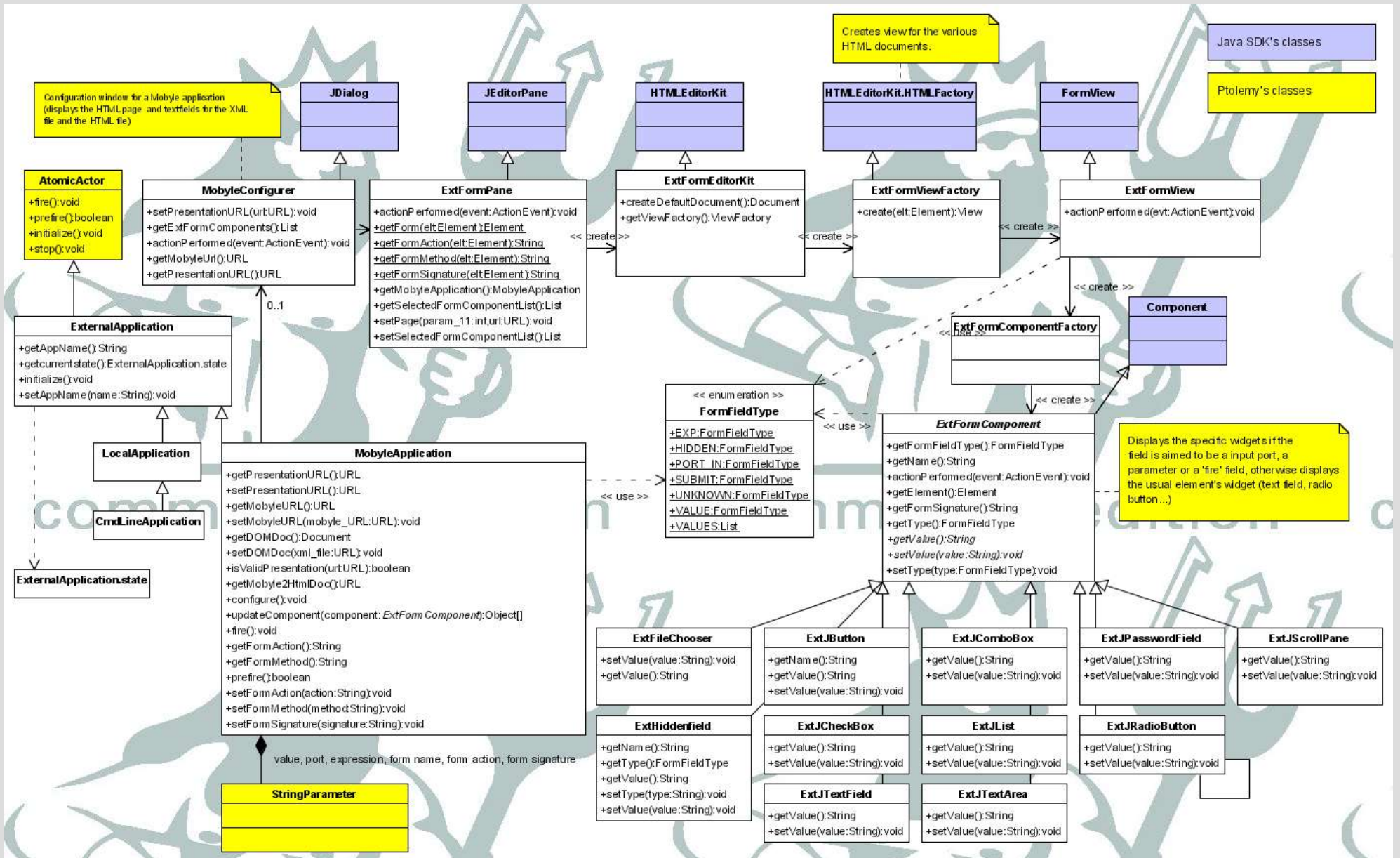
- Rien n'existe, donc :
 - soit partir de zéro,
 - soit adapter une application existante.
- Candidats : Kepler ou Ptolemy
- Kepler :
 - pas spécifique à la bioinformatique,
 - essentiellement ajout de briques.
- Donc...
 - Partir des sources de Ptolemy
 - Ajouter et modifier certains éléments (espace de travail, exécutions partielles, briques, ...)

Avantages et inconvénients

- Application aboutie, pérenne
 - Architecture bien conçue (packages, design patterns)
 - Application graphique peu dépendante du moteur de workflow.
 - Documentation détaillée, diagrammes statiques.
 - Réutilisation des composants de Kepler (pour les services web par exemple).
-
- Grosse application, beaucoup de concepts à assimiler
 - Effort à faire pour intégrer proprement les nouveaux développements (ne pas tout casser pour profiter des évolutions futures).

4 – Fonctionnalités développées

Méthodologie

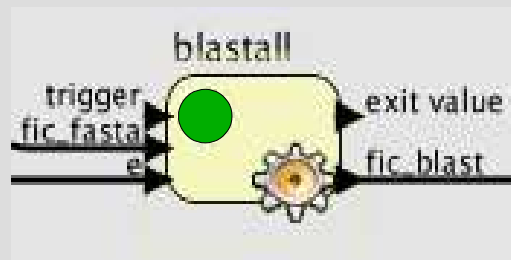


Briques : CmdLineApplication

- But :
 - Exécuter des applications locales (binaires, scripts, etc)
 - Valeur des paramètres définie par :
 - une valeur classique
 - `blast -p blastp -i monfichier -d nr -m8`
 - un port d'entrée
 - `blast -p blastp -i in{nom_port} -d nr -m8`
 - des paramètres de l'espace de travail
 - `blast -p blastp -in in{nom_fichier} -d nr -mexp{format}`
 - Envoi sur des ports de sortie
 - `blast ... -o out{port_sortie}={/home/franck/resu/il2.blast}`

Briques : CmdLineApplication

- souple ! :
 - `blast -p blastp -i exp{repertoire+in{monfichier}} ...`
`-o out{port_sortie}={exp{repertoire+"fichier_resu"}}`
- Suivre l'exécution de la brique.

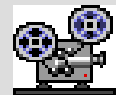
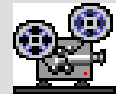


Briques : CheckPoint

But : Visualiser le contenu des fichiers échangés entre les briques.

- pendant ou après l'exécution du workflow
- en choisissant l'application de visualisation

The image shows a workflow diagram at the top and a screenshot of the ClustalX application below. The workflow diagram features a box labeled 'fichier d'alignements' containing the path '/home/franck/tmp/il.blast.fasta'. An arrow points from this box to a green circular node. From this node, an arrow labeled 'trigger' points to a yellow box labeled 'clustalw'. Another arrow labeled 'fichier' points from the 'clustalw' box to a second green circular node. A third arrow labeled 'exit value' points from the 'clustalw' box to the right. A context menu is open over the second green node, with the 'View Datas' option selected. The ClustalX application window is titled 'ClustalX (1.82)' and has a menu bar with 'File', 'Edit', 'Alignment', 'Trees', 'Colors', 'Quality', and 'Help'. Below the menu bar, there are controls for 'Multiple Alignment Mode' and 'Font Size: 10'. The main area of the window displays a multiple sequence alignment of protein sequences, with each sequence starting with a unique identifier (e.g., gi|47682793|gb) and a color-coded alignment of amino acid residues.

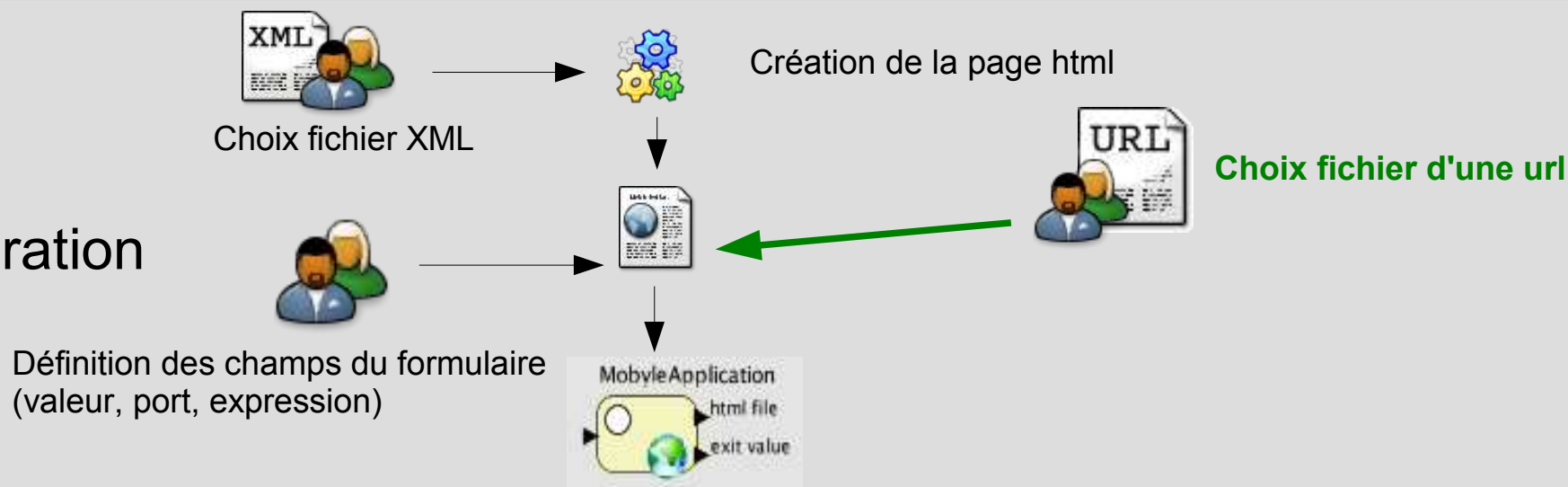


Briques : MobyApplication

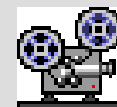
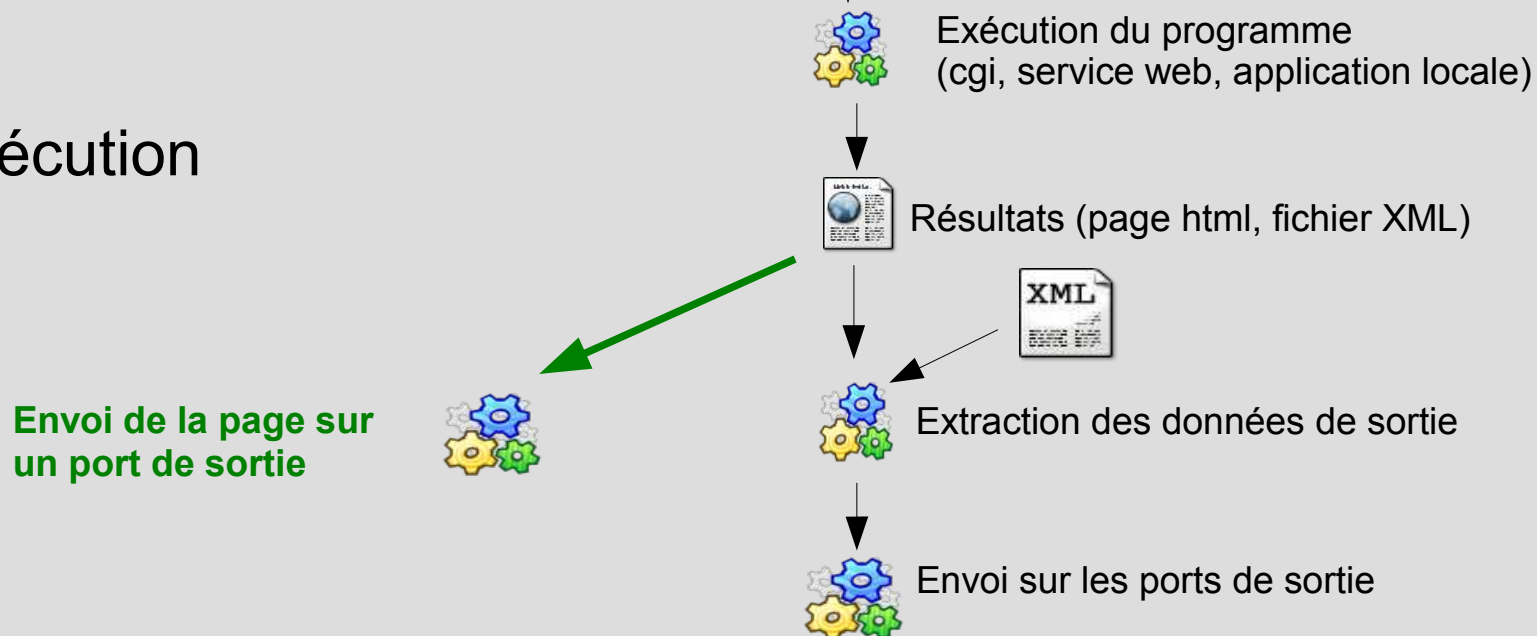
- But
 - Configurer des applications via le formalisme de Moby
- Moby (évolution de Pise)
 - Définition en XML de l'interface du programme
 - Génération de l'interface de configuration

Briques : MobyApplication

Configuration



Exécution



Briques : MobyApplication

- Limitations
 - Beaucoup de sites incompatibles
 - Restriction à HTML 3.2, sans frames, sans javascript
 - Développements importants pour MOBYLE.

5 – Application à la reconnaissance de l'IL-2

Rôle et importance de l'IL-2

- Rappels
 - IL-2 : cytokine sécrétée par les lymphocytes CD4
 - Structure connue
 - reconnue par plusieurs types de récepteurs (IL2R)
 - 3 chaînes membranaires $\alpha\beta\gamma$
 - Structure inconnue
 - pré-assemblage, rôle de l'assemblage non démontrés
- Importance médicale
 - Réduction du développements de tumeurs
 - Reconstitution du taux de CD4 (cas VIH)
 - Effets secondaires

Application à la reconnaissance de l'IL-2

- Laboratoire IGC
 - Mécanisme de transduction induite par l'IL-2 sur les NK.
 - Molécules de substitution de l'IL-2

Futur et réflexions

- Que reste-il à faire ?

- Corriger les bogues
- Menus spécifiques a la bioinformatique
- Execution partielle sur une partie du workflow
- Architecture de la brique MobyApplication
- Definition de fichiers XSLT pour Moby
- Choix d'applications en fonction des données
- Integration des services SoapLab
- Integration des services web dans les menus
- Alternative lors de l'exécution d'un acteur
- Prise en compte des collections de données
- Integration des sources de plusieurs projets
- Execution en arrière plan
- Compilation avec Ant
- Definition et execution dans une applet
- Flux d'entrée, de sortie et d'erreur pour 'CmdLineApplication'
- Ajout de jeux de tests avec Junit
- Affichage des logs, voir la bibliothèque log4j
- Definition de points d'arrêts
- Prendre en compte les pages html 4.0 avec javascript,...
- Ajout des types de données
- Integration des services BioMOBY
- Découverte des services web
- Intégrer un outil de repartition de charge
- Definition de plusieurs workflows dans l'espace de travail
- Integration de briques de visualisation internes
- Gestion des sources et des versions
- Execution d'un workflow en tant que service web
- Procedure d'installation
- Tester les nouvelles briques de Kepler/SPA

PN Director

