

Formation INRA-ACTA-ICTA

Introduction à la modélisation

Les modèles mathématiques pour l'agronomie et l'élevage

2nde session, du 28 novembre au 1^{er} décembre 2005

- Informatique et modèles -

Les choix informatiques à faire pour développer un modèle



Introduction

Introduction

Méthode

→ Démarche de développement

Manière de mener le processus de développement

Technique

→ Choix techniques informatiques

Langages, outils, environnements de développement, architecture

Introduction

Choix informatiques

Il n'y a pas de bons choix dans l'absolu mais
des choix bien adaptés aux besoins.

Les choix informatiques
dépendent des objectifs et des moyens du projet.

Avant de « se lancer » dans le développement,

Analyser les objectifs et moyens du projet

Avant de démarrer

Analyse des objectifs et moyens du projet

Éléments d'analyse des objectifs et moyens du projet

1) Utilisateurs, utilisations

Qui sont les utilisateurs ? Quels sont leur niveau informatique, leur maîtrise du sujet scientifique, leurs préoccupations ?

2) Les suites potentielles

Evolutions potentielles du logiciel dans le futur (couplages, applications logicielles dérivées) ?

Éléments d'analyse des objectifs et moyens du projet

3) Environnement technologique

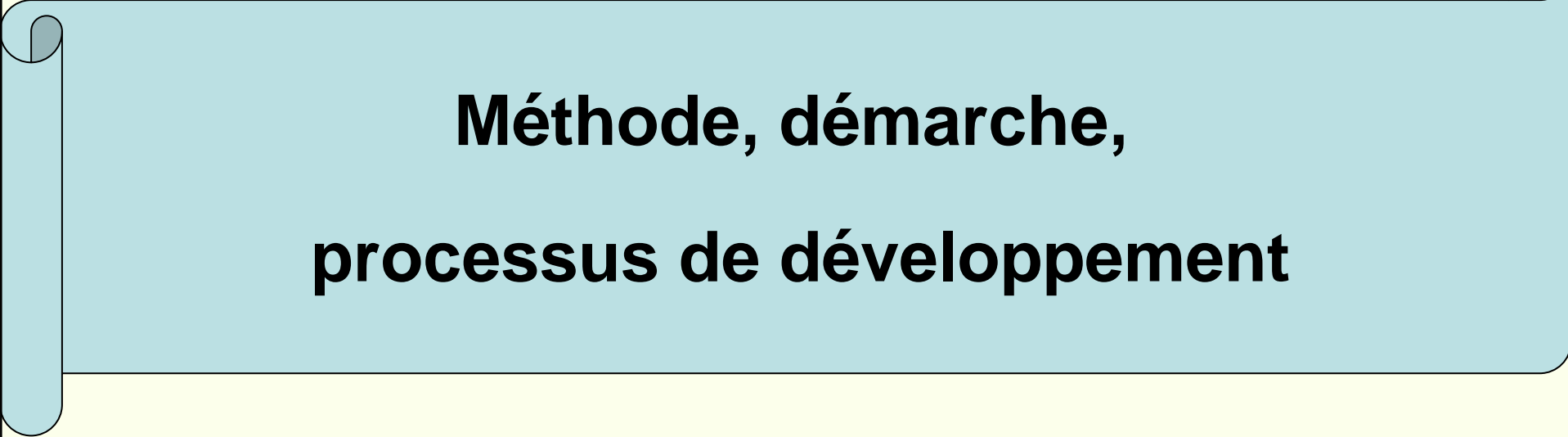
Plate(s)-forme(s) (Windows, Linux ...), compatibilité de langages ?

4) Ressources humaines

Taille et répartition de l'équipe ? Compétences, disponibilité, statut des personnes ?

5) Calendrier

Durée de développement, échéances de livraisons, jalons, contraintes ?



**Méthode, démarche,
processus de développement**

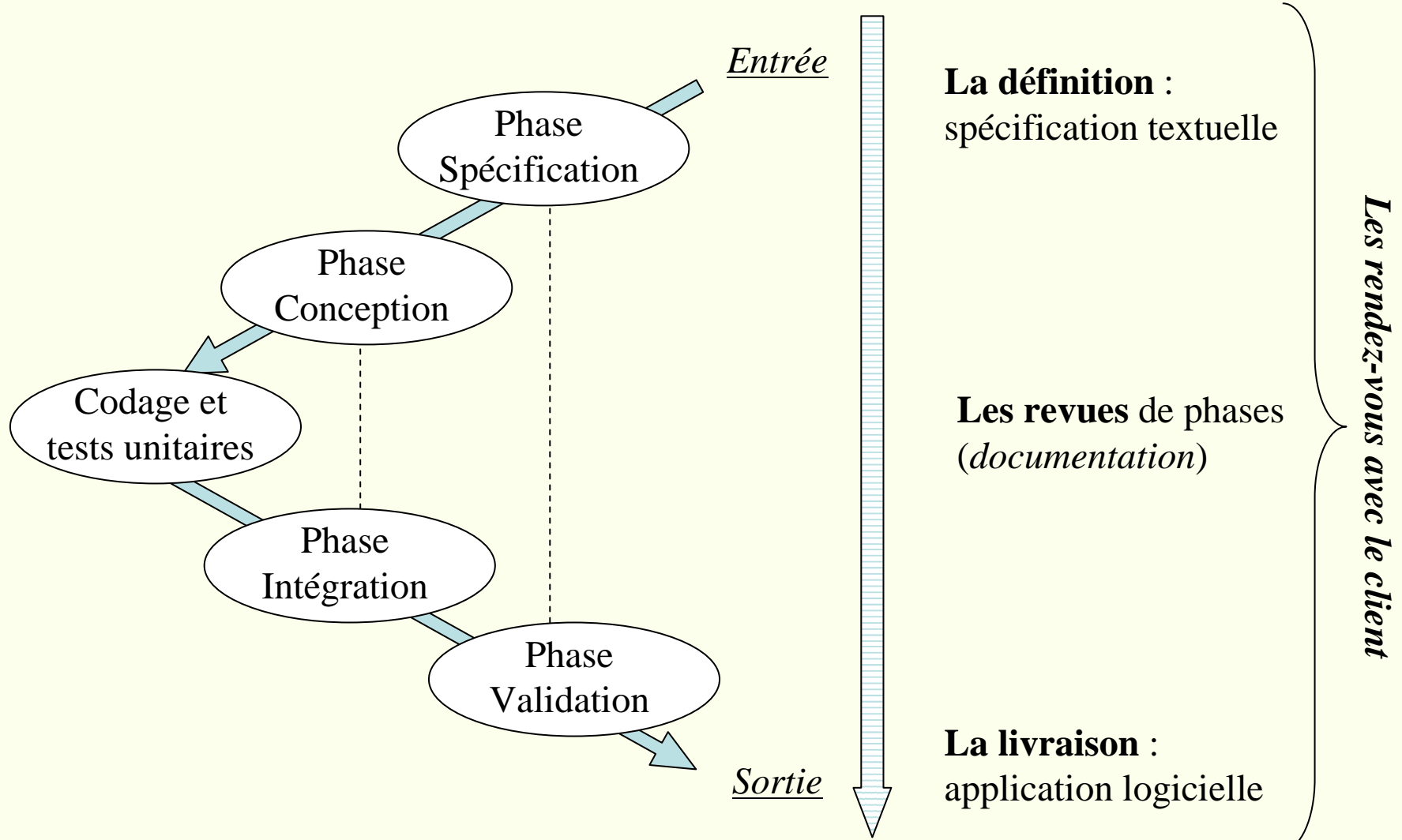
Processus de développement

Une démarche **méthodique**
de **développement informatique**
au profit des
viabilité et fiabilité du logiciel

Différentes approches du développement logiciel :

- Processus **en cascade** (méthode traditionnelle).
- Processus **itératifs** (méthodes agiles).

Méthode en cascade, cycle en V



*... Tests en environnement réel,
Maintenance, diffusion et support*

Fort caractère *itératif* du domaine de la modélisation

Contexte de la recherche

Construction rétroactive du modèle
outil de recherche

Évolutions
modifications

Concevoir le modèle,
Développer le modèle,
Faire tourner le modèle

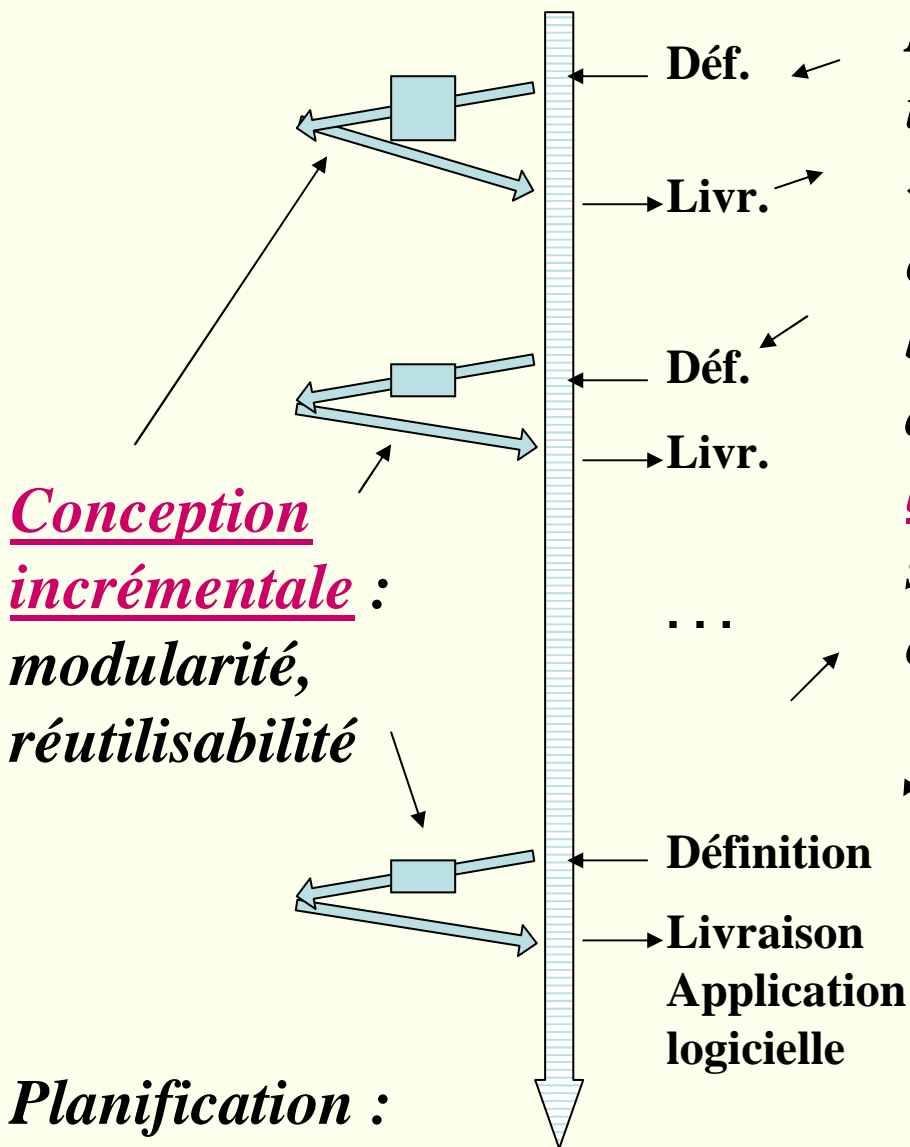
Objectifs multiples

Déclinaisons du logiciel

*Application logicielle
dérivée du modèle*

*Réutilisation avec
d'autres modèles
Couplage, intégration*

Méthodes agiles

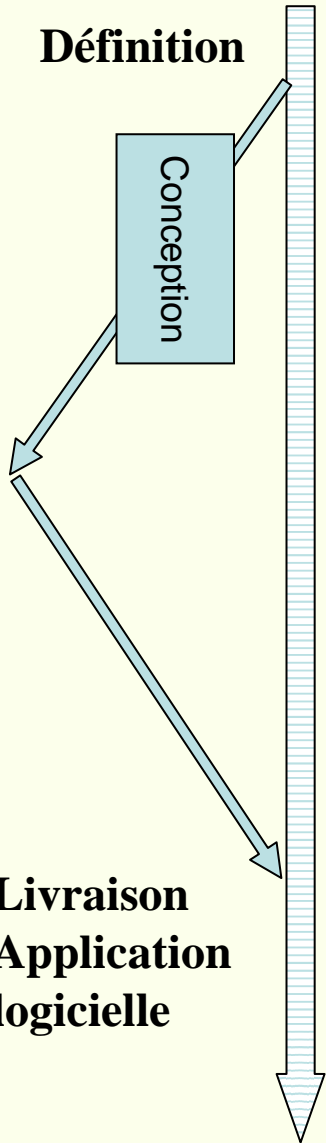


Conception incrémentale :
modularité,
réutilisabilité

Planification :
adaptabilité,
avancement « pas à pas »

*Des échanges incessants avec le « client » :
expression de besoin progressive,
avec l'application logicielle comme support de communication*

Méthode en cascade

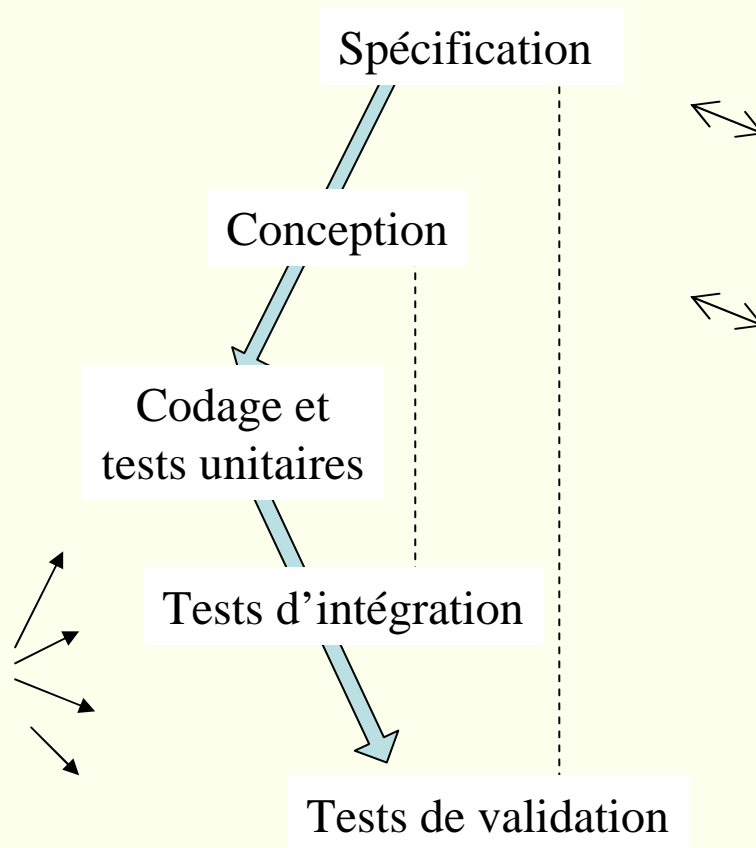


Livraison
Application
logicielle

La documentation et les tests

La documentation et les tests,
des éléments clés des activités logicielles,
qui contribuent à
la **viabilité** et la **fiabilité** du logiciel

La documentation et les tests



La documentation

Outil de définition du logiciel

Donne accès à l'information « que fait le logiciel » sans le faire tourner

Outil de construction du logiciel

Donne une vue d'ensemble architecturale sans avoir à se plonger dans le code

Outil de suivi du logiciel

La Fiche de Version donne connaissance de l'état effectif du logiciel

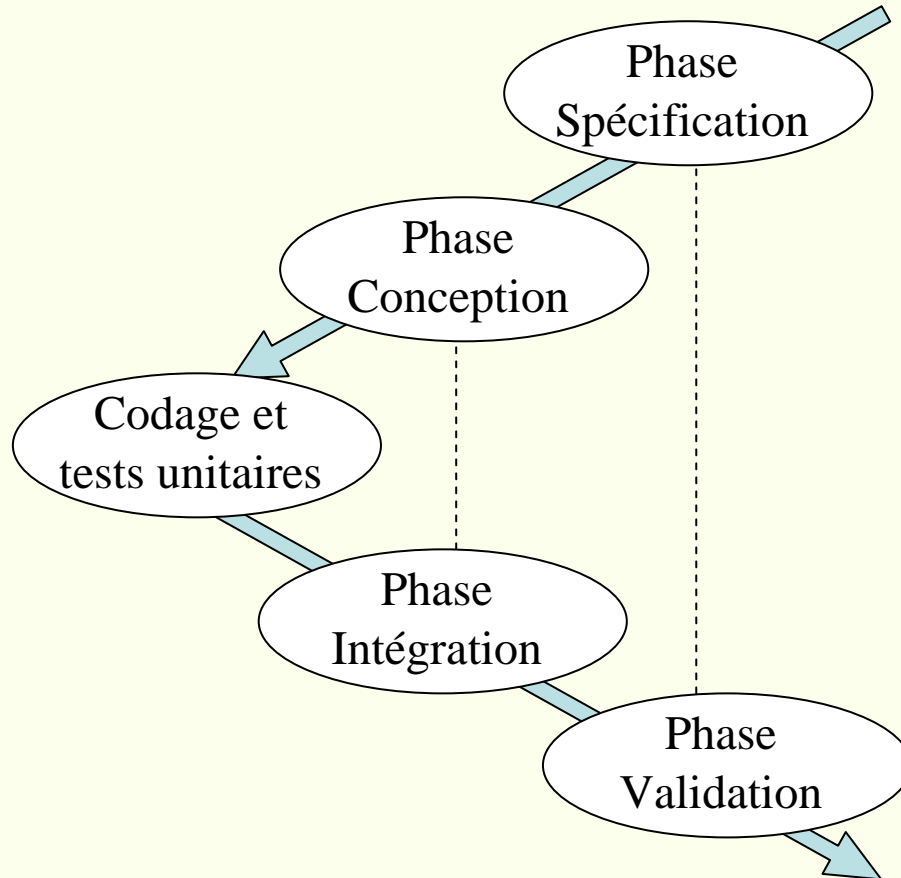
*Maintenance,
corrections et évolutions,
diffusion et support*

Des **tests** propres à chaque phase

La garantie de ce que fait le logiciel, non régression

Connaître et maîtriser le logiciel dans la durée, réutilisabilité

La documentation et les tests



La documentation de spécification

Définition
Idée de réutilisabilité

Les tests de validation

Garantie, fiabilité,
stabilité, non régression

Documentation de définition – illustration –

Représentation du système



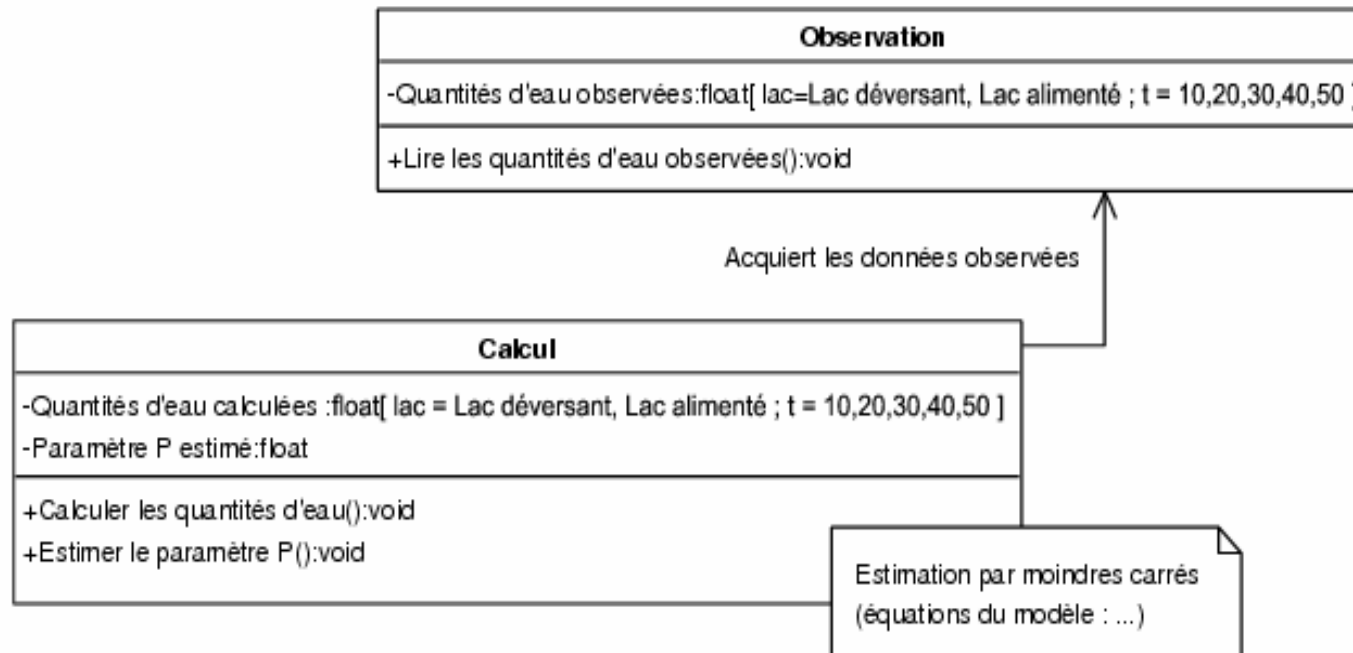
UML Diagramme de classes

Created with Poseidon for UML Community Edition. Not for Commercial Use.

Modélisation conceptuelle du système, conceptualisation du modèle.

Documentation de définition – illustration -

Définition des opérations d'observation et de calcul

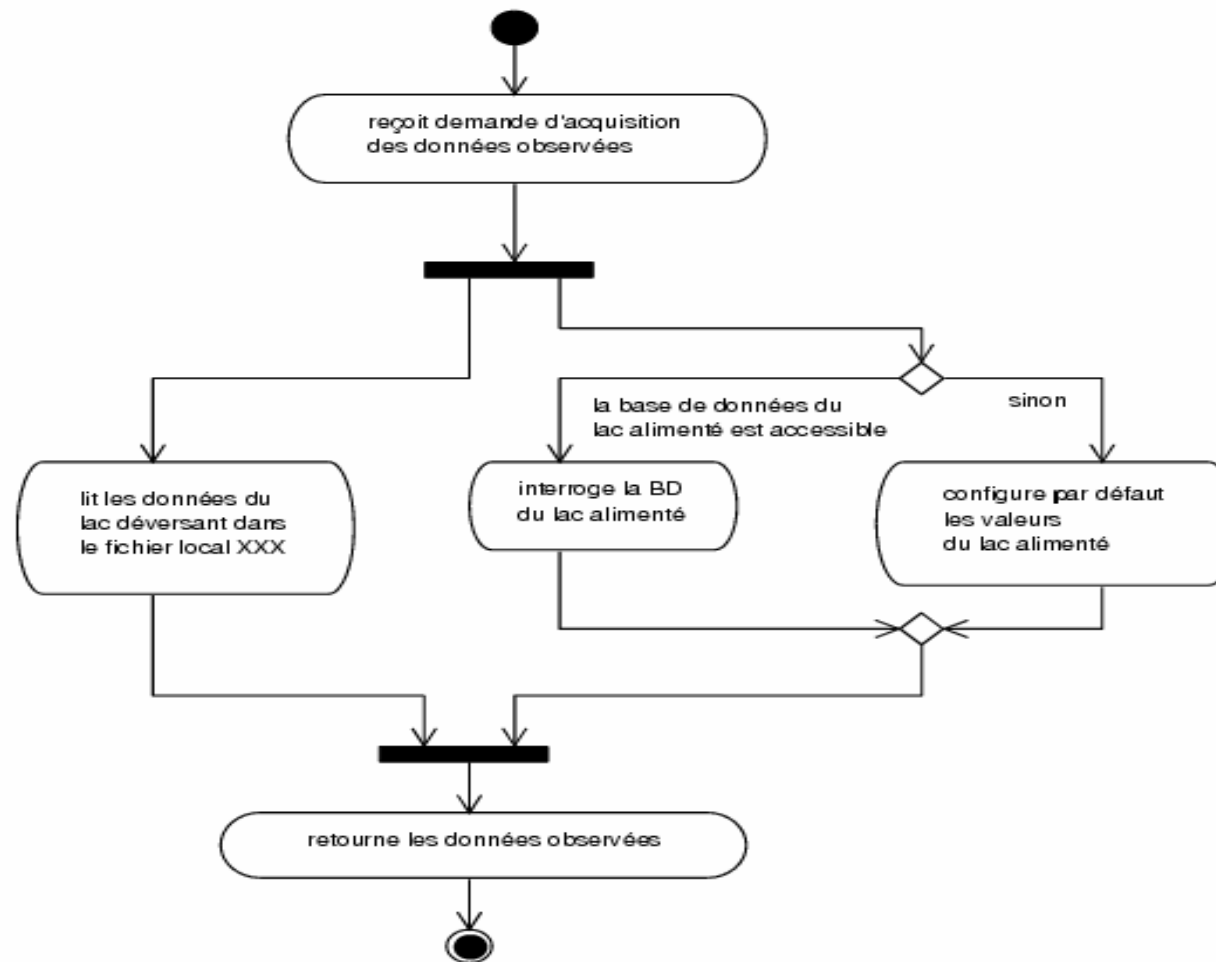


UML Diagramme de classes

Identification des exigences, des fonctionnalités du logiciel.

Documentation de définition – illustration –

Traitement d'une demande d'acquisition des données quantités d'eau observées

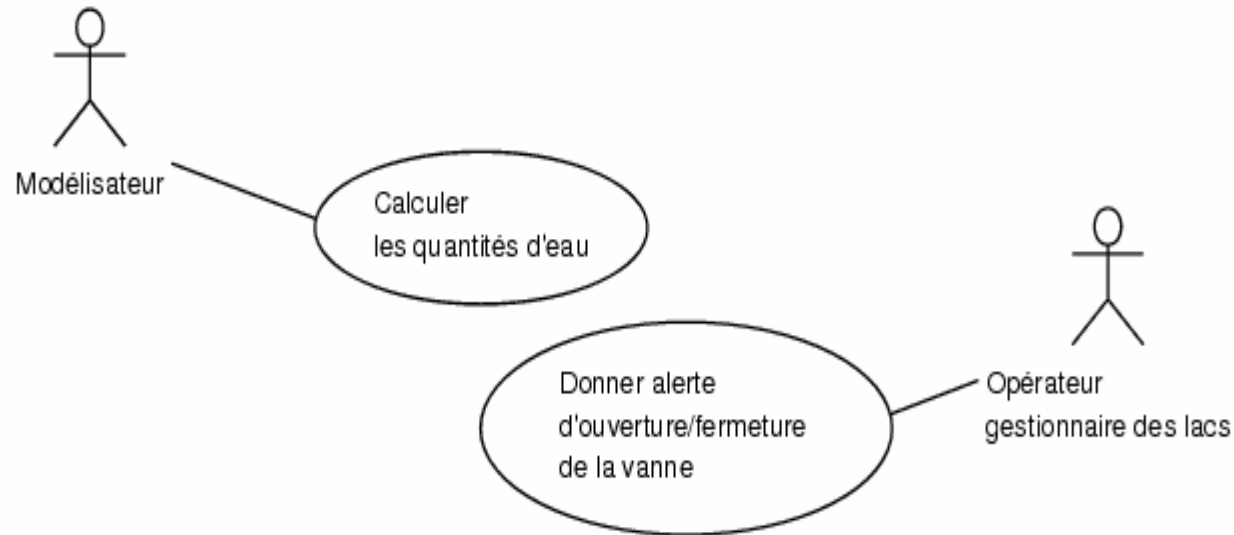


UML Diagramme d'activités

Exigences relatives aux traitements d'acquisition des données d'observation.

Documentation de définition – illustration –

Identification des manières dont le logiciel est utilisé (capture des exigences fonctionnelles)



UML Diagramme de cas d'utilisation

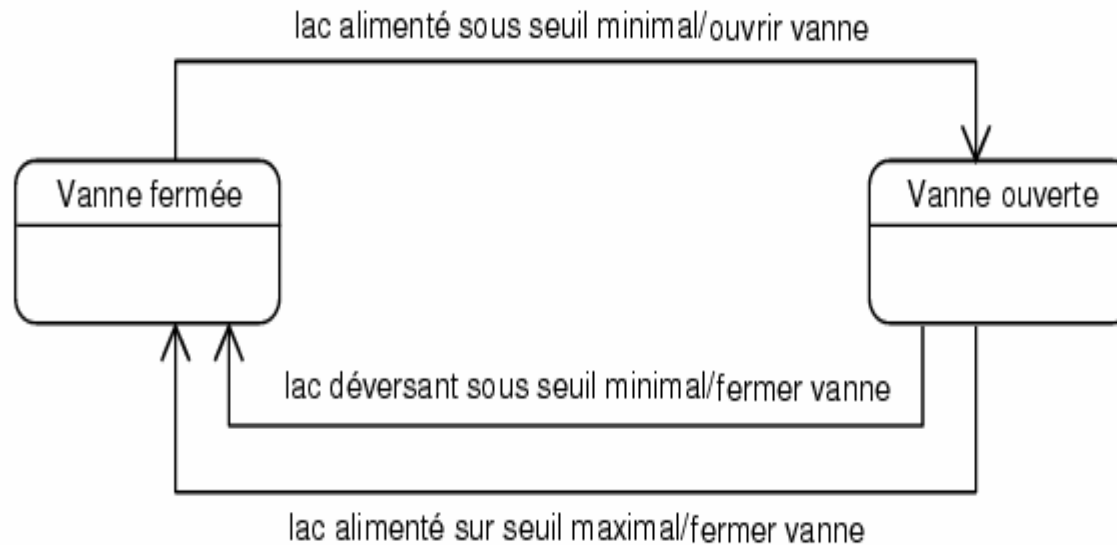
Le premier cas d'utilisation représente le contexte initial de développement du modèle : le modélisateur utilise le modèle dans le cadre de ses travaux de recherche.

Le second cas d'utilisation correspond à une évolution du logiciel : développement d'une application logicielle dérivée du modèle.

Documentation de définition – illustration -

Application logicielle dérivée du modèle

Machine d'états : ouverture/fermeture de la vanne entre le lac déversant et le lac alimenté

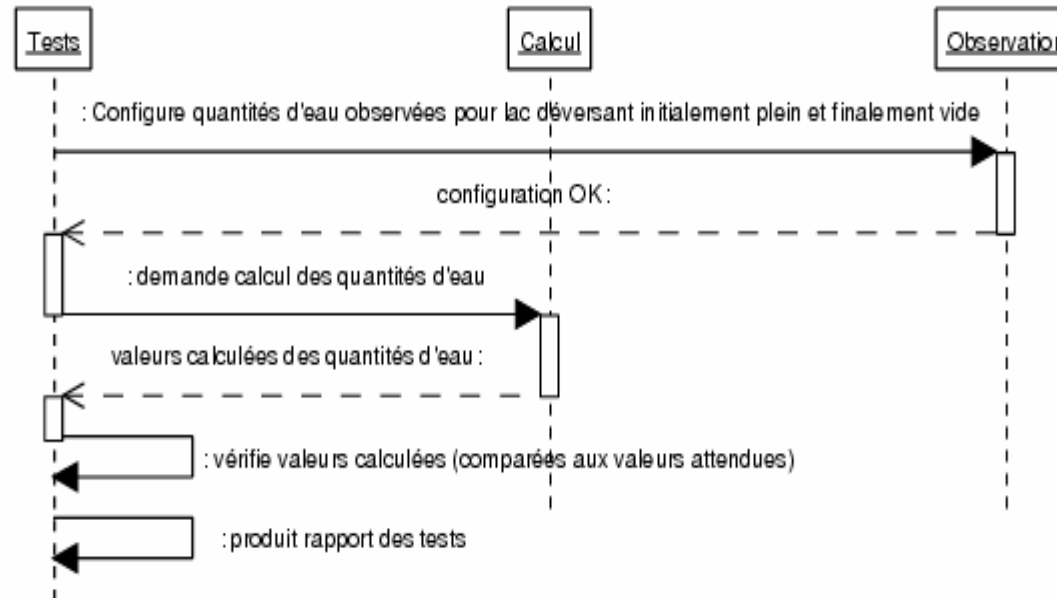


UML Diagramme d'états

Automate décrivant l'état de la vanne gérée par l'application logicielle qui a été greffée sur le modèle (voir en page précédente le second cas d'utilisation).

Documentation de test – illustration –

Scénario de test du vidage complet du lac déversant



UML Diagramme de séquence

Un document de description des tests peut contenir un catalogue de scénarios tels que celui-ci et pour chaque fonctionnalité du logiciel, la liste des scénarios la vérifiant (à rejouer en non régression).

UML « Unified Modeling Language »

UML regroupe **une famille de notations graphiques permettant de décrire et de concevoir un système** (en particulier système logiciel orienté objet).

Utilisation d'UML par le modélisateur et par l'informaticien

- Définition et conception du modèle, conceptualisation du modèle, modélisation conceptuelle de systèmes.
- Spécification et conception logicielle.

En tant que formalisme **commun aux différents acteurs**,
UML facilite les échanges, la communication, la compréhension.



Choix

techniques informatiques

Choix techniques informatiques

L'environnement technique informatique
auquel on a recours pour construire un logiciel
couvre des **langages de programmation**,
des **outils et environnements de développement**,
des **technologies informatiques ...**

Environnement technique informatique - des exemples -

Langages de programmation Fortran, C, java, C++.

Outils et macro-langages S-plus, R.

Outils Excel, ModelMaker, Stella, Vensim.

Langages interprétés python, perl, php.

Outil de compilation gcc.

Environnements de développement et langages Matlab, Scilab.

EDI (Environnements de développement intégrés) Delphi,
Eclipse.

Technologies XML, services Web.

Environnement de programmation de SMA (Systèmes Multi-Agents) Cormas.

Des critères de choix techniques informatiques

1) Prise en main

Il s'agit d'évaluer la facilité de prise en main,
la praticité, la complexité de l'objet

Des critères de choix techniques informatiques

2) Adapté aux besoins scientifiques

Il s'agit d'évaluer si l'objet répond aux préoccupations du modélisateur, s'il est adapté au besoin du modélisateur en terme de **construction** et de **mise au point du modèle**.

Des critères de choix techniques informatiques

3) Adapté aux besoins informatiques

Il s'agit d'évaluer si l'objet répond à des préoccupations informatiques en terme de qualité informatique, **évolutivité et interopérabilité** du logiciel qui est développé.

Des critères de choix techniques informatiques

Interopérabilité

L'interopérabilité est le fait que plusieurs systèmes, qu'ils soient identiques ou radicalement différents, puissent communiquer sans ambiguïté et opérer (travailler) ensemble. L'interopérabilité nécessite de se conformer à des normes clairement établies et univoques.

Choix d'architecture et conception logicielles

Les choix d'environnement technique informatique s'intègrent plus globalement dans les choix d'architecture et conception logicielles.

Des critères de choix techniques informatiques

4) Aspect commercial et confiance dans le produit

Il s'agit d'évaluer si l'objet est de confiance,
pérenne, stable, fiable,
ses conditions et droits d'utilisation.

Des critères de choix techniques informatiques

Comment évaluer ces critères ?

→ **Etudes techniques**

→ **Maquette d'évaluation**

Réaliser une maquette de petite dimension, simple mais représentative de la problématique, afin d'évaluer les performances, d'étudier la faisabilité.

→ **Retours d'expérience**

Recueillir des retours d'expérience et avis d'utilisateurs, développeurs.

Fiches techniques
diffusées par la plate-forme INRA-ACTA-ICTA
<http://www.modelia.org>

UML,
Aspects informatiques juridiques,
Interopérabilité, ouvert, norme, standard,
Environnements de développement informatiques, outils, langages, technologies,
Couplage, intégration de modèles,
Méthodologie, processus de développement, méthode en cascade, méthodes agiles,
Documentation : la documentation d'un projet informatique,
Le développement d'un produit logiciel ...

Des références

Méthode en cascade, méthodes agiles, UML.