

LES OUTILS CASE

5

AGL – Chapitre

Mme. Lilia SFAXI
Mme. Abir Gallas

Les outils CASE

- Les AGL intègrent différents outils d'aide au développement de logiciels

Les outils CASE:

Computer-Aided Software Engineering

Plan du Cours

■ Les Outils CASE

- ✓ Catégories d'AGL
- ✓ Classification des Outils CASE
- ✓ Intégration des Outils CASE
 - Intégration des données
 - Intégration de l'interface
 - Intégration des activités
- ✓ Types d'AGL
 - Environnements de Conception (Upper-CASE)
 - Environnements de Développement (Lower-CASE)

■ Les IDEs

Catégories d'AGL

- Les AGL peuvent être classés selon plusieurs aspects
 - ✓ **Richesse du support:** outils intégrés, aide à la démarche
 - ✓ **Type de problèmes:** logiciels embarqués, temps réel, applications métiers...
 - ✓ **Type de projet d'ingénierie logicielle:** développement logiciel (cf. cycle de vie), intégration de systèmes, système à base de connaissance...
 - ✓ **Ampleur du projet:** complexité, nombre de participants, durée...
 - ✓ **Gestion des ressources du projet:** management des ressources du projet (planification, ordonnancement,...)
 - ✓ **Phase du cycle de développement prise en compte:** conception et/ou développement.

Catégories d'AGL

- Certains outils interviennent durant la totalité du processus logiciel
- **Outils horizontaux**
 - ✓ Certains outils interviennent lors d'une ou plusieurs phases du cycle de vie du logiciel
- **Outils verticaux**

Outils Horizontaux

- Service pour l'ensemble du cycle de vie
 - ✓ Editeurs de texte
 - ✓ Gestion de projet
 - ✓ Gestion du dictionnaire de données
 - ✓ Administration et droit d'accès
 - ✓ Gestion des configurations
 - ✓ Documentation
 - ✓ Service de communication

Outils Horizontaux

Activité	Type d'outils
Gestion de Projet	Outils de planification, calcul de coûts, suivi de projet, tableur...
Outils Collaboratifs	Gestion des utilisateurs, droits d'accès, communication...
Documentation	Éditeurs de texte, d'images, de diagrammes...
Gestion de Configuration et Contrôle des versions	Analyse sémantique du code, de référentiel de composants objets, de manipulation de contenus web

Outils Verticaux

- Fonctions propres à chaque étape du cycle de vie
 - ✓ Spécification
 - ✓ Conception
 - ✓ Génération de code
 - ✓ IDE
 - ✓ Compilateurs
 - ✓ Génération d'IHM
 - ✓ Génération de tests
 - ✓ Validation
 - ✓ Prototypage
 - ✓ Maintenance

Fonctions Supplémentaires

- Rétro-Ingénierie (*Reverse Engineering*)
 - ✓ Etudier un objet pour en déterminer le fonctionnement interne ou la méthode de fabrication
- Réutilisation (*Reuse*)
 - ✓ Gestion des bibliothèques de composants

Classification des Outils CASE

IHM

Outils de Planning

Outils d'Analyse

Outils de Conception

Outils de Réalisation

Outils de Tests

Outils d'Evolution

Outils de Traces

Outils de Documentation

Outils de Gestion de Configuration

Outils de Gestion de Projet

Dictionnaire des Données

Système d'Exploitation

Plan du Cours

■ Les Outils CASE

- ✓ Catégories d'AGL
- ✓ Classification des Outils CASE
- ✓ **Intégration des Outils CASE**
 - Intégration des données
 - Intégration de l'interface
 - Intégration des activités
- ✓ Types d'AGL
 - Environnements de Conception (Upper-CASE)
 - Environnements de Développement (Lower-CASE)

■ Les IDEs

Intégration d'Outils CASE

- Un AGL intègre différents outils CASE de manière à les faire coopérer de façon uniforme
- Cette intégration peut s'effectuer à trois niveaux:
 - ✓ Intégration des données
 - ✓ Intégration de l'interface utilisateur
 - ✓ Intégration des activités

Intégration des Données

- Les outils CASE manipulent des données
 - ✓ Génération
 - ✓ Utilisation
 - ✓ Transformation
- Exemple
 - ✓ Spécification
 - ✓ Modèle conceptuel des données
 - ✓ Jeux de test
 - ✓ Code
 - ✓ Manuel utilisateur...

Intégration des Données

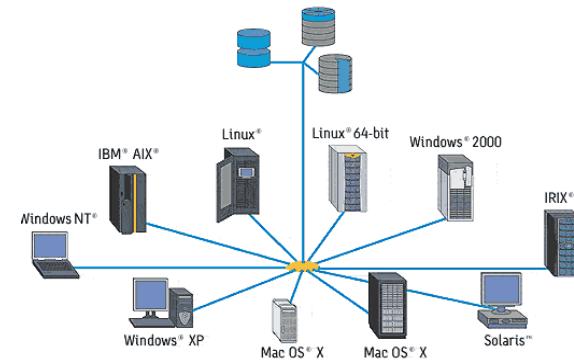
- Différents outils sont amenés à partager une même donnée, exemples:
 - ✓ Les tables générées par un éditeur de diagrammes sont utilisées par un SGBD
 - ✓ Le code généré par un éditeur de texte est compilé par un compilateur
 - ✓ A partir d'une spécification algébrique on peut générer des jeux de tests
 - ✓ ...

Intégration des Données

- Un AGL doit prendre en charge la communication de ces données entre les différents outils
 - ✓ Intégration des outils CASE par celle des données
- L'intégration des données peut être
 - ✓ Physique
 - ✓ Logique

Intégration des Données

- Intégration Physique
 - ✓ tous les outils de l'AGL utilisent un seul format de représentation des données
 - Ex: des fichiers sur une même machine.
 - ✓ Cette approche implique que tous les outils de l'AGL connaissent la structure logique (l'organisation) des fichiers qu'ils sont amenés à utiliser
- Il est nécessaire de normaliser la structure logique des fichiers



Intégration des Données

- Intégration Logique
 - ✓ Utilisation d'un système de gestion des objets
 - Gestion automatique des différentes entités et leurs inter-relations
 - ✓ Cette approche nécessite la définition des différents types de données manipulées
 - ✓ L'AGL doit également gérer la cohérence entre les différentes versions de ces données
 - Gestion des configuration

Plan du Cours

■ Les Outils CASE

- ✓ Catégories d'AGL
- ✓ Classification des Outils CASE
- ✓ Intégration des Outils CASE
 - Intégration des données
 - **Intégration de l'interface**
 - Intégration des activités
- ✓ Types d'AGL
 - Environnements de Conception (Upper-CASE)
 - Environnements de Développement (Lower-CASE)

■ Les IDEs

Intégration de l'Interface Utilisateur

- Tous les outils intégrés dans l'AGL communiquent avec l'utilisateur selon un schéma uniforme
 - ✓ Facilite leur utilisation
- Intégration des outils CASE par celle des interfaces

Plan du Cours

■ Les Outils CASE

- ✓ Catégories d'AGL
- ✓ Classification des Outils CASE
- ✓ Intégration des Outils CASE
 - Intégration des données
 - Intégration de l'interface
 - **Intégration des activités**
- ✓ Types d'AGL
 - Environnements de Conception (Upper-CASE)
 - Environnements de Développement (Lower-CASE)

■ Les IDEs

Intégration des Activités

- Un AGL peut gérer le séquencement des appels aux différents outils intégrés
 - ✓ Pour assurer la cohérences entre les différentes phases du processus logiciel.
- Cet aspect implique que l'on dispose d'un modèle du processus de développement bien accepté et partagé.
- Intégration des outils CASE par celle des activités

Plan du Cours

■ Les Outils CASE

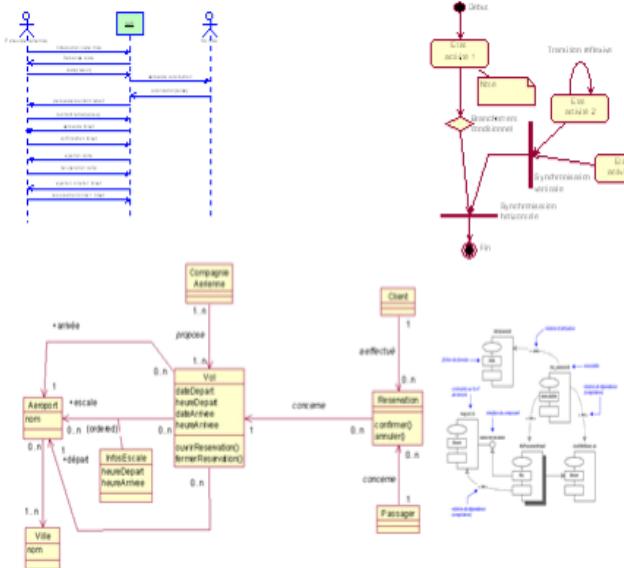
- ✓ Catégories d'AGL
- ✓ Classification des Outils CASE
- ✓ Intégration des Outils CASE
 - Intégration des données
 - Intégration de l'interface
 - Intégration des activités
- ✓ Types d'AGL
 - Environnements de Conception (Upper-CASE)
 - Environnements de Développement (Lower-CASE)

■ Les IDEs

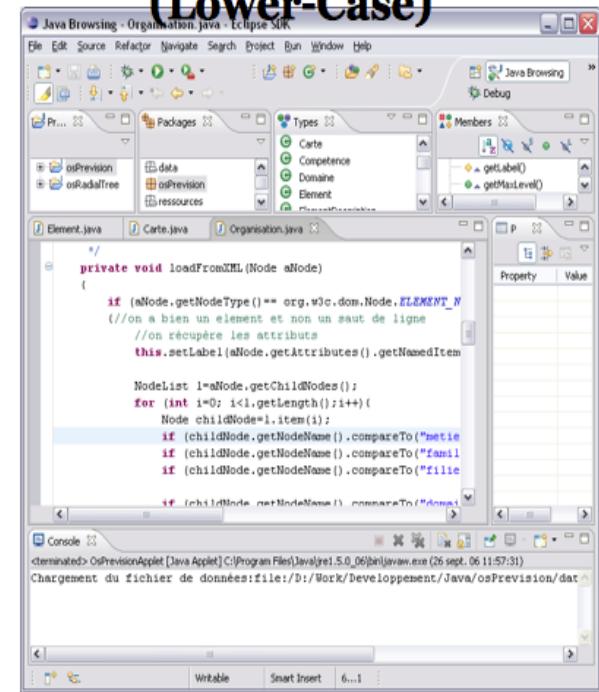
Types d'AGL

- On distingue essentiellement deux types d'AGL selon la nature des outils intégrés

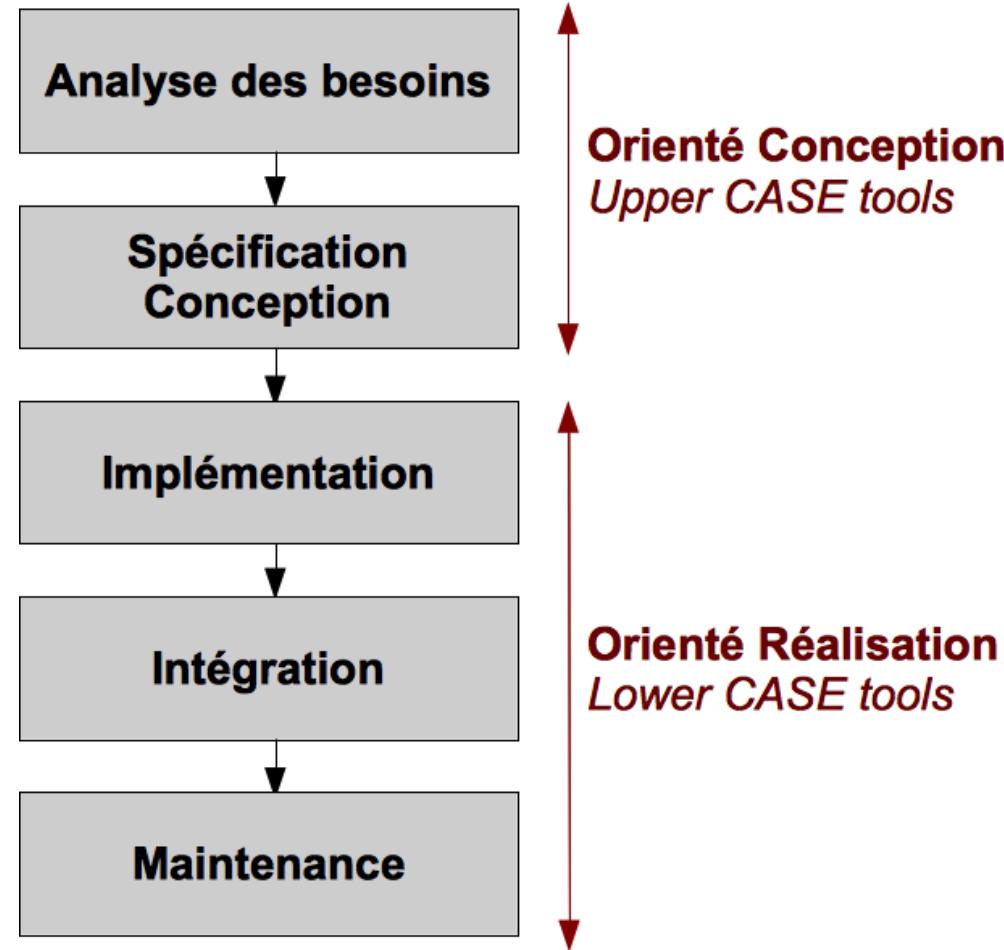
Les environnements de Conception (Upper-Case)



Les environnements de Développement (Lower-Case)



Types d'AGL



Environnements de Conception (Upper-CASE)

- Supportent les phases d'analyse et de conception du processus logiciel
- Généralement basés sur une méthode d'analyse et de conception
- Exemple: Rational Rose, AnalyseSI
- Intègrent généralement
 - ✓ Des outils pour l'édition de diagrammes
 - ✓ Des dictionnaires de données
 - ✓ Des outils pour d'édition de rapports
 - ✓ Des générateurs de squelettes de code
 - ✓ Des outils de prototypage
 - ✓ ...

Environnements de Développement (Lower-CASE)

- Supportent les phases d'implémentation et de test du processus logiciel.
- Exemple : Code::Blocks, KompoZer
- Intègrent généralement
 - ✓ Des éditeurs (éventuellement dirigés par la syntaxe),
 - ✓ Des générateurs d'interfaces homme/machine,
 - ✓ Des SGBD,
 - ✓ Des compilateurs,
 - ✓ Des optimiseurs,
 - ✓ Des debuggers,
 - ✓ ...

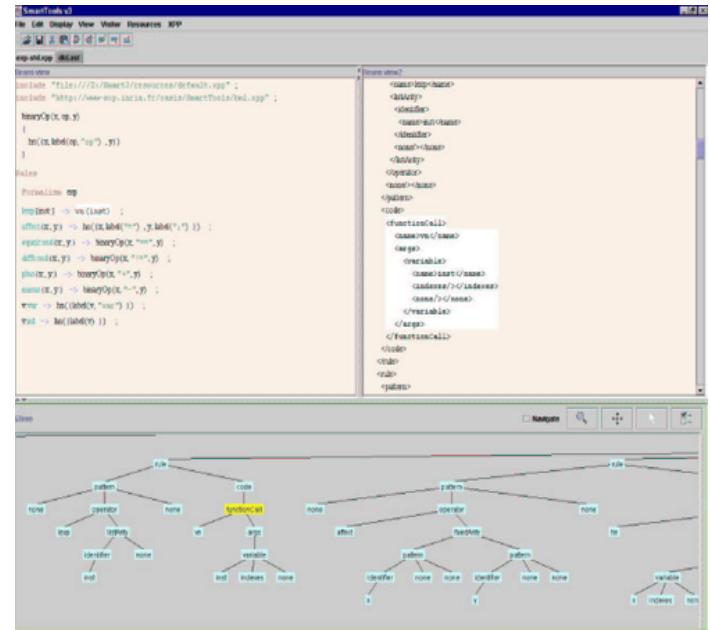
Environnements de Développement (Lower-CASE)

■ Environnements dédiés

- ✓ Certains environnements, plus évolués, sont dédiés à un langage particulier
- ✓ Exemple: C++ Builder, Smalltalk
- ✓ Ces différents environnements proposent
 - Des bibliothèques de composants
 - Une interface graphique
 - Des éditeurs dédiés au langage
 - Des interprètes
 - Des debuggers
 - ...

Environnements de Développement (Lower-CASE)

- Générateurs d'environnements de programmation:
 - ✓ À partir de la description formelle d'un langage, ils génèrent un environnement de programmation dédié au langage
 - ✓ Contenant:
 - un éditeur dédié au langage,
 - un pretty-printer,
 - un debugger,
 - un interpréteur, ...
 - ✓ Exemple:
 - Centaur
 - SmartTools



Plan du Cours

■ Les Outils CASE

- ✓ Catégories d'AGL
- ✓ Classification des Outils CASE
- ✓ Intégration des Outils CASE
 - Intégration des données
 - Intégration de l'interface
 - Intégration des activités
- ✓ Types d'AGL
 - Environnements de Conception (Upper-CASE)
 - Environnements de Développement (Lower-CASE)

■ Les IDEs

Integrated Development Environment

- IDE: Environnement intégré de développement
 - ✓ Ensemble d'outils qui facilitent le développement d'application
- Permettent de:
 - ✓ Ecrire et éditer le code source
 - ✓ Voir les erreurs pendant la saisie
 - ✓ Voir la syntaxe spécifique du code
 - ✓ Automatiser les tâches réitérées
 - ✓ Compiler le code
 - ✓ Accéder à la documentation, exemples...
 - ✓ Offrir des utilitaires *drag & drop* pour la création facile des dispositifs tels que les objets graphiques, ou créer des connexions avec les bases de données

Integrated Development Environment

- En outre, quelques IDEs font les tâches suivantes
 - ✓ Fournir des templates pour la création rapide des composantes
 - ✓ Fournir le *code completion* au moment de la saisie
 - ✓ Produire automatiquement du code, des rapports, des connexions avec les données...
 - ✓ S'intégrer avec des dépôts de code source, tels que CVS
 - ✓ S'intégrer avec des serveurs d'application tel que Apache Tomcat
 - ✓ S'intégrer avec des utilitaires de construction tels que Apache Ant.

Avantages d'un IDE

- Productivité
 - ✓ Un développement plus rapide
 - ✓ Peu d'erreur de syntaxe et de structure de code (améliorer le code et corriger plus vite les erreurs)
 - ✓ Aide des programmeurs inexpérimentés
- Consistance
 - ✓ Structure du code consistante
 - ✓ *Look & Feel*
- Courbe d'apprentissage accélérée
 - ✓ Possibilité de vérifier les résultats immédiatement
 - ✓ Accès immédiat à l'aide de contexte, aux portions de code visées

Inconvénients d'un IDE

- Coût des outils
- Temps d'exécution
- Attachement à un module de base de données, à une plateforme,...
- Manque de personnel expérimenté et des ressources
- L'incapacité de certains IDEs à fournir toutes les possibilités de langages

Quand utiliser un IDE?

- L'utilisation d'un IDE n'est pas toujours appropriée
- Un IDE est le plus utile quand:
- La vitesse de développement est un facteur
- Les coûts de développement et de cycle de vie sont un facteur
- De grandes équipes travaillent sur un même projet
- Les demandes de l'application ne dépassent pas les limites de l'environnement de déploiement de l'IDE utilisé.

Langages des IDEs

- Certains IDEs sont basés sur des langages spécifiques, tels que
 - ✓ Eclipse ou Netbeans: basés sur Java
 - ✓ MonoDevelop: basé sur C#
- L'ajout des langages alternatifs est souvent fourni par des *plugins*, leur permettant d'être installés sur le même IDE en même temps
 - ✓ Exemple: Plugin C/C++ pour Eclipse