
Glossaire

Diagramme de classes

Le diagramme de classes est un modèle utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations. Ce diagramme fait partie de la partie des diagrammes structurels (point de vue statique), ne s'intéressant pas aux aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe. Une classe est un ensemble de méthodes et de données (attributs) qui sont liées par un champ sémantique.

Les classes peuvent être reliées grâce au mécanisme d'héritage qui permet de mettre en évidence des relations de parenté. D'autres relations (association ou composition) sont possibles entre des classes, représentées par un arc spécifique dans le diagramme de classes.

Les diagrammes de classes peuvent être utilisés à différents moments du processus de développement logiciel :

- en phase d'analyse et de conception, pour comprendre un besoin et/ou pour définir le fonctionnement d'une application ;
- en phase de développement, pour définir la structure de l'application.

Un diagramme de classes peut également être fait pour faire de la rétro-ingénierie.

En fonction de l'utilisation du diagramme de classes, il faut adapter le niveau de précision des différents éléments (packages, visibilités, typages, navigabilité des associations...).

Référence :

https://fr.wikipedia.org/wiki/Diagramme_de_classes

Environnement de Développement Intégré (IDE)

Un environnement de développement (en anglais, *Integrated Development Environment*, abrégé en IDE) est un ensemble d'outils qui permet d'augmenter la productivité des développeurs qui conçoivent des logiciels en simplifiant les différentes opérations du développement logiciel. Il comporte, notamment, un éditeur de texte destiné à la programmation, des fonctions qui permettent, par pression sur un bouton, de démarrer le compilateur ou l'éditeur de liens ainsi qu'un débogueur. Mais, il peut aussi comprendre des outils de planification des tâches, de gestion des versions, de prototypage d'interfaces graphiques...

Quelques exemples :

- Eclipse : [https://fr.wikipedia.org/wiki/Eclipse_\(projet\)](https://fr.wikipedia.org/wiki/Eclipse_(projet))
- IntelliJ IDEA : https://fr.wikipedia.org/wiki/IntelliJ_IDEA
- Microsoft Visual Studio : https://fr.wikipedia.org/wiki/Microsoft_Visual_Studio
- XCode : <https://fr.wikipedia.org/wiki/Xcode>
- NetBeans : <https://fr.wikipedia.org/wiki/NetBeans>

Référence :

https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement

Logiciel d'automatisation de tâches

Un logiciel d'automatisation de tâches est un logiciel dont la fonction principale consiste à automatiser (ordonnancer et piloter) l'ensemble des actions (préprocessing, compilation, éditions de liens, packaging, déploiement, exécution des tests, génération de la documentation...) contribuant, à partir de données sources, à la production d'un logiciel opérationnel. L'ensemble des actions sont généralement décrites dans un script.

Exemples :

Outils	Script	Liens
Make	Makefile	https://fr.wikipedia.org/wiki/Make
Apache Ant	build.xml	https://ant.apache.org/ https://fr.wikipedia.org/wiki/Apache_Ant
Apache Maven	pom.xml	https://maven.apache.org/ https://fr.wikipedia.org/wiki/Apache_Maven
Gradle	build.gradle	https://gradle.org/ https://fr.wikipedia.org/wiki/Gradle

Référence :

https://fr.wikipedia.org/wiki/Moteur_de_production

Logiciel de gestion de versions

Un logiciel de gestion de versions (en anglais *Version Control System*, abrégé en VCS) est un logiciel qui permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectuées dessus. Il permet, notamment, en développement logiciel de conserver le code source relatif aux différentes versions d'un logiciel.

Exemples :

- Git : <https://fr.wikipedia.org/wiki/Git>
- Mercurial : <https://fr.wikipedia.org/wiki/Mercurial>
- Apache Subversion ou SVN : https://fr.wikipedia.org/wiki/Apache_Subversion

Référence :

https://fr.wikipedia.org/wiki/Logiciel_de_gestion_de_versions

Modèle

En informatique, un modèle permet de modéliser la conception ou le fonctionnement d'un ensemble de programmes informatiques.

« Un modèle est une abstraction... une représentation... une simplification d'un système étudié. » [FEB06]

« A model is a simplification of a system built with an intended goal in mind. » [BG01]

« A model is a description or specification of that system and its environment for some certain purpose. » [MM03]

« A model is an abstraction of some aspect of a system. The system described by a model may or may not exist at the time the model is created. Models are created to serve particular purposes, for example, to present a human understandable description of some aspect of a system or to present information in a form that can be mechanically analyzed. » [FR07]

Références :

- [FEB06] Jean-Marie Favre, Jacky Estublier et Mireille Blay-Fornarino. L'ingénierie dirigée par les modèles : au-delà du MDA. Cachan, France : Hermes-Lavoisier, fév. 2006.
- [BG01] Jean Bézivin and Olivier Gerbé. 'Towards a Precise Definition of the OMG/MDA Framework'. In: ASE'01: Proceedings of the 16th IEEE international conference on Automated Software Engineering. Washington, DC, USA: IEEE Computer Society, 2001, pp. 273-280.
- [MM03] Joaquin Miller and Jishnu Mukerji. MDA Guide Version 1.0.1. Tech. rep. Object Management Group (OMG), 2003.
- [FR07] Robert France and Bernhard Rumpe. 'Model-driven Development of Complex Software: A Research Roadmap'. In: FOSE '07: 2007 Future of Software Engineering. Washington, DC, USA: IEEE Computer Society, 2007, pp. 37-54.

Système de suivi de bugs

Un système de suivi de bugs (en anglais, Bug Tracking System) est un logiciel qui permet d'effectuer un suivi des bugs signalés dans le cadre d'un projet de développement logiciel. Il permet d'aider les utilisateurs et les développeurs à améliorer la qualité d'un logiciel.

Le plus souvent, le système utilise un dispositif de tickets qui permet à l'utilisateur qui a indiqué un défaut de suivre l'évolution de son traitement. Les tickets sont généralement affectés d'un indicateur du degré d'urgence de résolution du défaut : critique, urgent, normal ou non urgent par exemple. Cet indicateur est fourni par l'utilisateur ayant rapporté le défaut et il peut être modifié par l'administrateur du projet.

Exemples :

- Jira : <https://fr.wikipedia.org/wiki/Jira>
- Bugzilla : <https://fr.wikipedia.org/wiki/Bugzilla>
- GLPI : https://fr.wikipedia.org/wiki/Gestionnaire_Libre_de_Parc_Informatique
- Redmine : <https://fr.wikipedia.org/wiki/Redmine>

Référence :

https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_suivi_des_bugs
