

Chapitre II

conception des

logiciels embarqués



Chapitre II conception des logiciels embarqués

II.1. Conception des systèmes embarqués :

Les systèmes embarqués de première génération étaient suffisamment simples pour que leur conception ne requière pas de méthodologie particulière : quelques essais-erreurs pouvaient suffire pour satisfaire aux contraintes. Avec la seconde génération, la complexité est devenue trop importante pour que la conception puisse être menée à bien sans méthode. Un premier type de flot de conception a donc été utilisé, inspiré par les flots de conception pour les systèmes généralistes. Ce flot est présenté dans la première section. La troisième génération remet en question ce type de flot, et de nouvelles méthodes émergent. Elles sont présentées dans la section suivante.



Chapitre II conception des logiciels embarqués

II.1. Conception des systèmes embarqués :

II.1.1. Flots classiques de conception des systèmes embarqués

II.1.1.1. Le flot :

Le flot classique de conception des systèmes embarqués est représenté sur les **figures suivantes**. Ce flot part d'une spécification souvent informelle du système. Il distingue immédiatement les parties logicielles des parties matérielles. Ces parties sont développées indépendamment l'une de l'autre par des équipes différentes. A la fin, une équipe d'intégration assemble les parties, ce qui pose souvent des problèmes d'incompatibilité. Cette intégration donne directement un prototype à tester. En cas d'erreur, il peut être nécessaire de recommencer complètement le flot.

Dans un tel flot il peut être difficile de développer complètement le logiciel sans que le matériel soit défini. C'est pour cela que son développement devait souvent attendre que la partie matérielle soit décrite pour être achevé.



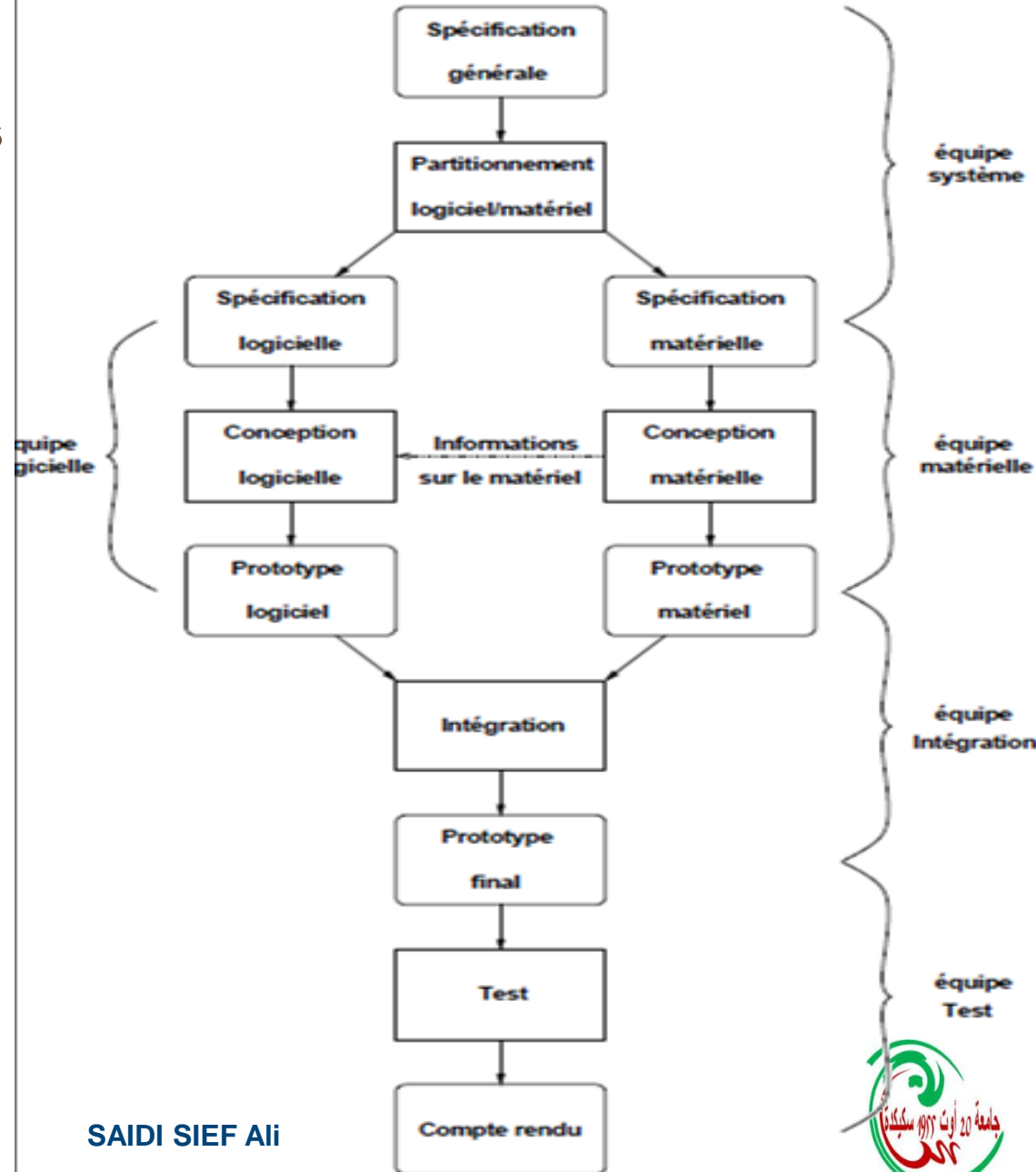
Chapitre II conception des logiciels embarqués

II.1. Conception des systèmes embarqués :

II.1.1. Flots classiques de conception des systèmes embarqués

II.1.1.1. Le flot :

Flot de conception Classique



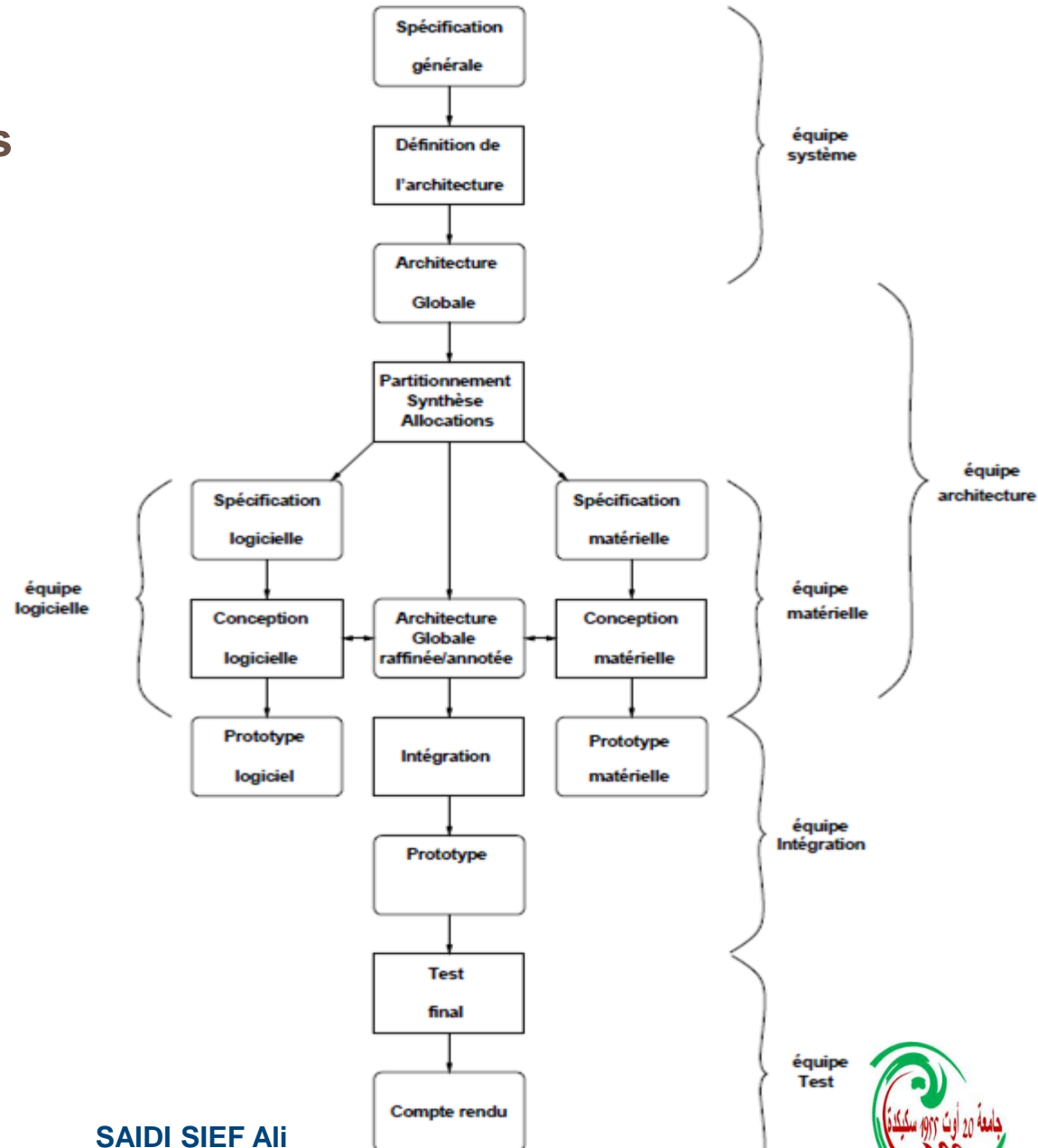
Chapitre II conception des logiciels embarqués

II.1. Conception des systèmes embarqués :

II.1.1. Flots classiques de conception des systèmes embarqués

II.1.1.1. Le flot :

Flot de conception Récent (codéveloppement)



Chapitre II conception des logiciels embarqués

II.1. Conception des systèmes embarqués :

II.1.1.2. Les limitations du flot classique :

Le flot classique de conception de systèmes embarqués possède de nombreuses faiblesses qui le rendent inadapté pour supporter la complexité des systèmes embarqués de troisième génération.

Tout d'abord, il manque une description globale qui accompagnerait la conception du système (logiciel et matériel) du début à la fin. Cette description permettrait à toutes les équipes de conception de bien connaître l'ensemble du système, ce qui éviterait de nombreuses erreurs. Il devrait également être possible d'effectuer des vérifications du système complet à tous les stades de la conception.

La séparation entre le logiciel et le matériel est effectuée trop tôt dans le flot de conception : au stade où elle est effectuée il n'est souvent pas possible de savoir quelle est la meilleure configuration. A l'inverse, l'intégration des différentes parties est effectuée trop tard dans le flot : il est souvent trop tard pour lever les incompatibilités.

Les faiblesses de ce type de flot font que les temps de développement ne sont plus réalistes pour la conception des systèmes embarqués actuels.



Chapitre II conception des logiciels embarqués

II.1. Conception des systèmes embarqués :

II.1.2. Flots de conception récents :

Si le flot de conception classique pouvait suffire pour les première et deuxième générations de systèmes embarqués, la troisième génération, fortement hétérogène et multimaître, est trop complexe pour que les limitations précédemment énoncées soient acceptable telles que le codéveloppement.

II.1.2.1. Codéveloppement :

Le codéveloppement ou codesing a pour but de développer conjointement les diverses parties d'un système hétérogène (logiciel, électronique, mécanique, etc.). C'est pourquoi une description globale du système est nécessaire.

Deux approches sont possibles pour cette description : la première consiste à décrire toutes les parties dans un langage unique. Ce langage doit alors avoir une sémantique pour chacune des parties. L'avantage est qu'il est plus simple pour les outils et les utilisateurs de gérer un langage unique. L'inconvénient est qu'il est difficile, sinon impossible, de définir un langage qui soit vraiment adapté à toutes les parties du système.



Chapitre II conception des logiciels embarqués

II.1. Conception des systèmes embarqués :

II.1.2. Flots de conception récents :

II.1.2.1. Codéveloppement :

La deuxième approche consiste à utiliser plusieurs langages, chacun étant adapté à une partie du système. Un langage de coordination sert à faire le lien entre toutes les descriptions. L'avantage est que, cette fois, il est possible d'avoir le langage optimal pour chaque partie. L'inconvénient est que la gestion de tous ces langages est difficile.

Une partie difficile du codéveloppement est le découpage entre le logiciel et le matériel. De nombreuses heuristiques ont été développées, etc. cependant l'approche manuelle prévaut toujours.

La figure précédente (**flot de conception récent**) donne un exemple de flot basé sur le codéveloppement en comparaison avec le flot classique : dans le nouveau flot, la séparation entre le logiciel et le matériel est effectuée plus tard, et une nouvelle équipe fait son apparition : l'équipe d'architecture, qui s'occupe du système global et de la coordination entre les équipes.



Chapitre II conception des logiciels embarqués

II.1. Conception des systèmes embarqués :

II.1.2. Flots de conception récents :

Le codéveloppement repose aussi sur l'utilisation d'une autre technique permettant la simulation du système aux divers niveaux d'avancement dans la conception. Ces simulations, appelées cosimulations, sont décrites dans la section suivante.

II.1.2.2. Cosimulation :

La cosimulation a pour but de simuler conjointement les diverses parties d'un système hétérogène. Cela permet d'effectuer la validation d'un système complet avant le prototype, mais aussi à divers niveaux d'abstraction.

Il existe deux méthodes pour effectuer cette cosimulation (figure ci-après) :

- la première consiste à traduire les descriptions des diverses parties dans un langage unique pour la simulation. Très souvent, il s'agit d'un langage de programmation tel que le C pour accélérer les simulations. La difficulté est d'être assuré que la traduction et la simulation du langage unique ne changent pas la sémantique des descriptions des diverses parties.



Chapitre II conception des logiciels embarqués

II.1. Conception des systèmes embarqués :

II.1.2. Flots de conception récents :

II.1.2.2. Cosimulation :

2- La deuxième méthode consiste à conserver les descriptions spécifiques des diverses parties et à exécuter en parallèle les divers simulateurs. Un programme, appelé bus de cosimulation, cette tâche peut s'avérer difficile à effectuer lorsque les modèles de simulation sont différents ; de plus les communications entre les simulateurs peuvent s'avérer coûteuses.

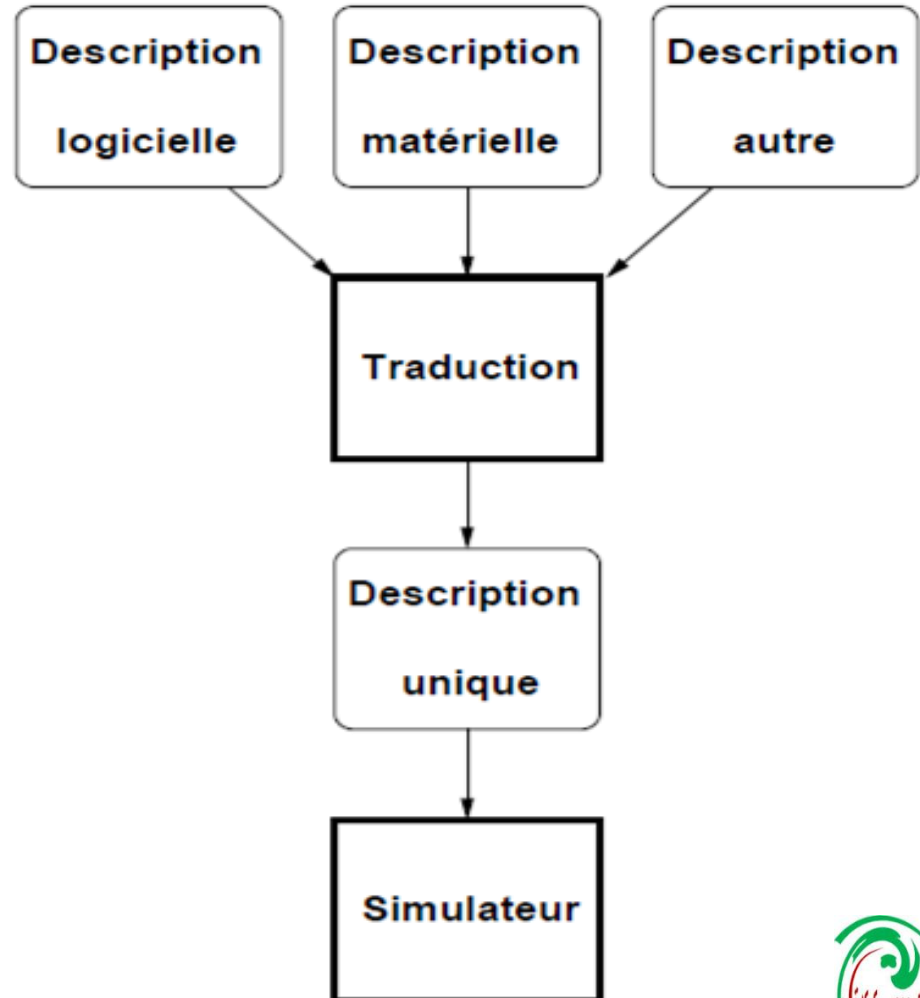


Chapitre II conception des logiciels embarqués

II.1. Conception des systèmes embarqués :

II.1.2. Flots de conception récents :

II.1.2.2. Cosimulation :



Cosimulation avec description unique

Chapitre II conception des logiciels embarqués

II.1. Conception des systèmes embarqués :

II.1.2. Flots de conception récents :

II.1.2.2. Cosimulation :

Cosimulation multilingage

