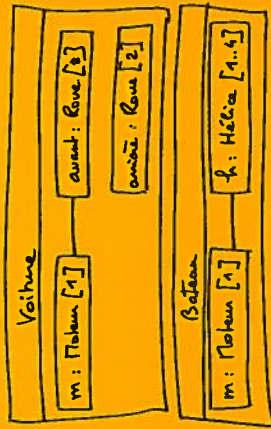
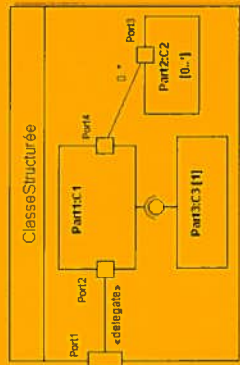


Diagramme de structure composite

Montre l'organisation interne d'un élément structurel complexe



- **Classe structurée** : classe contenant des parties parts, des connecteurs et éventuellement des ports qui délimitent à eux tous sa structure interne.
- **Partie** : fragment structuré d'une classe qui décrit le rôle joué par une instance dans le contexte de la classe structurée. Elle possède un nom, un type et une multiplicité.
- **Connecteur** : relation contextuelle entre des parties dans le contexte d'une classe structurée.
- **Port** : point d'interaction individuel entre une classe et son environnement. Dans une classe structurée, on peut connecter les ports à des parties internes déléguées ou à la spécification de comportement de l'objet dans son ensemble.



Trucs et astuces

- La relation de composition classique fonctionne bien pour modéliser de la décomposition hiérarchique. Cependant, elle présente des limitations significatives lorsqu'il s'agit de relier des éléments au même niveau de décomposition. Préférer alors la classe structurée.
- La différence entre un connecteur et une association classique est la suivante : chaque association est indépendante des autres, alors que tous les connecteurs d'une classe structurée partagent un contexte unique.

Diagramme de composants

Montre des structures complexes, avec leurs interfaces fournies et requises



- **Composant** : partie modulaire d'un système qui encapsule son contenu. Un composant définit son comportement en termes d'interfaces fournies et requises.

Diagramme de composants

Trucs et astuces

- Un composant est une sorte de classe structurée.
- Le diagramme de composants montre les unités logicielles à partir desquelles on construit le système informatique ainsi que leurs dépendances.

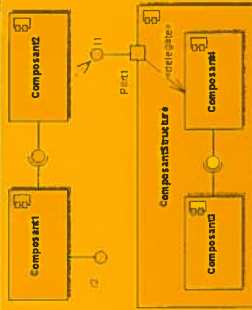
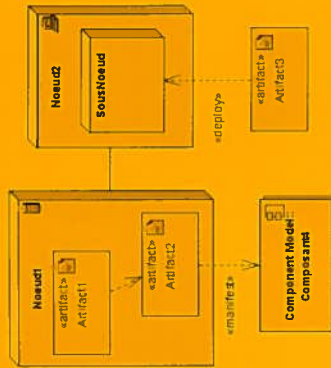


Diagramme de déploiement

Montre le déploiement physique des artefacts sur les ressources matérielles

- **Artefact** : spécification d'un élément physique d'information modèle, fichier ou logiciel pouvant être lié à un composant relation <<manifest>> et déployé sur un nœud.
- **Nœud** : élément physique existant à l'exécution et représentant une ressource de calcul, doté de mémoire et souvent de capacités de traitement. Un nœud peut être soit un simple élément matériel de-vice, soit un environnement d'exécution logiciel.



Trucs et astuces

- Le diagramme de déploiement doit rester simple. Il permet de faire le lien entre le monde physique, matériel, et les abstractions logiques que sont les composants et les classes.
- Cette vue permet de traiter les conséquences de la distribution de l'architecture informatique et de l'allocation de ressources.
- Un artefact peut être déployé sur un nœud en montrant l'imbriication graphique ou en le reliant par une relation « deploy ». Il est utile de montrer la relation « manifest » entre un artefact et un composant.

Du même auteur...

- P. ROQUES, - *UML 2. Modéliser une application web*, 4^e édition, N°12389, 2008, 246 pages (collection Les Cahiers du Programmeur).
- P. ROQUES, F. VALLEE, - *UML 2 en action. De l'analyse des besoins à la conception*, 4^e édition, N°12104, 2007, 382 pages (collection Architecture logiciel).
- P. ROQUES, - *UML 2 par la pratique. Études de cas et exercices corrigés*, 7^e édition, N°12565, 2009, 396 pages (collection Noire).



Conception : Nord Compo

Pascal Roques

EYROLLES