



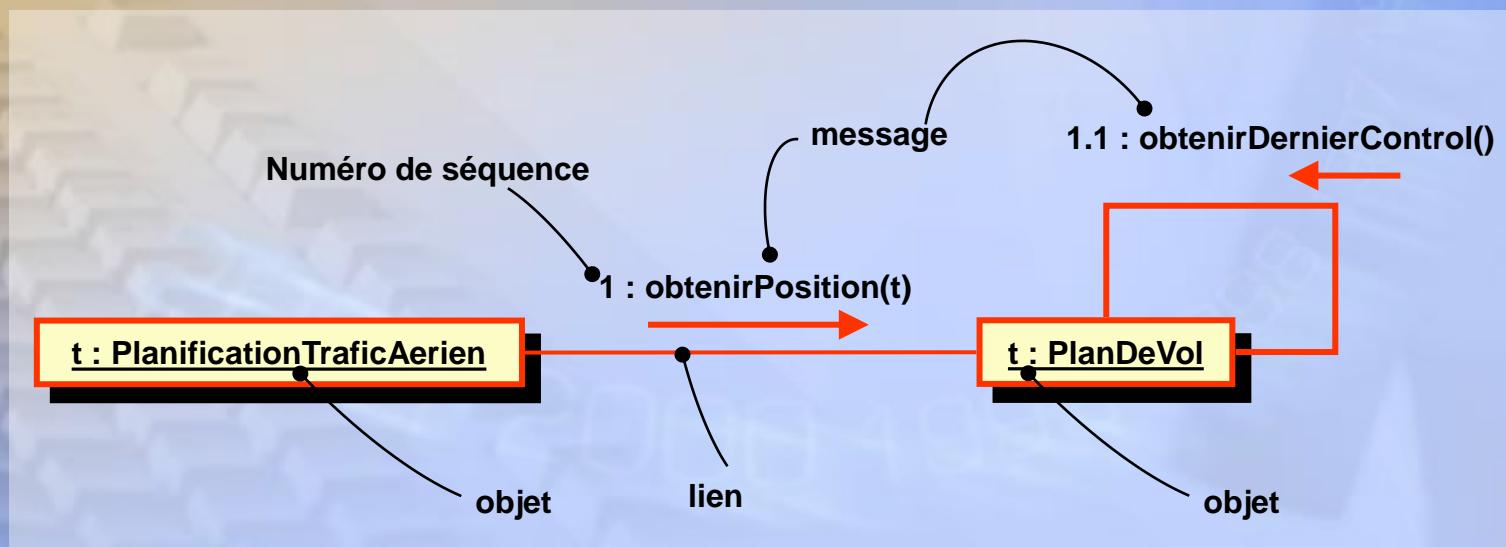
# Modélisation de comportement élémentaires

# Interactions

- Dans tous les systèmes dignes d'intérêt, les objets ne sont jamais inactifs mais interagissent les uns avec les autres en s'envoyant des messages.
- Une interaction est un comportement qui comprend un ensemble de messages échangés au sein d'un groupe d'objets, dans un contexte particulier, pour atteindre un objectif.
  - On utilise les interactions pour modéliser l'aspect dynamique de collaborations qui représentent des sociétés d'objets qui jouent des rôles spécifiques
  - On peut modéliser une interaction de deux façons,
    - en mettant l'accent soit sur l'ordre chronologique des messages,
    - soit sur leur séquencement dans le contexte d'une organisation structurelle des objets.
- Les interactions bien structurées sont comme les algorithmes bien structurés: efficaces, simples, flexibles et compréhensibles.
- On utilise les interactions pour modéliser le flot de contrôle au sein d'une opération, d'une classe, d'un composant, d'un cas d'utilisation ou du système dans son ensemble.

# Interactions (suite)

- Grâce aux diagrammes d'interaction, il est possible d'envisager ces flots de deux manières.
  - On peut soit mettre l'accent sur l'ordre selon lequel les messages sont envoyés dans le temps,
  - soit sur les relations structurelles entre les objets au sein d'une interaction, puis considérer la manière dont ces messages sont gérés dans le contexte de cette structure.



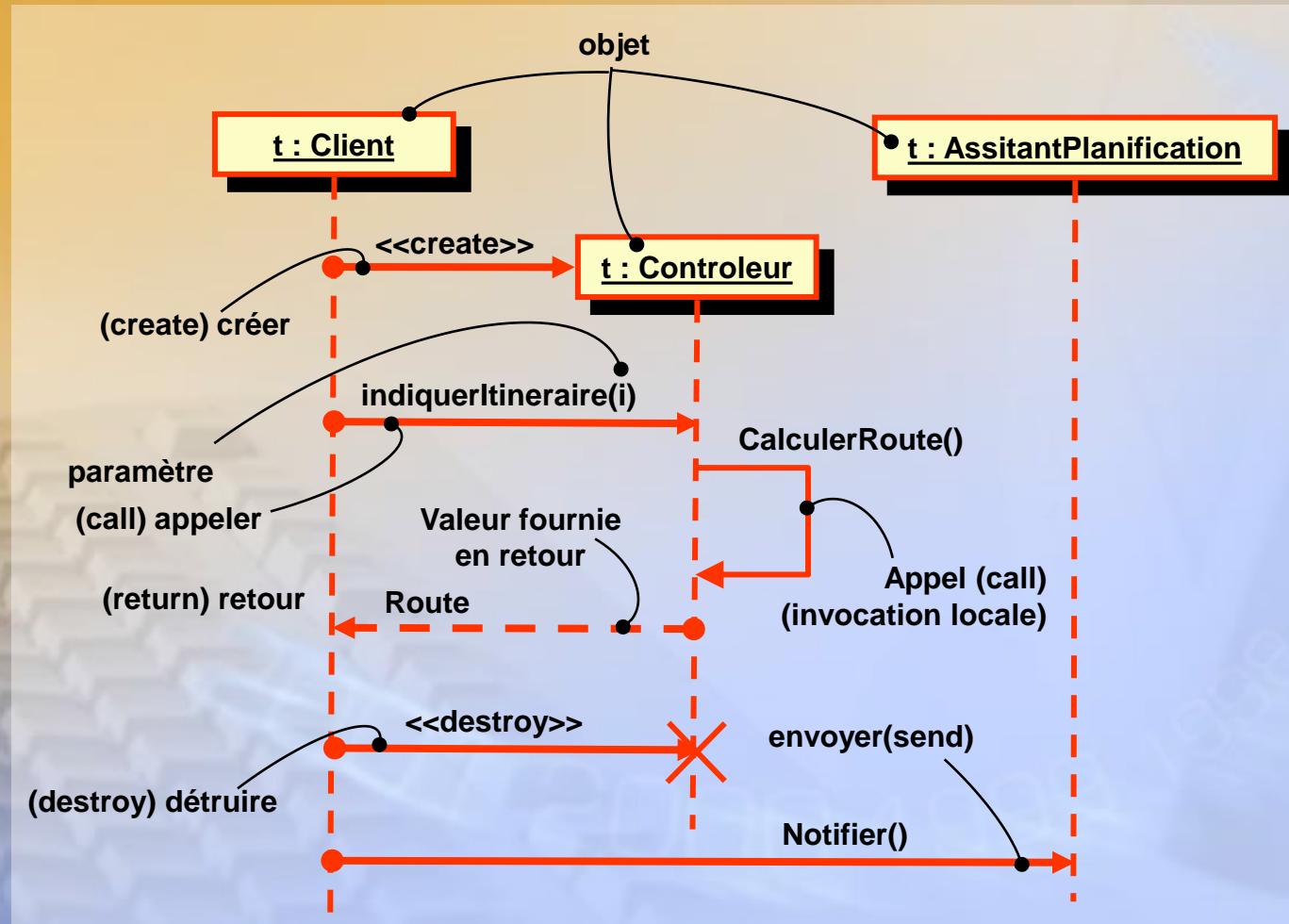
# Liens

- Un lien décrit un chemin suivant lequel un objet peut expédier un message à un autre objet (ou à lui-même).
- La plupart du temps, il suffit d'indiquer qu'un tel chemin existe. S'il est nécessaire d'être plus précis sur la façon dont il existe, il est possible de décorer l'extrémité concernée du lien au moyen d'un des stéréotypes standard suivants:
  - **association** indique que l'objet correspondant est visible par association.
  - **self** indique que l'objet correspondant est visible parce que c'est lui qui invoque l'opération.
  - **global** indique que l'objet correspondant est visible parce qu'il est de portée globale.
  - **local** indique que l'objet correspondant est visible parce qu'il est de portée locale.
  - **paranmeter** indique que l'objet correspondant est visible parce qu'il est passé en paramètre.

# Messages

- Si l'on veut modéliser le changement d'état d'une société d'objets pendant une certaine période, on l'imagine comme le film d'un ensemble d'objets, dont les images représentent les instants qui se succèdent.
- Un message est la spécification d'une communication entre objets, qui transporte des informations et qui s'affiche dans le but de déclencher une activité. La réception d'une instance de message peut être considérée comme une instance d'un événement.
- Lorsqu'on transmet un message, l'action qui en résulte est un énoncé exécutable qui forme une abstraction d'une procédure de calcul. Cette action peut conduire à un changement d'état.
- En UML, on peut modéliser plusieurs genres d'actions:
  - call invoque une opération pour un objet. Un objet peut s'envoyer un message, qui résulte en l'invocation locale d'une opération.
  - return renvoie une valeur à l'émetteur.
  - send envoie un signal à un objet.
  - create crée un objet.
  - destroy détruit un objet.
- Un objet peut se détruire lui-même.

# Exemple de représentation



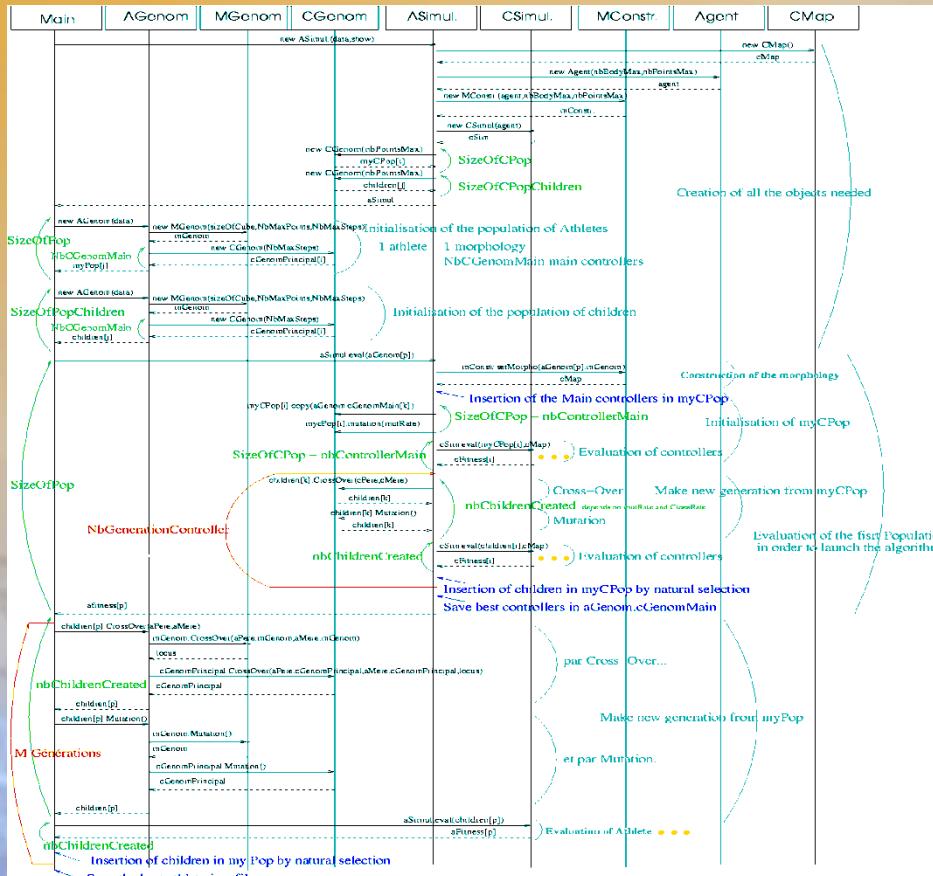
# Les diagrammes d'interaction

- Les diagrammes de séquence et les diagrammes de collaboration - qui sont tous des diagrammes d'interaction sont deux des cinq diagrammes utilisés en UML pour modéliser les aspects dynamiques des systèmes.
- Un diagramme d'interaction représente une interaction, c'est-à-dire un ensemble d'objets et leurs relations, ainsi que les messages qui peuvent circuler entre eux.
  - Un diagramme de séquence est un diagramme d'interaction qui insiste sur le classement des messages par ordre chronologique,
  - alors qu'un diagramme de collaboration est un diagramme d'interaction qui met l'accent sur l'organisation structurelle des objets qui envoient et reçoivent des messages.
- Les diagrammes d'interaction servent à modéliser les aspects dynamiques d'un système.
  - Pour l'essentiel, cela implique la modélisation des instances concrètes ou prototypes de classes, d'interfaces, de composants, et de nœuds, ainsi que celle des messages qui sont échangés entre eux, le tout dans le contexte d'un scénario qui illustre un comportement.
  - Les diagrammes d'interaction peuvent exister seuls pour visualiser, spécifier, construire et documenter la dynamique d'une société d'objets particulière ou être utilisés pour modéliser un flot de contrôle spécifique d'un cas d'utilisation.

# Diagramme de séquence

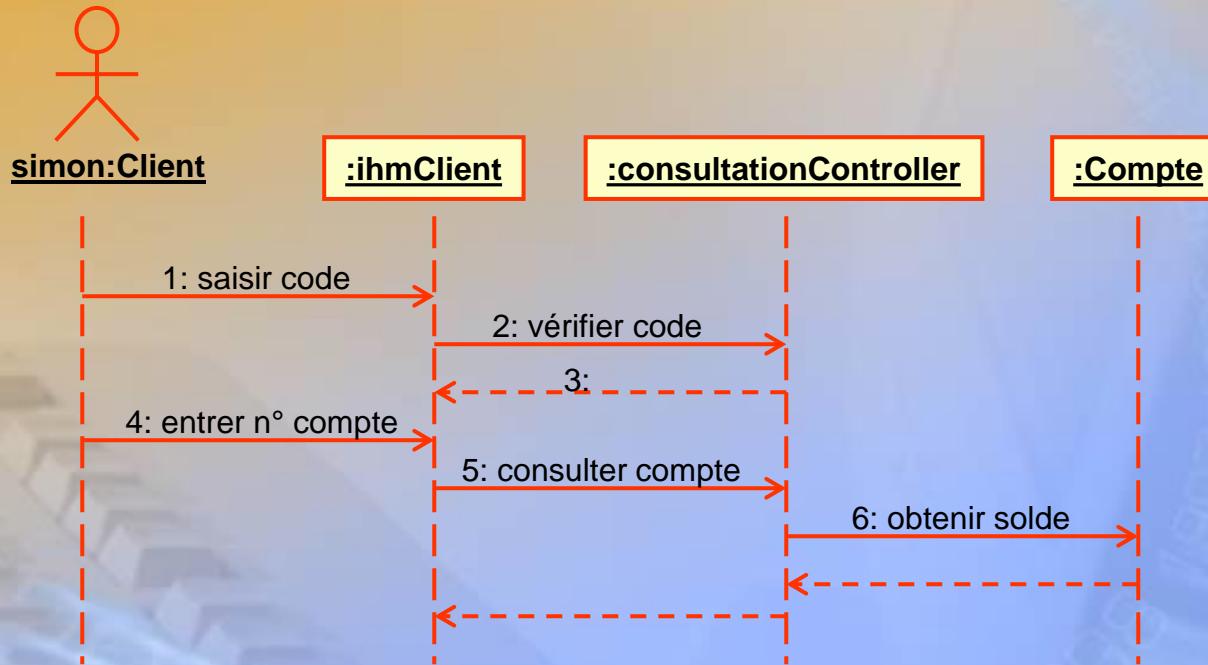
- **Représentation des interactions entre acteurs et objets**
- **Vision temporelle d'une interaction**
  - Chaque objet est symbolisé par une barre verticale
  - Le temps s'écoule de haut en bas, de sorte que la numérotation des messages est optionnelle.
  - Diagramme dual du diagramme de collaboration
- **Souvent utilisé pour représenter une instance de cas d'utilisation**
- **De manière plus générale, représentation temporelle d'une interaction**
  - Bien adapté pour de longues séquences
  - Ne visualise pas les liens
  - Complémentaire du diagramme de collaboration

# A éviter absolument...



A un ensemble de diagrammes de séquences doit correspondre un algorithme (simple)...

# C'est mieux



# Usage du diagramme de séquence

- En tant que document d'analyse : sert à représenter de plus en plus précisément la dynamique du système
- En tant que document de conception : sert à figer les interactions entre les sous-parties du logiciel
- En tant que document d'implémentation : sert à décrire les algorithmes (la partie interaction)
- Peut conduire à des re-découpages des vues statiques

**Essentiel pour le passage de l'analyse à la conception**

# Diagramme de collaboration

- **Représentation d'une collaboration entre rôles**
  - Booch : Société d'objets collaborant (UML 1.5 : de rôles communicants)
- **Représentation spatiale d'une interaction**
  - Mise en avant de la structure
  - Représentation des structures complexes (récursives par exemple)
- **Pas d'axe temporel**
  - Diagramme dual du diagramme de séquence
- **Des rôles ou des objets dans une situation donnée**
- **Des liens relient les objets qui se connaissent**
- **Les messages échangés par les objets sont représentés le long de ces liens**
- **L'ordre d'envoi des messages est matérialisé par un numéro de séquence**

# Séquence et collaboration comparé

Diagramme de séquence

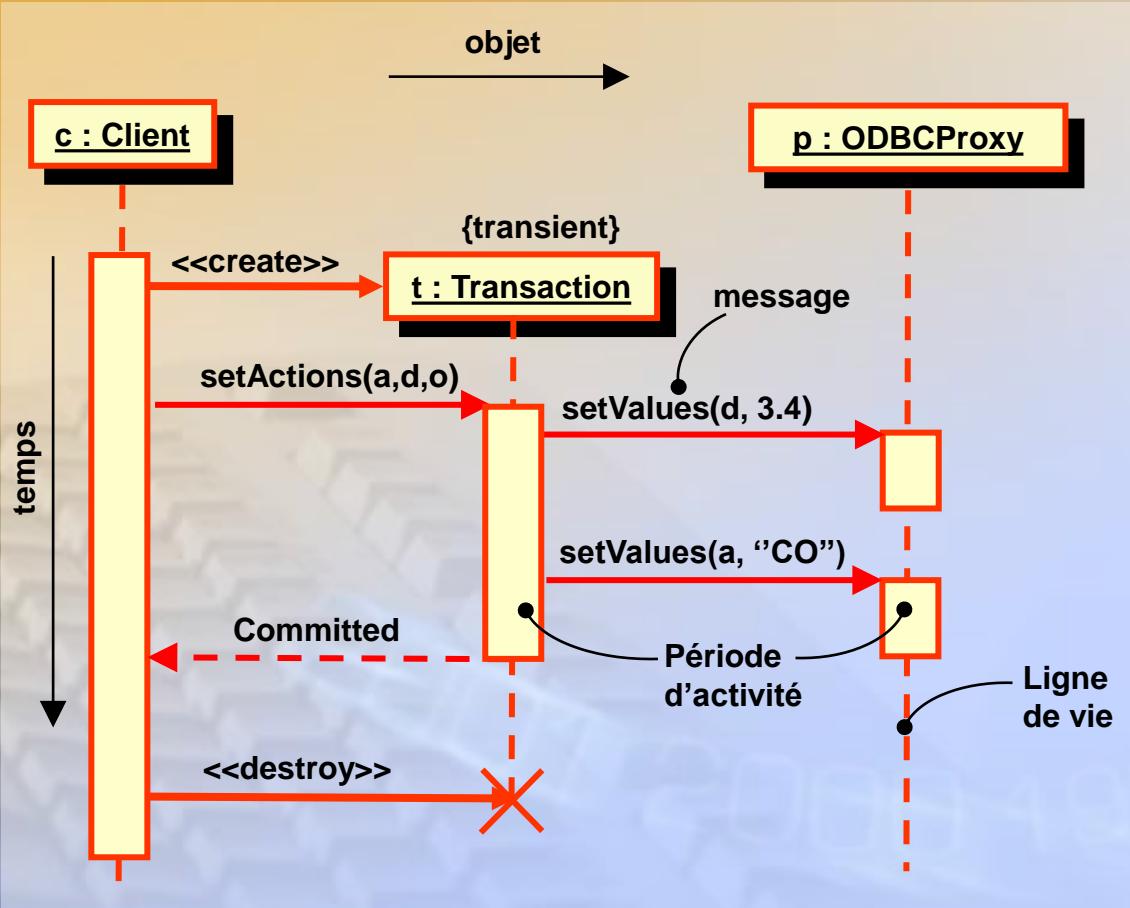
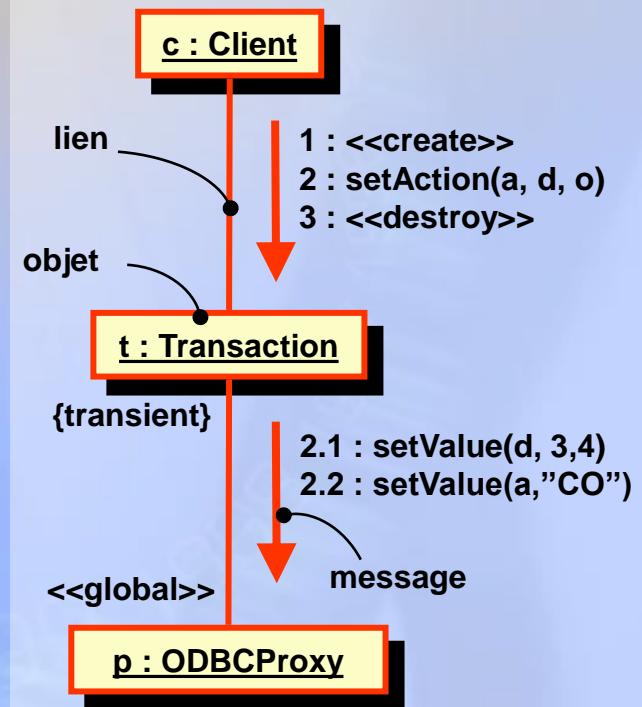
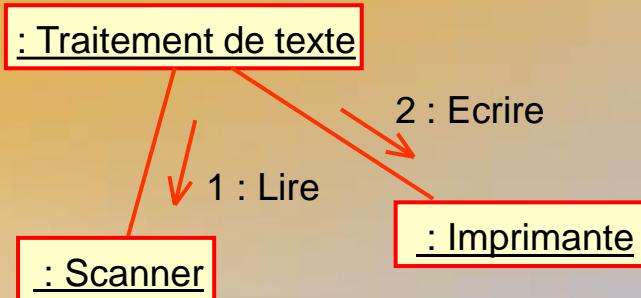


Diagramme de collaboration

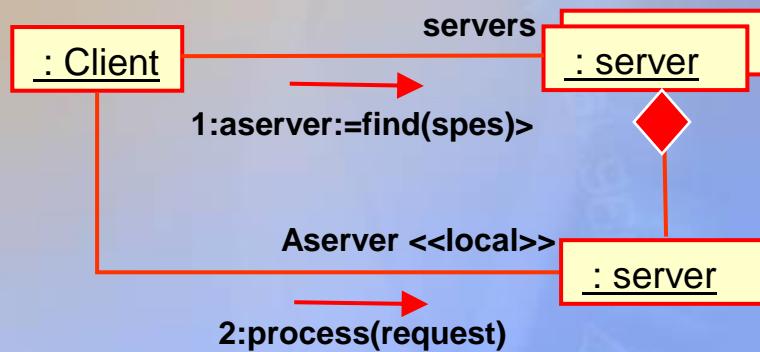


# Exemples

## Object Actif



## Multi-objets



## Avec Acteur



# A vous de jouer



- Exercice 1 : Retrait en espèce**
- Exercice 2 : Imprimante**
- Exercice 3 : Simple Watch**
- Exercice 4 : Ascenseur**

# Corrections



# Diagrammes d'activités

# Modélisation de comportement avancés

# Evénement et signaux

# Automates à états finis

# Temps et espace

# Diagrammes d'états-transitions

# Plus d'information ...

- <http://www.rational.com/products/rup/>
- <http://www.therationaledge.com>
- <http://www.ambisoft.com/>
- <http://www.ronin-intl.com/publications/unifiedProcess.html>

# Dialogue

