

Etude de cas UML

SOMMAIRE

<i>I. Introduction</i>	<i>3</i>
<i>II. Spécifications</i>	<i>3</i>
1. Description des cas d'utilisation	3
2. Description des scénarios	4
3. Structure de l'IHM	7
<i>III. Analyse</i>	<i>11</i>
1. Dictionnaire des données	11
2. Diagramme des classes	13
3. Diagramme d'état transition	14
4. Diagrammes de séquence	15
5. Diagrammes de collaboration	17
<i>IV. Conclusion</i>	<i>19</i>

I. Introduction

UML est un langage de modélisation fondé sur les concepts objet : l'objet d'UML est de fournir une notation standard utilisable dans le développement de systèmes informatiques basés sur l'objet.

Cependant, UML n'est pas une méthode car il n'inclut pas la manière d'utiliser les concepts qu'il se propose c'est à dire l'enchaînement des étapes qui mènent à la résolution des problèmes posés.

En conséquence, nous allons utiliser la démarche proposée par les enseignements de l'UT1 pour traiter l'étude de cas ASSURANCETOURISK.

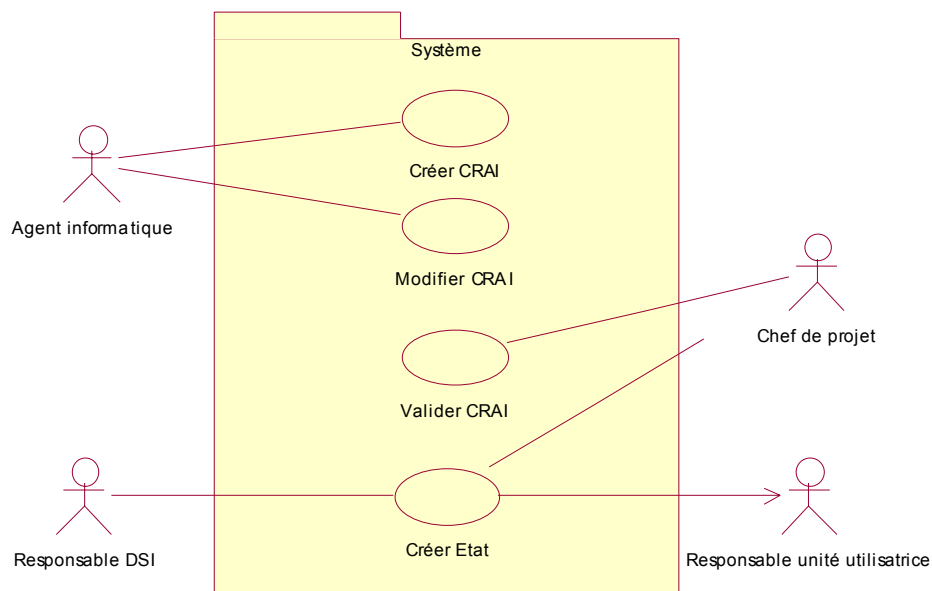
II. Spécifications

L'énoncé d'un besoin exprime un comportement ou une propriété que le futur système doit respecter, la formulation doit se faire en termes compréhensibles. Dans notre cas, les besoins sont exprimés dans l'énoncé du problème :

- Envisager une automatisation complète du processus de gestion du CRAI.
- Fournir un état récapitulatif des activités mensuelles.

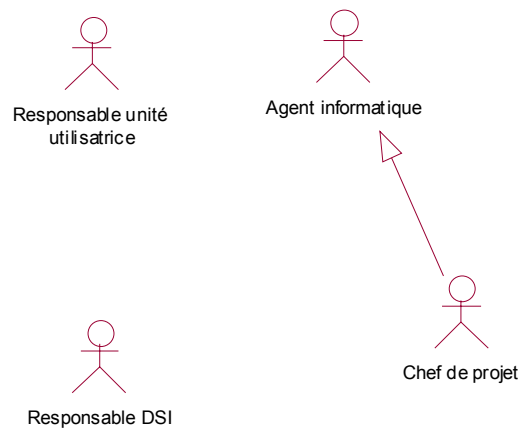
1. Description des cas d'utilisation

Les cas d'utilisations sont des outils formels qui permettent de consigner et d'exprimer des interactions entre les utilisateurs et le système. On peut noter que les cas d'utilisations sont utilisés durant tout le processus car ils servent à la création de l'IHM, à la spécification des tests (recette)...



Ces cas d'utilisation seront détaillés ci-après dans les scénarios.

La hiérarchie des acteurs est la suivante :



2. Description des scénarios

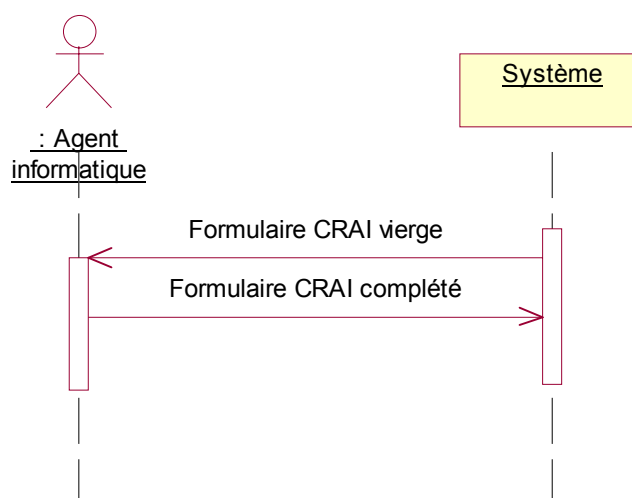
Un cas d'utilisation est une abstraction de un ou plusieurs scénarios, une instance de cas d'utilisation est appelée un scénario.

Les scénarios pourraient être décrits sous forme textuelle mais nous vous proposons le formalisme vu en cours.

On utilisera des diagrammes de séquences particuliers pour décrire formellement ces scénarios.

NB : Les scénarios ne traitent pas l'identification des acteurs, on considère donc que lors des interactions avec le système, les acteurs sont identifiés au préalable.

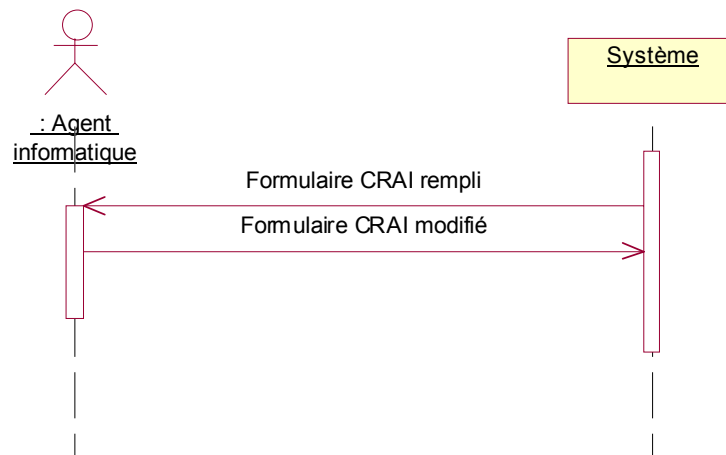
- *Scénario Créer CRAI*



Objectif : un agent, y compris un chef de projet, souhaite saisir un CRAI. Un CRAI est en fait une liste d'interventions.

Le système affiche un formulaire avec les zones de saisies nécessaires pour un CRAI. L'agent peut saisir ainsi toutes les informations.

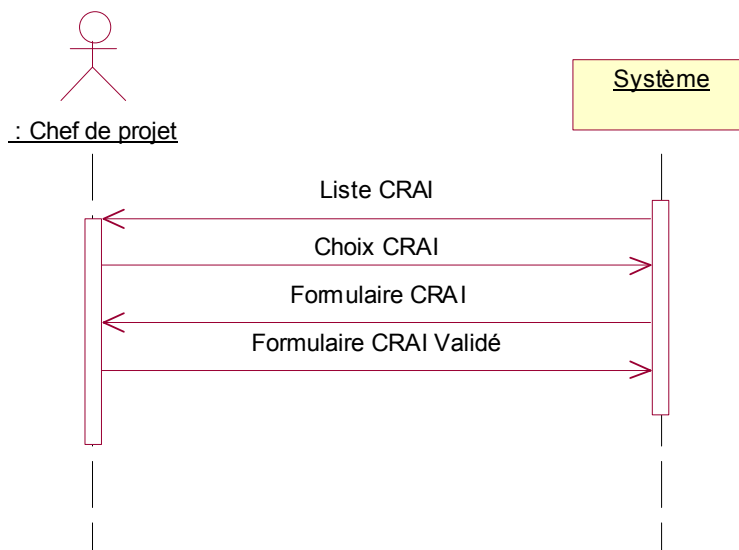
- *Scénario Modifier CRAI*



Objectif : L'agent informatique modifie un CRAI qu'il a saisi auparavant et le renvoie.

Le système renvoie le CRAI de la semaine en cours pour permettre la modification de celui-ci. Si le CRAI a déjà été validé, on lui présente le CRAI de la semaine suivante.

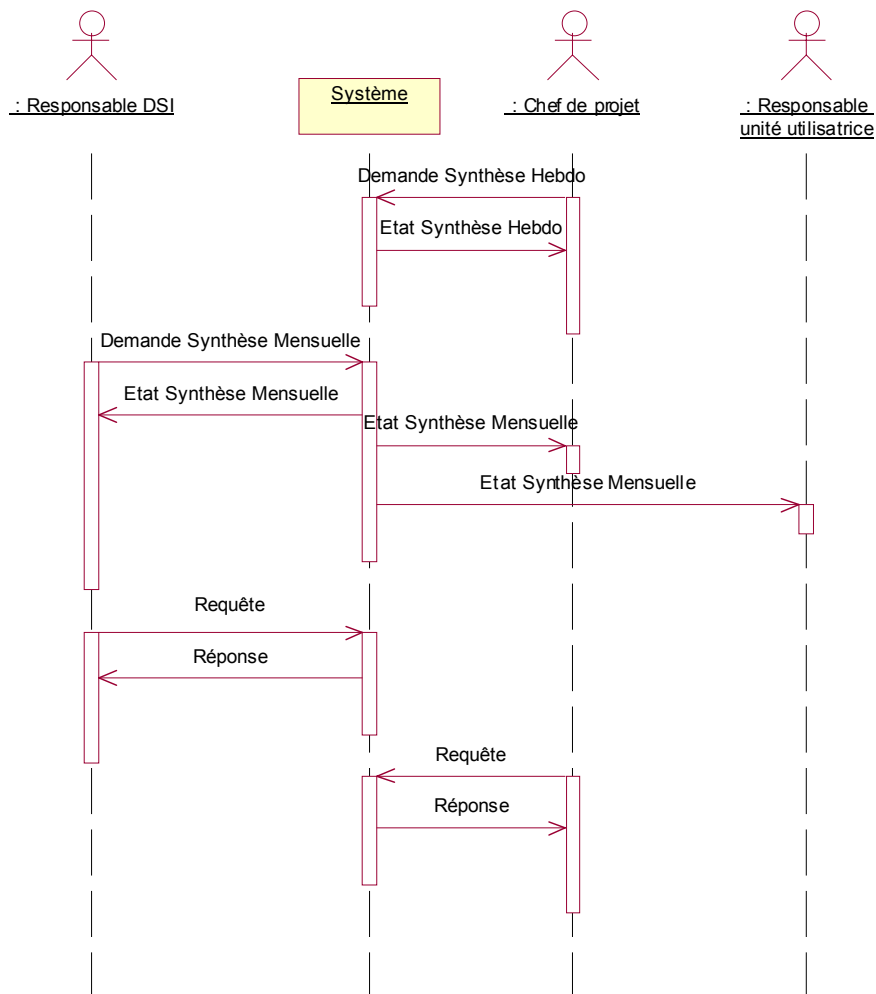
- *Scénario Valider CRAI*



Objectif : un chef de projet doit valider les CRAI à chaque fin de semaine.

Le chef de projet doit vérifier le CRAI de son choix (en fait toutes les interventions qui constituent le CRAI), celui-ci lui sera renvoyé par le système, s'il est correct, il le valide, sinon il faut le corriger avant validation.

- *Scénario Créer Etat*



Objectif: le responsable DSI ou le chef de projet ont la possibilité de créer des états hebdomadaires ou mensuels.

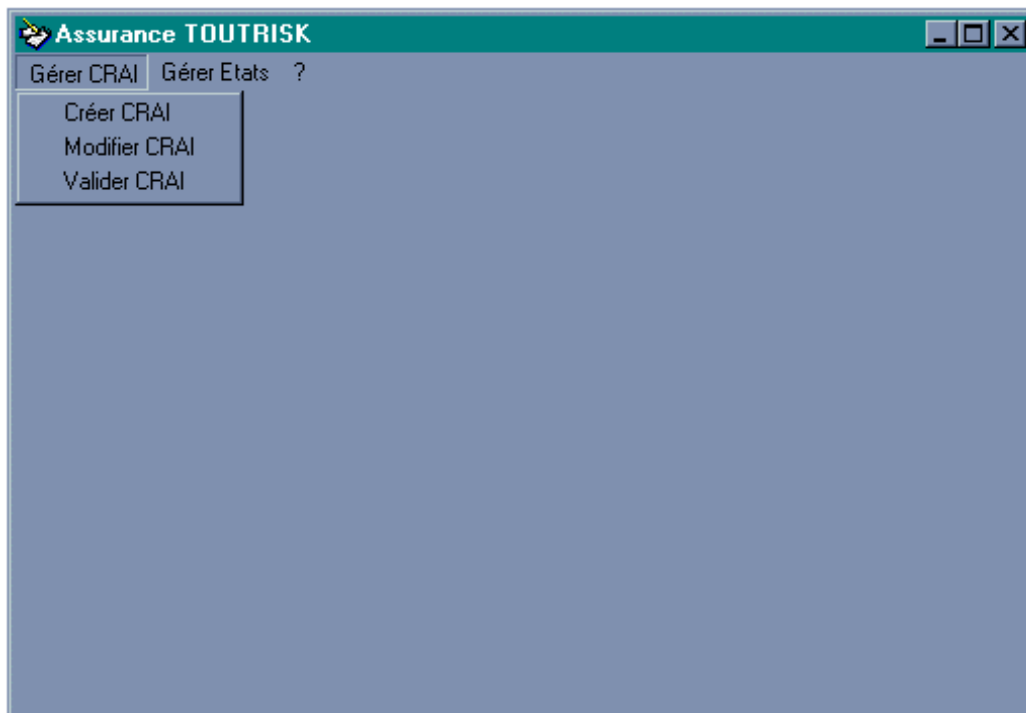
Trois scénarios sont à distinguer :

- Création d'un état hebdomadaire par le chef de projet pour gérer ses projets.
- Création d'un état mensuel par le DSI dont une copie est transmise au chef de projet et au responsable de l'unité utilisatrice.
- Création d'une requête libre dans le but de donner plus de souplesse au système proposé.

3. Structure de l'IHM

A partir du diagramme des cas d'utilisation, nous avons défini l'IHM suivante, celle ci est donnée à titre d'exemple, c'est lors de la phase de conception que le choix de l'architecture logicielle cible sera effectivement réalisé.

On peut remarquer que chaque cas d'utilisation identifié est représenté dans la structure des menus déroulants.



- *Ecran pour valider un CRAI*

Compte Rendu d'Activités

Matricule : Année :

Nom agent : Semaine :

Code activité	Code projet	Code unité	Description de l'interv.	Temps hebdo.	Validation
▼	▼	▼			<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
<i>Cumul</i>					<input type="checkbox"/>

Mode :

- *Ecran pour le choix d'un CRAI par un chef de projet*

Choix d'un Compte Rendu d'Activités

Code projet :

Code agent :

Cet écran est affiché avant le précédent pour que le chef de projet puisse choisir le CRAI correspondant à valider pour un agent particulier.

III. Analyse

L'analyse consiste à partir des cas d'utilisation et des besoins recueillis à élaborer la structure du système à un niveau d'abstraction qui va au-delà de l'implémentation physique.

L'essentiel est de s'assurer que tous les besoins fonctionnels sont réalisés quelque part dans le système.

1. Dictionnaire des données

Dictionnaire des données				
Sigle	Libellé	Type	Règle	
CRAI	Formulaire CRAI	Struct.	Matricule, Nom, Prénom, Année, Semaine, Code activité, Code projet, Description de l'intervention, Unité utilisatrice concernée, Temps hebdomadaire (en heures), Validation, CumulAgent, Mode	
Matricule	Matricule d'un employé	Entier		
Nom	Nom d'un agent	Chaîne		
Prénom	Prénom d'un agent	Chaîne		
Année	Année de création du CRAI	Date		
Semaine	Semaine de création du CRAI	Date		
CodeActivité	Code de l'activité réalisée	Entier		
CodeProjet	Code du projet concerné	Chaîne		
Description	Description de l'intervention	Chaîne		
CodeService	Code du service	Chaîne		
Temps	Temps hebdomadaire (en heures)	Entier		
Validation	Validation de l'intervention	Booléen		
CumulAgent	Cumul des heures hebdomadaires d'un agent	Heure		
				Un chef de projet est un agent
				Un chef de projet est un agent
			Texte libre	
			Calculée par $\sum(\text{TempsPassé})$	
ChoixCRAI	Choix d'un CRAI pour validation	Struct.	Code projet, Code agent	
SynthèseHebdo	Synthèse des Activités Hebdomadaires	Struct.	Code projet, Code agent, Année, Semaine, Code activité, Description de l'intervention, Unité utilisatrice concernée, Temps hebdomadaire (en heures), Cumul	
LibelléService	Libellé du service	Chaîne		
CumulProjet	Cumul des heures hebdomadaires d'un projet	Heure		
			Calculée par $\sum(\text{Temps hebdomadaire})$	
SynthèseMensuelle	Synthèse des Activités Mensuelles	Struct.	Année, Semaine1, Semaine2, Code projet, Libellé du projet, Unité utilisatrice concernée, Temps mensuel projet (en heures), Cumul projets, Temps mensuel unité (en heures), Cumul unités,	
Mois	Mois	Date		
LibelléProjet	Libellé du projet	Chaîne		

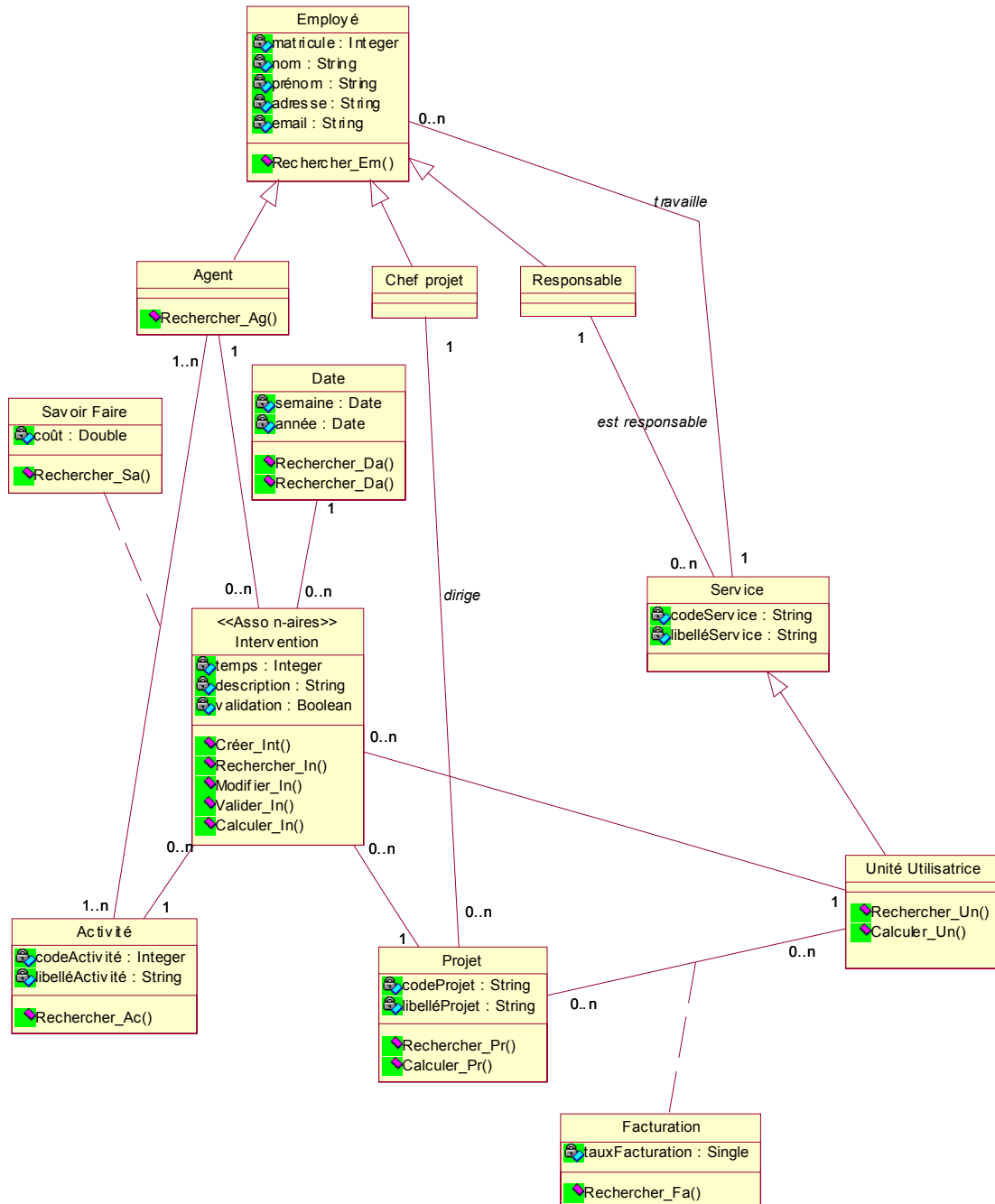
<i>TempsMensuelProjet</i>	<i>Temps mensuel projet (en heures)</i>	<i>Heure</i>	<i>Calculée par \sum(Temps hebdomadaire consacrée à une intervention incluse au projet)</i>
<i>TempsMensuelUnité</i>	<i>Temps mensuel unité (en heures)</i>	<i>Heure</i>	<i>Calculée par \sum(Temps hebdomadaire consacrée à une intervention liée à l'unité)</i>
<i>CumulProjets</i>	<i>Cumul projets</i>	<i>Heure</i>	<i>Calculée par \sum(TempsMensuelProjet)</i>
<i>CumulUnités</i>	<i>Cumul unités</i>	<i>Heure</i>	<i>Calculée par \sum(TempsMensuelUnité)</i>
Adresse	Adresse de l'employé	Chaine	
Email	E-mail de l'employé	Chaine	
Cout	Coût de l'intervention	Réel	
TauxFacturation	Taux de facturation	Réel	
LibelléActivité	Libellé de l'activité	Chaine	

Légende:

Les données calculées sont présentées en *italique*

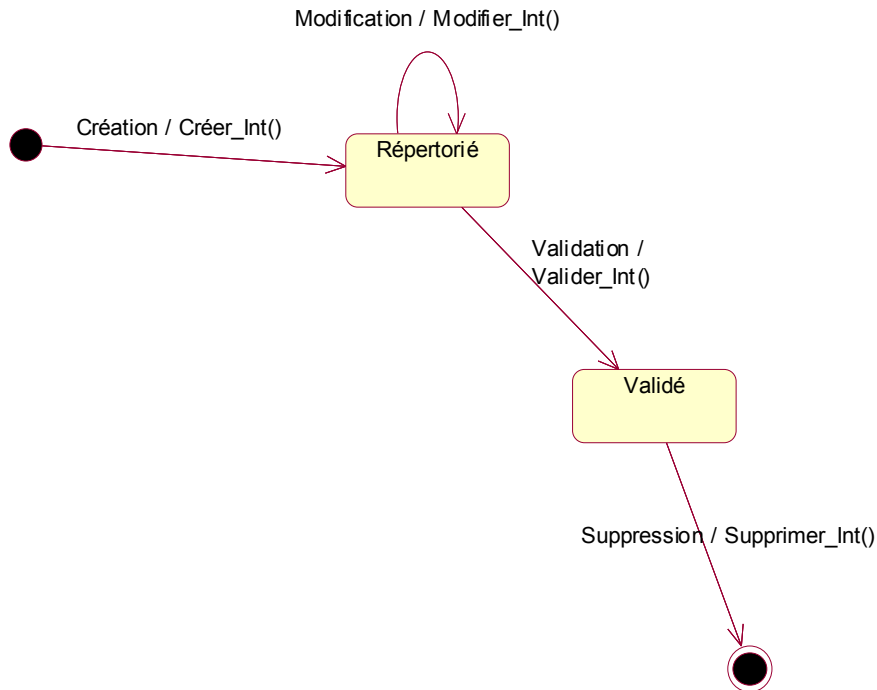
2. Diagramme des classes

Ce diagramme donne une représentation statique du système. Il se compose de classes et de leurs relations. Une classe regroupe des données et des méthodes.



3. Diagramme d'état transition

Voici le DET de la classe Intervention. Les autres DET n'ont pas été présentés car ils n'étaient pas pertinents.

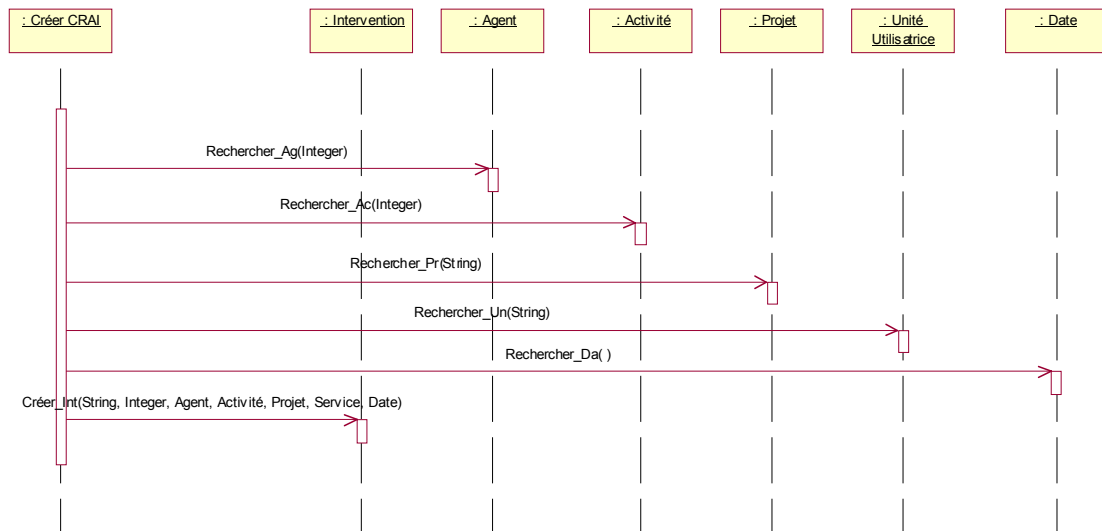


4. Diagrammes de séquence

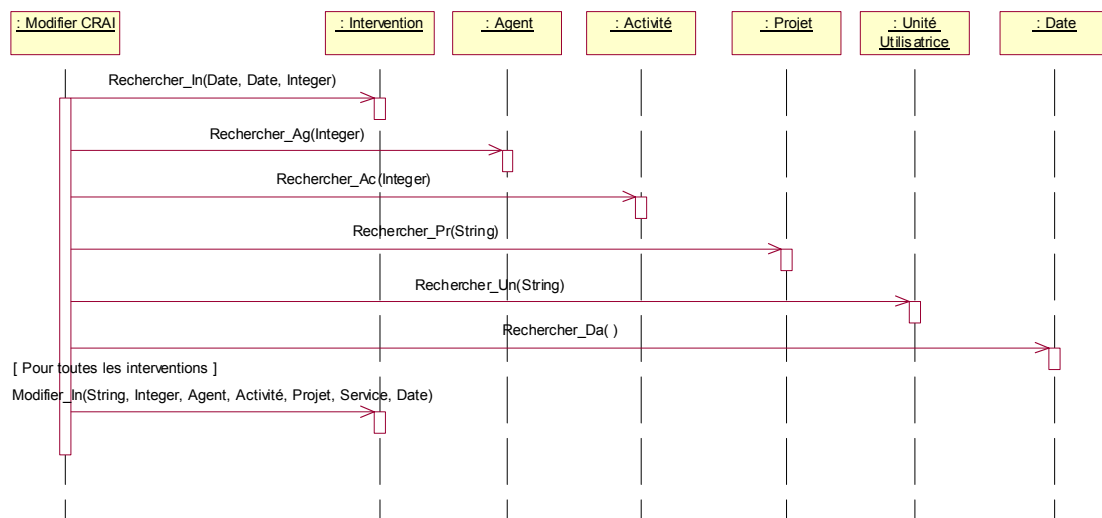
Dans les diagrammes suivants, nous avons choisi d'utiliser l'objet interface comme chef d'orchestre pour gérer les différents échanges entre objets.

- *Créer CRAI*

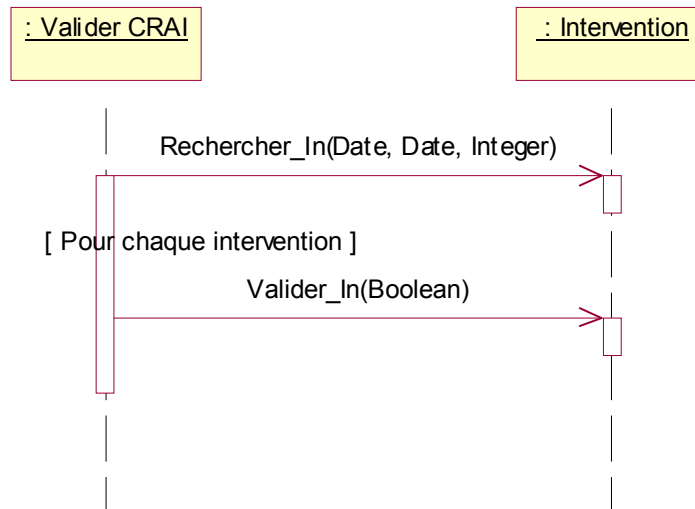
Ce diagramme présente les interactions entre les différents objets et l'objet interface de Créer CRAI.



- *Modifier CRAI*

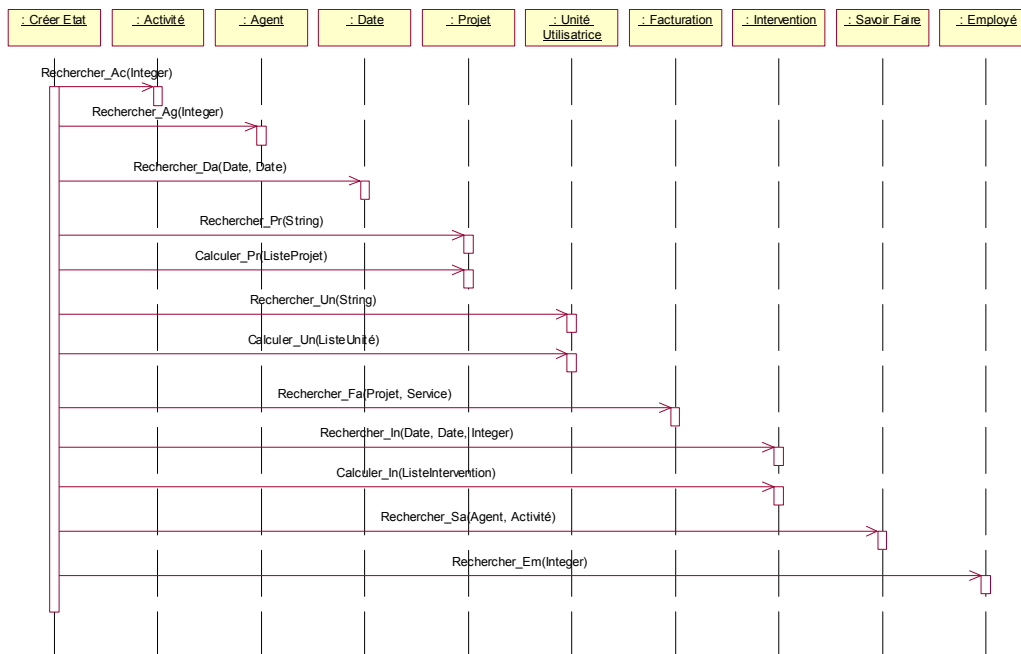


- **Valider CRAI**



- **Créer Etat**

L'objet interface envoie plusieurs messages sur un même objet selon la requête qui sera considérée puisque cet état décrit une interaction entre objets pour la création des états mensuels et hebdomadaires.

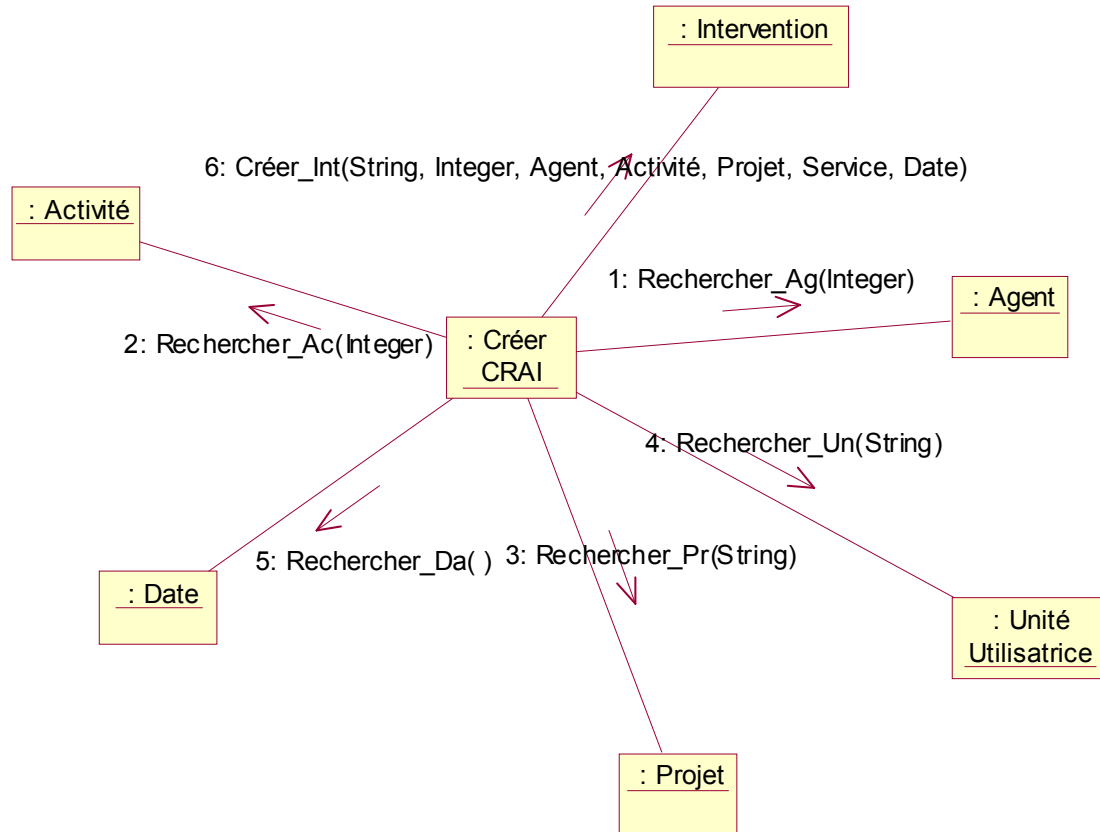


En ce qui concerne la création d'une requête libre, le diagramme n'a pas été représenté : les objets nécessaires seront fonction de la requête.

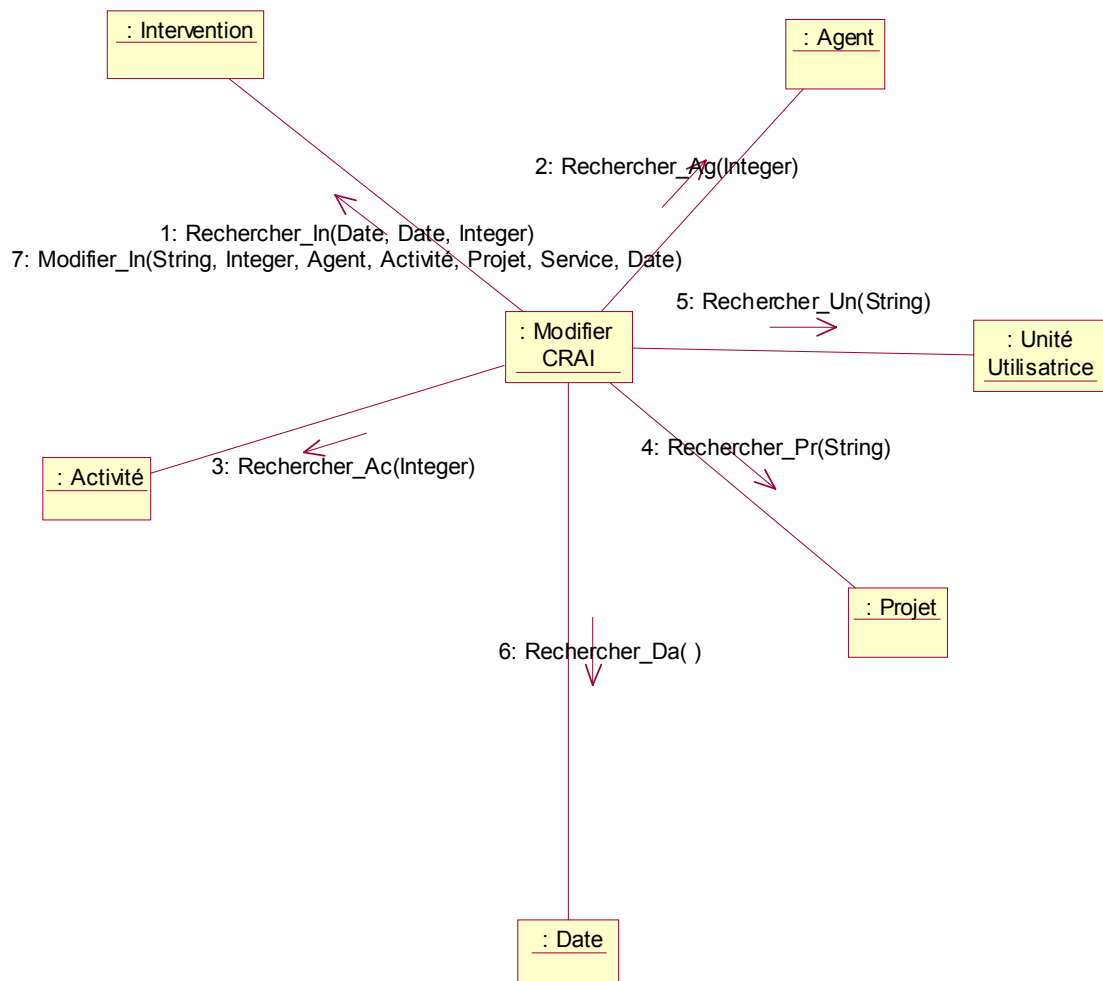
5. Diagrammes de collaboration

Ces diagrammes ont été générés automatiquement avec ROSE à partir des diagrammes de séquence.

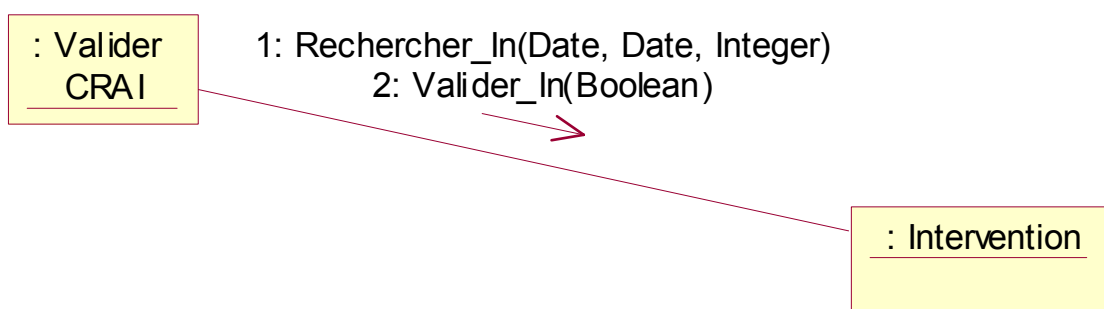
- **Créer CRAI**



• **Modifier CRAI**



• **Valider CRAI**



IV. Conclusion

La démarche que nous avons utilisée nous a permis de fournir les spécifications fonctionnelles, statiques et dynamiques du système étudié. Une étude plus complète aurait consisté à poursuivre le travail présenté en phase de conception dans laquelle les abstractions du métier mises en évidence par notre analyse auraient été confrontées à la réalité logicielle.

Enfin, la démarche proposée est orientée données. Toutefois, nous aurions pu adopter une méthode basée sur l'approche par composants, par exemple la démarche E-process de la société B&T Associés.