**KAOS** (Keep All Objectives Satisfied) : approche de modélisation des exigences centrée utilisateur

KAOS est une méthode appartenant aux GOREs c'est à dire une méthode basée sur les buts.

L'ontologie de KAOS comprend **des objets**, qui sont des objets d'intérêt dans le système composite dont les instances peuvent évoluer d'un état à l'autre. Les objets peuvent être des **entités**, **des relations** ou des **événements**.

**Les opérations** sont des relations entrées-sorties sur des objets. Les applications d'exploitation définissent des transitions d'état. Les opérations sont déclarées par des signatures sur des objets et ont des conditions pré, post et déclenchement. KAOS fait une distinction entre les conditions pré- / post-condition de domaine pour une opération et les pré- / post-conditions souhaitées pour elle. Les premières sont indicatives et décrivent ce qu'une application de l'opération signifie dans le domaine (sans aucune prescription quant à quand l'opération doit ou ne doit pas être appliquée) alors que ces derniers sont optatifs et capturent un renforcement supplémentaire des conditions permettant d'atteindre les objectifs.

Un **agent** est un type d'objet qui agit comme un processeur pour les opérations. Les agents sont des composants actifs qui peuvent être des humains, des dispositifs, des logiciels, etc. Les agents effectuent des opérations qui leur est attribué. KAOS permet aux analystes de préciser quels objets sont observables ou contrôlables par les agents.

Un **but** dans KAOS est défini comme une "déclaration d'intention prescriptive sur un système dont la satisfaction dans les genres exige la coopération de certains des agents formant ce système". Les objectifs de KAOS peuvent faire référence aux services (objectifs fonctionnels) ou à la qualité des services (objectifs non fonctionnels). Dans KAOS, les objectifs sont organisés dans l'habituelle AND / OR raffinement-abstraction Hiérarchies. Le raffinement d'objectif se termine lorsque chaque sous-objectif est réalisable par un agent individuel qui lui est affecté. Cela signifie que l'objectif doit être exprimable en termes de conditions qui sont surveillables et contrôlables par l'agent. L'exigence et l'attente dans KAOS sont définis de la manière habituelle - le premier étant un but sous la responsabilité d'un agent dans le système futur et le dernier étant un objectif sous la responsabilité d'un agent dans l'environnement. Les modèles de définition des objectifs sont utilisés pour la légère modélisation. Ceux-ci sont spécifiés dans la logique temporelle et incluent des modèles tels que réaliser, cesser, maintenir, optimiser et éviter.

Dans l'ensemble, une spécification KAOS est une collection des modèles de base suivants :

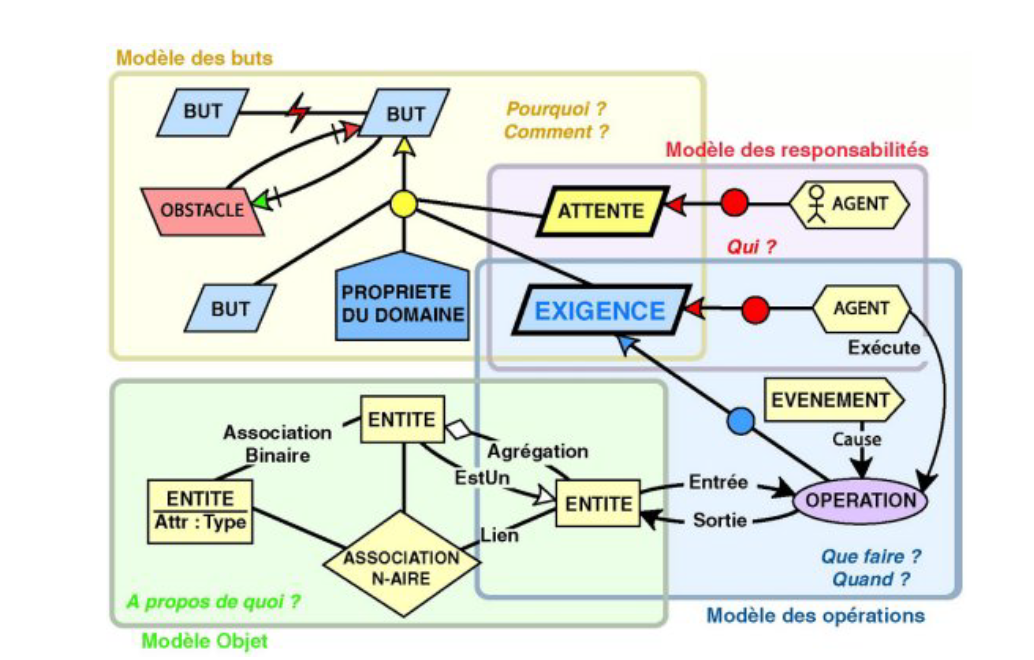
• **un modèle de but** où les objectifs sont représentés et attribués aux agents,

• **modèle d'objet**, qui est un modèle UML qui peut être dérivé des spécifications formelles des objectifs puisqu'ils se réfèrent à des objets ou à leurs propriétés,

• **le** **modèle d'exploitation**, qui définit les différents services à fournir par les agents logiciels.

En conclusion, KAOS est une méthodologie pour l'analyse des besoins axées sur les objectifs qui est fournie avec un cadre formel solide. Lors du raffinement de l'objectif, de l'opérationnalisation des objectifs, de l'analyse des obstacles et de l'atténuation, KAOS s'appuie fortement sur des modèles de raffinement formels qui sont prouvés une fois pour toutes

Schéma descriptif des concepts de KAOS



Vocabulaire KAOS

Agent

Objet actif \* (= processeur) effectuant des opérations \* pour atteindre des objectifs \*. Les agents peuvent être le logiciel considéré comme un ensemble ou des parties de celui-ci. Les agents peuvent également provenir de l'environnement \* du logiciel étudié ; Les agents humains sont dans l'environnement \*.

Association

Objet\*, dont la définition repose sur d'autres objets liés par l'association.

Système composite

Le logiciel étudié et son environnement \*

Conflit

Les objectifs \* sont contradictoires si, dans certaines conditions limites, les objectifs ne peuvent pas être atteints.

Propriété du domaine

Assertion descriptive des objets \* dans l'environnement \* du logiciel. Il peut s'agir d'un invariant de domaine ou d'une hypothèse. Un invariant de domaine est une propriété connue pour contenir dans chaque état d'un objet de domaine, par exemple, une loi physique, une régulation, ... Une hypothèse est une propriété sur un objet de domaine supposé détenir.

Entité

Objet autonome \*, c'est-à-dire dont la définition ne repose pas sur d'autres objets.

Environnement

Partie de l'univers capable d'interaction avec le logiciel étudié.

Un événement

Objet instantané \* (c'est-à-dire un objet vivant dans un seul état) qui déclenche des opérations \* effectuées par des agents \*.

Attente

Objectif \* attribué à un agent \* dans l'environnement \*.

Modèle formel

Modèle \* dans lequel les concepts ont été mathématiquement formalisés. Dans Objectiver, le modèle formel est construit sur un modèle semi-formel. Seule une partie ou tout le modèle semi-formel peut être formalisé. Objectiver utilise la logique temporelle de premier ordre avec des extensions en temps réel pour formaliser les concepts.

Objectif

Assertion prescriptive capturant un objectif à atteindre par la coopération des agents \*; Il prescrit un ensemble de comportements souhaités. Les exigences \* et les attentes \* sont des objectifs.

Maquette

Représentation abstraite d 'un système composite. Un modèle d'Objectiver représente un système composite \* au moyen de concepts de différents types, principalement des objets \*, des propriétés désirées ou indésirables (buts \*, obstacles \*) et des comportements (opérations \*).

Objet

Chose d'intérêt dans le système composite \* étant modélisé dont les instances peuvent être distinctement identifiées et peuvent évoluer d'un état à l'autre. Agents, événements, entités et associations sont des objets.

Obstacle

Condition (autre qu'un but) dont la satisfaction peut empêcher l'atteinte d'un ou de plusieurs objectifs \*; Il définit un ensemble de comportements indésirables.

Opération

Spécifie les transitions d'état des objets \* qui sont entrées et / ou sorties de l'opération. Les opérations sont effectuées par des agents \*.

Opérationnalisation

Relation reliant une exigence \* aux opérations \*. Détient lorsque chaque exécution des opérations (éventuellement contrainte à cette intention) entraînera l'exigence \*. Fait le lien entre les propriétés attendues (buts \*) et les comportements (opérations \*)

Raffinement

Relation liant un but \* à d'autres objectifs qui sont appelés ses sous-objectifs. Chaque sous-but contribue à la satisfaction de l'objectif \* qu'il affine. La conjonction de tous les sous-objectifs doit être une condition suffisante impliquant l'objectif qu'ils affinent.

Exigence

Objectif \* attribué à un agent \* du logiciel étudié.

Responsabilité

Relation entre un agent \* et une exigence \*. Détient quand un agent \* a la responsabilité d'atteindre l'exigence liée \*.

Modèle semi-formel

Modèle dans lequel les concepts ne sont pas mathématiquement formalisés Dans Objectiver, chaque concept du modèle reçoit un nom, un type, une définition textuelle, des valeurs pour les attributs et une représentation graphique.

Sources :

<http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf>

<https://hal.archives-ouvertes.fr/hal-01084878>

<https://pdfs.semanticscholar.org/63d8/94c192ea2895f25888e238856dd441ae3fec.pdf>