

Questionnaire d'approfondissement

Cours Systèmes Multi-Agents Master Développement Logiciel

1. Dans la section 1.2, il est fait référence au terme "agent" et à ses différentes significations. Combien y a t il de communautés différentes qui selon vous utilise ce terme et quelles différences apparaissent dans leur définition respective du mot "agent"?
2. En section 1.2.2, nous faisons référence aux modes "push" et "pull" de transferts d'information. Savez vous assurément les distinguer?
3. Dans cette même section, nous avons employé le terme "machine abstraite". A mon sens, cela est comparable à la notion de "machine virtuelle". Cette comparaison fait elle écho à vos oreilles? Pourquoi?
4. Avant de passer à la section 1.2.3, nous faisons (encore) référence au mode "asynchrone" d'exécution (ou d'interaction) des agents. Selon vous, combien de modes y a t il et quels sont ces autres modes?
5. En fin de sous-section 1.2.3.1, nous parlons d'adaptation par modification de l'organisation du système (et même d'auto-organisation). Il existe d'autres façons de modifier le comportement global d'un système. Combien et lesquelles?
6. Il existe un type particulier de SMA reconnaissables par leur structure organisationnelle particulière. Il s'agit des holons. Kesaco?
7. Les mécanismes d'interactions dans un SMA situé se révèlent souvent très simples et très puissants. La stigmergie (section 1.2.3.2) en est un. Savez vous quelle propriétés sur les chemins ont les "algos de fourmis"? Connaissez vous ce qu'est une descente de gradient? Pour les déplacements dans un environnement situé, on utilise parfois un "tirage de monte-carlo" pour introduire de l'indeterminisme dans les déplacements. Savez vous "tirer" de la sorte?
8. En section 1.2.5.1, nous abordons les processus de méthodes de développement SMA. D'un point de vue processus, l'agilité (Scrum en tête) semble être la réponse la plus aboutie à la flexibilité nécessaire pour la prise en compte des variations d'exigences au cours du développement. Pourtant, parmi les processus et méthodes SMA que nous avons vus ensemble, pas d'agilité en vue alors que flexibilité et adaptation semblent être des exigences récurrentes. Votre avis sur ce constat?
9. Dans la section 1.3, nous abordons les aspects architecturaux liés à l'ingénierie des SMA, ce qui nous fera aborder les exigences qui y sont relatives. Avec votre expérience/contexte de stage, quelles sont les exigences qui peuvent vous "attirer" vers une solution SMA?
10. Le vocabulaire que nous avons utilisé a été mûrement réfléchi. Dans un tel chapitre qui traite d'architecture logicielle et d'ingénierie logicielle à base de systèmes multi-agents, nous n'avons pas confondu les termes paradigme, approche, plateforme, style, patron de conception et patron d'architecture. Ont-ils bien des sens différents pour vous ?
11. Dans la sous-section 1.3.2.1 qui traite du haut niveau d'abstraction du style SMA, nous parlons de la vision que promeuvent les paradigmes de programmation et en particulier une vision inspirée de la machine. Retrouvez-vous cette « machine » dans les paradigmes que vous connaissez ?

12. Nous prétendons que la force du style SMA provient de la séparation des volets social et autonomie ? Est-ce une particularité de ce style ?
13. A partir de cette même sous-section 1.3.2.1, nous insistons beaucoup sur la « distance » entre les domaines du problème et de l'approche. « Sentez » vous bien cette distance ? En avez-vous déjà fait l'expérience ?
14. Ensuite, dans la partie 1.3.2.2, nous voulons faire remarquer que le degré d'abstraction est variable dans une conception avec SMA et que cela fait l'originalité de ce style. Comprenez-vous bien cette originalité et le potentiel qu'elle représente pour le concepteur ?
15. En matière d'exigences, la fin de la partie 1.3.2.4 soulève les faiblesses du style SMA vis-à-vis des patrons de conception objet, notamment la flexibilité (par exemple Inversion of Control et Dependency Injection) ou extensibilité (par exemple la Fabrique ou le polymorphisme). Connaissez-vous ces patrons ? Sont-ils effectivement, pour vous, des réponses à ces exigences ? Comprenez-vous cette faiblesse ?
16. A propos d'inversion de contrôle, la sous-section 1.3.3.1 parle d'inversion de contrôle « sémantique » que l'on trouve dans le style SMA. En quoi le patron de conception est-il structurel et le style SMA « sémantique » ?
17. Toujours sur le registre « sémantique », la partie suivante (1.3.3.2) parle de couplage sémantique pour la composition des agents. On retrouve cette idée dans d'autres approches à composants ou services. En connaissez-vous ?
18. La partie sur l'émergence (1.3.3.3) insiste surtout sur les avantages du style SMA vis-à-vis d'une exigence sur la dynamique de la composition. On retombe sur une métaphore qui m'est chère entre bug et plante indésirable. Vous me suivez ?
19. Le problème de la vérification et de la validation des logiciels à base de SMA est posé à la section 1.3.4.4. Qu'y a-t-il de particulier dans cette approche ?
20. La fin du chapitre se focalise sur un point : tout l'objectif est de réduire au minimum la distance entre problème et approche. La partie 1.5 propose une solution pour réduire l'effort entre le domaine conceptuel et celui de l'approche en basant cette dernière sur les notions d'objets et de composants. Le « fossé » se réduit mais n'est pas comblé. Selon vous, une approche basée sur l'ingénierie dirigée par les modèles permet-elle de combler complètement ce fossé ?