
Caractérisation de l'environnement d'exécution pour la conception d'un système d'information mobile et distribué

Mathieu Petit — Cyril Ray — Christophe Claramunt

Institut de recherche de l'Ecole Navale

BCRM de Brest, Ecole navale

CC 600, 29240 Brest Cedex 9, France

{mathieu.petit, cyril.ray, christophe.claramunt}@ecole-navale.fr

RÉSUMÉ. Cet article présente une approche géographique pour la conception d'un système d'information mobile et distribué. La démarche proposée considère en plus de la description des tâches de l'utilisateur, une description de l'environnement d'exécution du système comme point d'entrée de la conception. Cette description repose sur une caractérisation de l'espace géographique réparti en régions types qui vont définir des situations contextuelles propres à enrichir ou à contraindre les tâches de l'utilisateur et les possibilités fonctionnelles offertes par le système. Cette caractérisation de l'environnement est intégrée à une méthode de conception éprouvée, centrée sur la décomposition de la tâche de l'utilisateur à partir d'une description scénarisée de ses intentions et de ses buts. Chaque étape de la conception est décrite dans le cadre de l'analyse et de la mise en œuvre d'un système de suivi de compétition nautique.

ABSTRACT. The research presented in this paper introduces the principles of a geographic approach to the design of a mobile and adaptive information system. This conceptual framework considers, besides the user goals and intentions, a geographic description of the execution environment as a main input of the design process. This description characterises several regions of interest, whose interactions are likely to enhance the functional capabilities at the system level. The environmental constraints are integrated within a user-centred design approach, on top of a conceptual model. This approach is illustrated by the design of a maritime tracking system.

MOTS-CLÉS : Informatique nomade, Conception centrée utilisateur, Système d'information géographique adaptatif

KEYWORDS: Ubiquitous computing, User-centered design, Adaptive geographic information system

1. Introduction

Les avancées technologiques observées depuis le milieu des années 1990 ont favorisé l'émergence de nouveaux modes d'utilisation de l'outil informatique. Dans le domaine des systèmes d'information, l'internet, les technologies de transmission sans fil et la démocratisation des systèmes de positionnement permettent l'utilisation nomade de systèmes d'information et la localisation de l'utilisateur dans son environnement. La multiplicité des plateformes d'utilisation a initié la répartition des systèmes en différents tiers et les architectures logicielles sont désormais distribuées. Un système d'information doit dorénavant faire le lien entre une utilisation en mobilité et une plateforme distribuée évoluant dans un espace géographique. Ces nouvelles contraintes d'architecture et d'utilisation des systèmes d'information demandent une conception appropriée, qui caractérise et intègre la versatilité des conditions d'exécution (Satyanarayanan, 1996).

La versatilité de l'environnement d'utilisation, prise dans le sens de l'informatique contextuelle, est abordée dans le cadre des systèmes interactifs (Edmonds, 1999). Plusieurs modèles pour la formalisation des contextes d'utilisation ont été proposés. Les Contexteurs sont des briques logicielles interconnectées qui captent des flux d'informations contextuelles et adaptent le système à différentes situations d'interaction (Rey *et al.*, 2004). Dans le modèle du Context-Toolkit, les « interacteurs de contextes » fusionnent des signaux en provenance de capteurs ; les « interpréteurs » dotent ces mesures d'une sémantique ; puis, les « agrégateurs » conditionnent l'application en fonction de l'interprétation du contexte (Dey *et al.*, 2001). Sans toujours se conformer à un formalisme de description de contexte, l'utilisation simple du contexte, et en particulier des informations de localisation, est souvent présente en informatique mobile (Schmidt *et al.*, 1999). Dans ce sens, « PhotoMap » propose un outil d'étiquetage contextuel de photographie en fonction du lieu de la prise de vue (Viana *et al.*, 2007). Des outils reposant sur la localisation de l'utilisateur (Reichenbacher, 2003) où sur sa manipulation de cartes numériques (Mac Aoidh *et al.*, 2007) ont été proposés pour l'adaptation d'un contenu cartographique sur un terminal mobile.

Ces outils et ces exemples d'utilisations possibles du contexte adaptent dynamiquement les capacités interactives d'un système. L'objectif consiste à offrir à l'utilisateur une plus grande souplesse d'interaction en adaptant le contenu proposé (*niveau des données*), les moyens d'agir sur ce contenu (*niveau de l'interaction*), et la façon dont il est présenté (*niveau de l'interface*). La variété des plateformes client, qui dépasse aujourd'hui largement le simple ordinateur de bureau, et l'utilisation nomade de systèmes d'information impliquent d'associer la notion de *système* au contexte. Ce dernier influence les procédures disponibles pour l'utilisateur et modifie sa tâche en adaptant les capacités fonctionnelles du système. L'information géographique sur l'environnement d'exécution défini par la distribution spatiale des composants du système, constitue un point d'entrée pour une démarche de conception qui intègre les contextes du niveau du système. La description géographique d'un système localisé à l'exécution caractérise les différentes situations contextuelles qui influencent son comportement à l'exécution. Le modèle proposé introduit des propriétés d'équiva-

lence entre contextes et des contraintes de mobilité des composants qui restreignent le nombre de cas de contextes d'exécution à prendre en compte lors de la conception. Les contraintes contextuelles sont intégrées dans un modèle conceptuel centré utilisateur, qui propose des comportements alternatifs du système adaptés à ces nouvelles situations. L'ensemble de la démarche de conception, depuis l'analyse géographique de l'environnement, jusqu'à la réalisation des maquettes de l'interface utilisateur, est expérimenté dans le cas de la mise au point d'un système de suivi de compétition nautique.

2. Conception de systèmes d'information et contexte d'usage

Plusieurs démarches de conception de systèmes d'information en contexte ont été proposées (Eisenstein *et al.*, 2001; Luyten *et al.*, 2003; Paterno *et al.*, 2003). Elles se conforment au cadre de référence unifié pour la modélisation d'un système interactif (Calvary *et al.*, 2003). En premier lieu, elles répondent à la question "quels sont les objectifs des utilisateurs du système?" puis énoncent les étapes nécessaires à l'accomplissement de ces objectifs par un ensemble de *tâches* s'appuyant sur des *informations* caractérisant les données d'entrée-sortie utilisées par le système. Cette description est détaillée au sein d'un document textuel : *le scénario nominal*. Il s'agit d'une description informelle du fonctionnement du système construite à partir des besoins des utilisateurs. Dans ce sens, les experts du domaine, les concepteurs du système et éventuellement les futurs utilisateurs se réunissent en groupe de travail pour écrire ce scénario. Ce document servira de fil conducteur à l'équipe de développement pour la suite du processus de conception du système.

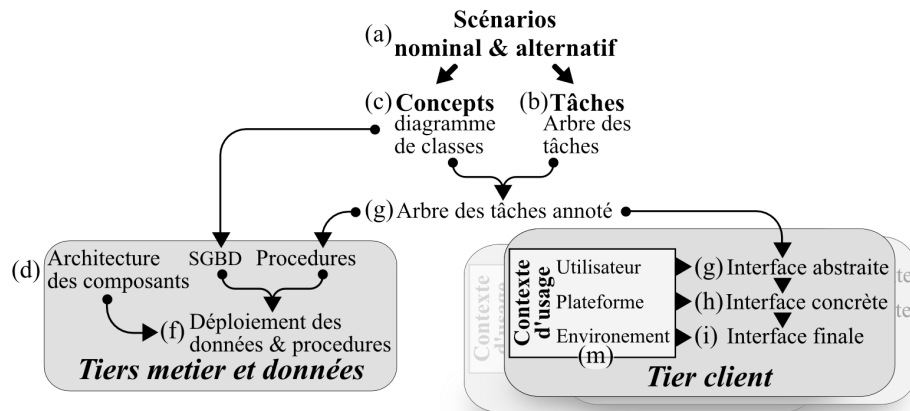


Figure 1. Cadre de conception d'un système d'information adaptatif

L'écriture du scénario nominal pour la description des tâches s'établit comme le point d'entrée d'une large majorité des méthodes de conception centrées utilisateur

(Bastide *et al.*, 2003). En partant de cette spécification, les tâches et les informations du domaine sont décomposées et modélisées selon une conception descendante :

- 1) organisation des informations du système en un ensemble de données informatiques (figure 1(c)),
- 2) décomposition et organisation des tâches de l'utilisateur (figure 1(b)) ;
- 3) conception des tiers métier et données : cette étape aborde le choix de la distribution physique du système (figure 1(d)), l'implémentation du système de gestion de données et du cœur métier (figure 1(c)), et le déploiement sur l'architecture (figure 1(f)) ;
- 4) conception du tiers client (Calvary, 2007) : l'interface abstraite décompose l'interaction graphique avec le système en dédiant un espace de dialogue à chaque tâche (figure 1(g)). L'interface concrète organise les interacteurs (listes, menus, boutons, etc.) au sein de chaque espace de dialogue (figure 1(h)). Enfin, l'interface finale propose une implémentation logicielle de l'interface concrète, en se reposant sur une bibliothèque de composants graphiques (Myers, 1995).

Les étapes de conception des tiers "métier", "données" et "client" peuvent être semi-automatiques. Pour le tiers métier, l'organisation des informations est assortie de méthodes de manipulation pour former un diagramme de classes, et un patron de code est généré. Pour le tiers client, des outils comme Theresa permettent la génération des différents niveaux d'abstraction de l'interface utilisateur à partir d'une description formalisée des tâches (Mori *et al.*, 2004). L'ensemble du processus de conception est répété. Chaque itération ajuste ou enrichi le cycle précédent, jusqu'à obtenir une version du système qui satisfait entièrement aux exigences des utilisateurs (Boehm, 1986).

Le contexte d'usage, qui adapte dynamiquement les capacités interactives du système, est pris en compte au niveau de la conception du tiers client. Le cadre de référence unifié distingue trois dimensions contextuelles dans l'environnement immédiat de l'utilisateur : le contexte utilisateur, le contexte de la plateforme client, et le contexte d'environnement d'utilisation (figure 1(m)). Chaque triplet du contexte d'usage engendre la conception d'une partie cliente qui est *adaptée* pour l'utilisateur considéré.

3. Environnement géographique d'un système d'information

L'utilisation d'un système d'information mobile et distribué impose à la conception une réflexion sur les conditions d'exécution de ce système. L'étude de l'environnement géographique d'exécution amène les concepteurs à décrire l'ensemble des possibilités d'architectures des composants du système. Ces architectures, variables à l'exécution, sont groupées en fonction des fonctionnalités qu'elles activent pour l'utilisateur. Chaque composant tient potentiellement un rôle dans le système. En se basant sur une architecture de tiers, nous distinguons trois familles de composants : les serveurs de traitement, les serveurs de données, et les plateformes clientes.

Cette décomposition du système en tiers répartis est expérimentée dans le contexte d'un championnat annuel de voile¹. Cet évènement rassemble un large public, entre compétiteurs, passionnés de voile et visiteurs découvrant le monde de la mer. Les régates du championnat ont parfois lieu loin de la côte et les spectateurs suivent avec difficulté l'évolution des bateaux au loin. Pour proposer un complément d'information, un système de suivi à distance est mis en place. Un module de positionnement en temps réel a été développé et équipe les bateaux (figure 2(a)) (Claramunt *et al.*, 2007). Leurs positions GPS sont transmises par communication sans fil (bande ISM) à une station de collecte qui fait office de serveur de données (figure 2(c)). Les utilisateurs peuvent suivre l'évolution des bateaux sur un dispositif mobile (figure 2(b)) en accédant aux procédures de suivi diffusées depuis le serveur de traitement (figure 2(d)).

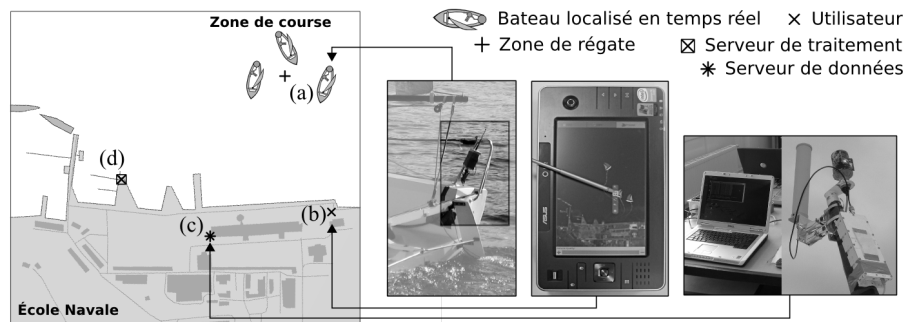


Figure 2. Infrastructure du système de suivi en temps réel

Ce cas d'étude illustre un système d'information construit sur trois composants matériels, et qui intègre des données localisées. L'organisation des composants et les échanges d'informations entre ceux-ci ne sont pas fixés. Chaque composant, potentiellement mobile, rend l'architecture du système variable à l'exécution et l'adaptation nécessaire. Par exemple, le serveur de données étant différencié du serveur de traitement, il est possible que les procédures de traitements ne reçoivent plus d'informations à un moment de l'exécution. Ce contexte d'exécution doit être caractérisé à la conception en prévoyant un comportement approprié du système. L'utilisation de technologies de transmission sans fil dote chaque composant d'une région d'influence dans laquelle il est capable d'envoyer et de recevoir des informations. Ces régions, et leurs intersections, constituent le point d'entrée de la description des différents contextes d'exécution.

1. <http://www.gpen.ecole-navale.fr>

3.1. Architecture distribuée et environnement d'exécution

Un système d'information interactif, qui stocke, traite et transforme de l'information à l'initiative et à destination d'utilisateurs, distingue les niveaux des données, des traitements et du client. D'un point de vue fonctionnel, une architecture d'un système d'information en tiers répartis implémente ces niveaux sur un ensemble de composants matériels dédiés. Par exemple, le composant « Interface client » C_{U_x} se charge des interactions avec l'utilisateur et de l'affichage des informations demandées. Le composant « Serveur de données » C_{D_x} importe et exporte les sous-ensembles de données à traiter. Enfin, le composant « Serveur de traitement » C_{P_x} , héberge les procédures de manipulation et d'analyse des données (Eckerson, 1995).

3.1.1. Définition des régions d'intérêt

Le fonctionnement du système se fonde sur l'échange d'informations entre les différents tiers, depuis la source des données, jusqu'à l'interface utilisateur. En considérant l'espace géographique d'exécution d'un système, il est possible de doter chacun des composants du système d'une empreinte spatiale, qui matérialise la région dans laquelle chaque composant est capable d'échanger des informations (Longley *et al.*, 2005). Cette définition de la répartition spatiale d'un système d'information, nous permet de distinguer un ensemble de régions d'intérêt qui caractérisent l'environnement géographique d'exécution du système :

- la ou les région(s) P_x des traitements, où les procédures utiles pour la réalisation de traitements sur des données sont disponibles ;
- la ou les région(s) D_x de diffusion des données, où les données brutes du système sont disponibles ;
- la ou les région(s) U_x du/des l'utilisateur(s), où le résultat de tout ou d'une partie des traitements effectués sont diffusables ;
- la ou les région(s) S_x d'origine sémantique des données du système, qui matérialisent l'origine de tout ou d'une partie des informations localisées.

La délimitation des régions U_x , D_x et P_x est respectivement issue des caractéristiques des composants matériels support de ces régions. Dans l'exemple du système de suivi de navigation, les emprises spatiales des régions de diffusion des données D_1 , de traitement P_1 et de l'utilisateur U_1 sont liées aux capacités de transmission sans fil des serveurs C_{D_1} et C_{P_1} , et du client C_{U_1} (figure 3(a)). Ces régions, qui reposent sur un composant matériel du système, sont dites *actives*, dans le sens où la connaissance des caractéristiques du composant sous-jacent suffit à déterminer l'empreinte spatiale de la région.

Une région S_x d'origine des données existe dans le cas des systèmes qui manipulent au moins pour partie des informations localisées. Elle représente la région de l'espace de provenance géographique des données. Au contraire des autres régions, les régions d'origine des données sont *inactives*, dans le sens où leur emprise spatiale est définie par des données issues d'un composant non sous-jacent. Dans le cas du suivi

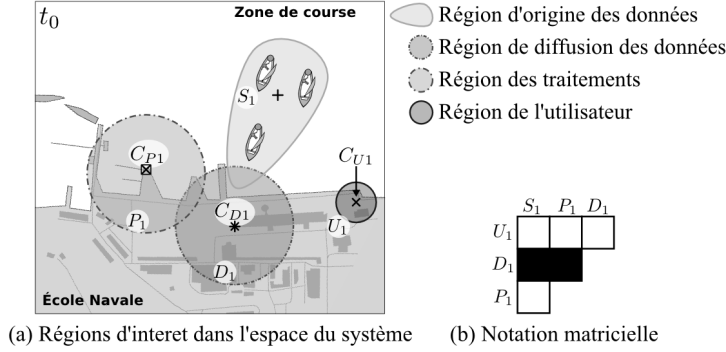


Figure 3. Décomposition géographique de l'environnement du système de suivi de régates

de régates, l'origine des données couvre les positions des bateaux suivis (figure 3(S_1)) et ces informations de localisation sont disponibles dans la région D_1 : la région S_1 est construite par rapport aux données disponibles au niveau du serveur C_{D1} sous-jacent à la région D_1 .

3.1.2. Contexte d'exécution

Les régions d'intérêt forment les constituants du contexte d'exécution. Elles caractérisent l'état de l'architecture du système à un instant donné et influencent le niveau fonctionnel atteignable dans cette situation. Si dans le système de suivi le composant C_{D1} tombe en panne, alors la région D_1 est perdue, les données sont indisponibles et le suivi des bateaux est impossible. Toutefois, la région des traitements reste active et doit adapter les procédures diffusées à ce mode de fonctionnement dégradé. Ces situations sont deux exemples de contextes d'exécution d'un même système.

Plus généralement, dans un système réparti, le contexte d'exécution évalue les capacités à échanger des informations entre les composants actifs de ce système. Comme l'emprise spatiale d'une région d'intérêt est définie par rapport à cette capacité d'échange (section 3.1.1), le contexte d'exécution est donné par l'ensemble des paires de régions en intersection non vide. Par exemple, à t_0 , le contexte d'exécution du système de suivi s'écrit $\{P_1 \cap D_1 \neq \emptyset, S_1 \cap D_1 \neq \emptyset\}$ (figure 3). Comme D_1 et P_1 s'intersectent, ce contexte caractérise une architecture dans laquelle l'échange d'information est possible entre les composants C_{P1} et C_{D1} , mais où l'utilisateur reste isolé.

Une notation matricielle matérialise les interactions entre ces régions (figure 3(b)). Par convention, une cellule noire représente le cas de deux régions à intersection non vide, tandis qu'une cellule blanche représente deux régions disjointes.

3.2. Dynamique des régions et contexte d'exécution

Dans un système distribué et mobile, l'empreinte spatiale des régions évolue en fonction du temps. Ces empreintes spatiales sont associées aux composants du système dans le cas de régions D_x , P_x et U_x . Pour la région d'origine, les variations dans la localisation des données modifient l'empreinte spatiale de S_x . Chaque configuration distinguant un contexte d'exécution, il existe, dans le système de suivi de régates distribué en quatre régions, 64 configurations et contextes possibles. Chaque contexte entraîne un comportement du système adapté (Petit *et al.*, 2006). Dans le cas général d'un système à n régions d'intérêt, l'ensemble des contextes d'exécution croît de façon exponentielle en $O(2^n)$. Plus le système contient de régions, et plus il devient difficile de caractériser l'ensemble des contextes d'exécution, et de définir un comportement propre à chacun de ces contextes. Afin de limiter cette croissance, des cas d'équivalence entre configurations et des contraintes au niveau de la mobilité des régions sont introduits.

3.2.1. Équivalences contextuelles

Un contexte d'exécution caractérise la distribution et les relations des composants du système à un instant donné. L'échange d'informations entre composants dépend de la localisation des régions et de leur étendue. En particulier, quand l'utilisateur se trouve en dehors des régions d'intérêt, ou quand des groupes de régions ne sont pas connectés, il existe au niveau de chaque composant un contexte propre qui est une sous-partie du contexte d'exécution. Il représente le contexte partiel dérivé au niveau de chaque composant en considérant ses capacités à échanger des informations avec les autres tiers.

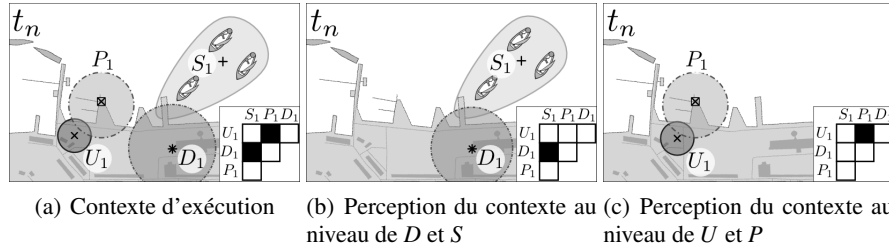


Figure 4. Exemples d'interprétations locales du contexte d'exécution

Dans l'exemple de la figure 4(a), les groupes de régions $\{D_1, S_1\}$ et $\{P_1, U_1\}$ sont disjoints. Le composant C_{D_1} n'échange pas d'information avec C_{U_1} ou C_{P_1} , et n'a pas connaissance de la situation des régions P_1 et U_1 . Au niveau de la région D_1 , le contexte d'exécution se résume à $\{D_1 \cap S_1 \neq \emptyset\}$ (figure 4(b)). Similairement, une interprétation locale du contexte d'exécution est faite au niveau des régions U_1 et P_1 , où l'on perçoit uniquement $\{P_1 \cap U_1 \neq \emptyset\}$ (figure 4(c)).

Avec l'objectif d'offrir une flexibilité fonctionnelle pour l'utilisateur, le contexte du point de vue de sa région est déterminant. Si deux configurations différentes de régions sont perçues de la même manière au niveau de U_x , alors elles sont *équivalentes* et l'une comme l'autre donneront lieu à la même adaptation fonctionnelle du système. Perçues comme $\{U_1 \cap P_1 \neq \emptyset\}$ localement à U_1 (figure 4(c)), les configurations de régions $\{P_1 \cap U_1 \neq \emptyset, D_1 \cap S_1 \neq \emptyset\}$ (figure 4(a)) et $\{P_1 \cap U_1 \neq \emptyset\}$ sont équivalentes.

Un ensemble de propriétés d'équivalences entre configurations de régions est défini dans la suite de la section. Ces propriétés sont illustrées par le cas d'étude du suivi de régate. Chaque équivalence s'exprime relativement au point de vue de l'utilisateur et les propriétés retenues groupent les contextes selon le niveau fonctionnel offert à l'utilisateur. Les propriétés portent sur le rapport de la région de l'utilisateur aux autres types de région : traitements, origine ou données. Dans le cas d'un système non interactif, les concepteurs énoncent d'autres propriétés d'équivalences. Celles-ci se bornent, par exemple, à décrire les contextes équivalents pour une adaptation fonctionnelle au niveau du serveur de traitement.

Propriété 1 (Utilisateur seul)

Lorsque la région de l'utilisateur U_x n'intersecte aucune région d'intérêt, le système est hors de portée de cet utilisateur. Les configurations présentant une région de l'utilisateur isolée des autres sont équivalentes.

Dans l'exemple proposé (figure 5(a)), entre t_n et t_{n+x} , l'intersection des régions D_1 et P_1 devient vide. Le système reste toujours hors de portée de U_1 et selon l'utilisateur, le contexte est interprété comme vide. Les contextes d'exécution $\{P_1 \cap D_1 \neq \emptyset\}$ et $\emptyset$ sont équivalents selon la propriété « Utilisateur seul ».

Propriété 2 (Données indisponibles)

Si la région des données D_x n'intersecte ni la région de l'utilisateur, ni la région des traitements, alors aucune information brute ou information traitée n'est présentée à l'utilisateur. Des configurations présentant une région D_x disjointe de U_x et P_x sont équivalentes.

À t_n comme à t_{n+x} , la région des données D_1 n'est intersectée ni par l'utilisateur, ni par les traitements. Les données ne peuvent pas être traitées au niveau de C_{P_1} ni présentées sur le client C_{U_1} . Les contextes d'exécution $\{U_1 \cap P_1 \neq \emptyset\}$ et $\{P_1 \cap U_1 \neq \emptyset, S_1 \cap D_1 \neq \emptyset\}$ sont équivalents selon la propriété « Données indisponibles » (figure 5(b)).

Propriété 3 (Traitement indisponible)

Si la région des traitements P_x n'est pas accessible à l'utilisateur, alors la situation de P_x n'intervient pas dans l'adaptation du système au contexte d'exécution. Du point de vue de l'utilisateur, aucun résultat de traitement de données n'est retourné. Les configurations égales à l'exception des intersections partagées avec la région P_x , et dans lesquelles U_x n'intersecte pas P_x , sont équivalentes.

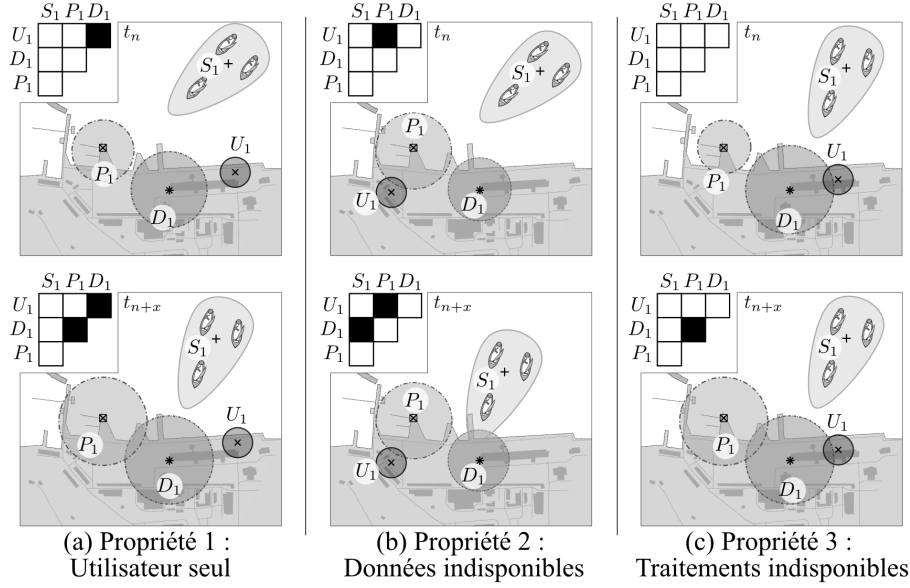


Figure 5. Exemples d'équivalences de contexte d'exécution

Entre t_n et t_{n+x} , la région de traitement P_1 n'est jamais accessible à l'utilisateur. Les résultats du traitement des données ne peuvent pas être répercutés au niveau du composant C_{U_1} . Les contextes d'exécution $\{U_1 \cap D_1 \neq \emptyset\}$ et $\{P_1 \cap D_1 \neq \emptyset, U_1 \cap D_1 \neq \emptyset\}$ sont équivalents selon la propriété « Traitement indisponible » (figure 5(c)).

Propriété 4 (Origine distante)

La positionnement de U_x par rapport à la région d'origine des données S_x permet de distinguer deux niveaux fonctionnels, entre un utilisateur « acteur » en proximité spatiale de la source des données localisées, et un utilisateur « spectateur », observateur lointain de la région d'origine. Du point de vue fonctionnel, les configurations d'intersections non vides et supplémentaires entre S_x et les régions D_x et P_x n'influencent pas l'adaptation du système. Les configurations égales à l'exception des intersections non vides et partagées avec la région S_x , et dans lesquelles U_x n'intersecte pas S_x , sont équivalentes.

Dans l'exemple proposé (figure 6(a)), entre t_n et t_{n+x} , les régions D_1 et S_1 ne s'intersectent plus. Durant l'intervalle, l'utilisateur en U_1 reste spectateur éloigné de la région d'origine. Le passage d'un contexte à un autre ne provoque pas d'adaptation fonctionnelle. Les contextes d'exécution $\{P_1 \cap D_1 \neq \emptyset, U_1 \cap D_1 \neq \emptyset, D_1 \cap S_1 \neq \emptyset\}$ et $\{P_1 \cap D_1 \neq \emptyset, U_1 \cap D_1 \neq \emptyset\}$ sont équivalents selon la propriété « Origine distante ».

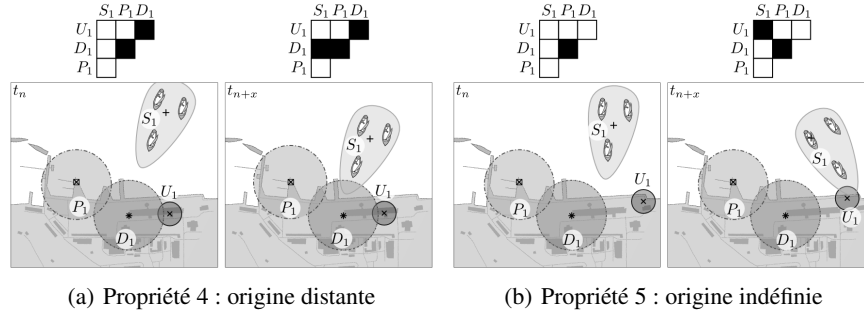


Figure 6. Exemples d'équivalences de contexte d'exécution

Propriété 5 (Origine indéfinie)

Pour détecter une intersection avec la région d'origine S_x , l'utilisateur doit échanger des informations avec le composant C_{D_x} qui définit l'empreinte spatiale de S_x . Les configurations égales, à l'exception de l'intersection entre U_x et S_x , et dans lesquelles le composant C_{U_x} n'échange pas d'information avec le composant définissant la région d'origine S_x , sont équivalentes.

Dans l'exemple proposé (figure 6(b)), la région S_1 définie par les bateaux suivis va intersecter la région de l'utilisateur à t_{n+x} . Comme la variation de l'emprise spatiale de la région S_1 n'est pas transmise entre D_1 et U_1 , la promotion de l'utilisateur en tant qu'acteur n'est pas détectée et le système ne s'adapte pas. Les configurations $\{P_1 \cap D_1 \neq \emptyset\}$ et $\{P_1 \cap D_1 \neq \emptyset, U_1 \cap S_1 \neq \emptyset\}$ sont équivalentes selon la propriété « Origine indéfinie ».

3.2.2. Contraintes de mobilité

Dans la plupart des systèmes décrits, il est possible de borner la mobilité des composants dans des portions de l'espace qui représentent les limites fonctionnelles du système. Nous définissons l'espace de mobilité ζ_{R_x} comme la portion de l'espace que peut occuper l'empreinte spatiale d'une région d'intérêt $R_x \in \{U_x, D_x, P_x, S_x\}$ au cours de l'exécution du système. Une région R_x est mobile lorsque R_x est strictement incluse dans ζ_{R_x} (i.e., $R_x \subsetneq \zeta_{R_x}$) et est fixe dans l'environnement quand ζ_{R_x} est égale à R_x (i.e., $\zeta_{R_x} = R_x$). Les espaces de mobilité du système limitent le nombre de contextes d'exécution possibles :

- avec $R_x \in \{U_x, D_x, P_x, S_x\}$, lorsque tous les espaces de mobilité ζ_{R_x} tels que R_x est mobile sont disjoints et que les espaces de mobilité ζ_{R_x} qui s'intersectent sont contraints par $\zeta_{R_x} = R_x$, une seule configuration de région et un seul contexte d'exécution sont définis. Le système est *fortement contraint* ;
- lorsque tous les espaces de mobilité du système sont en intersection non vide deux à deux, et que chaque région R_x est mobile le système est dit *non contraint*. Dans

ce cas, seules les propriétés d'équivalences entre configurations de régions permettent de limiter la croissance exponentielle du nombre de cas de contextes d'exécution (section 3.2.1).

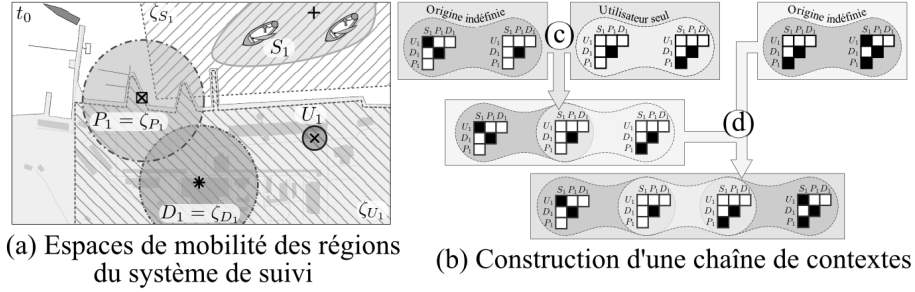


Figure 7. Mobilité des régions et chaîne de contextes équivalents

Un système *faiblement contraint* correspond à un nombre de contextes d'exécution variant dans l'intervalle $[1, 2^N]$, où N est le nombre de régions d'intérêts du système.

Dans le cas du suivi de régate (figure 7(a)), l'implantation des serveurs C_{P_1} et C_{D_1} est contrainte par les possibilités techniques du site et les besoins des organisateurs. Les données diffusées pour le système de suivi doivent être disponibles dans l'espace dédié aux spectateurs, mais surtout rester hors de la zone de course. Les organisateurs décident de limiter la portée des données temps-réel. L'espace de mobilité $\zeta_{D_1} = D_1$ fixe la région D_1 autour de C_{D_1} (figure 7(D_1)). Les procédures de traitements sont diffusées dans une région à forte concentration d'utilisateurs, près de l'arrivée. Le composant serveur C_{P_1} est fixe et $\zeta_{P_1} = P_1$ (figure 7(P_1)). Les utilisateurs sont mobiles dans un espace ζ_{U_1} qui s'étend le long de la ligne de côte. La zone maximale de la course définit l'espace de mobilité ζ_{S_1} de la région d'origine S_1 (figure 7(ζ_{S_1})).

En conséquence de ces contraintes de mobilité, le système de suivi vérifie trois propriétés. D'une part, comme $\zeta_{D_1} \cap \zeta_{P_1} \neq \emptyset$, l'intersection des régions fixes D_1 et P_1 n'est jamais vide. Les composants C_{D_1} et C_{P_1} sont toujours capables d'échanger des informations. D'autre part, la région de l'utilisateur U_1 , mobile dans ζ_{U_1} , ne peut être à la fois en intersection avec D_1 et S_1 . Enfin, comme l'intersection des espaces de mobilité ζ_{S_1} et ζ_{D_1} est vide, les régions D_1 et S_1 sont toujours disjointes. Ces propriétés permettent d'exclure de l'ensemble des contextes observables les configurations qui ne contiennent pas l'élément $D_1 \cap P_1 \neq \emptyset$, celles qui contiennent l'élément $D_1 \cap S_1 \neq \emptyset$, et celles qui ont $\{D_1 \cap U_1 \neq \emptyset, U_1 \cap S_1 \neq \emptyset\}$ comme sous-ensemble. Pour le système de suivi, qui définit les quatre régions P_1 , U_1 , D_1 et S_1 , l'ensemble des contextes est réduit de $2^4 = 64$ cas à onze effectivement observables (figure 8).

Des groupes de contextes équivalents, issus de l'application des propriétés, sont combinés pour former des chaînes de contextes générant la même adaptation pour l'utilisateur. Si un contexte donné est présent dans deux groupes d'équivalence, alors ces groupes forment une chaîne d'équivalence. Par exemple, la chaîne représentée

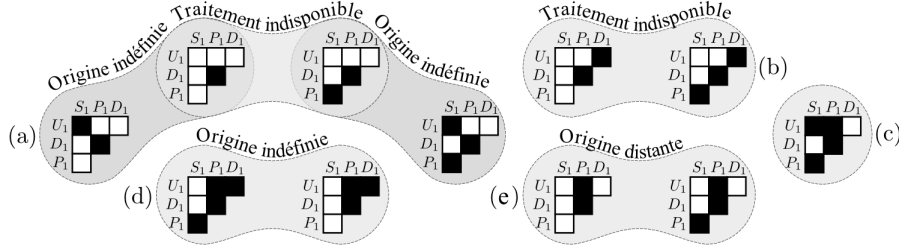


Figure 8. Chaînes de contextes d'exécution du système de suivi

en figure 8(a) est construite en combinant deux groupes de contextes équivalents par les propriétés *origine indéfinie* et *traitements indisponibles* (figure 7(c)). Une troisième paire de contextes équivalents par la propriété *origine indéfinie* est combinée à la chaîne (figure 7(d)). Les contextes de cette chaîne sont tous équivalents. Dans ce cas, l'utilisateur ne peut jamais communiquer avec les composants actifs du système. Plus généralement, un contexte appartient au moins à une et une seule chaîne d'équivalence, ces dernières formant des ensembles disjoints. Dans le cas du suivi de régates, après les combinaisons des groupes de contextes, cinq chaînes de contextes sont formées parmi les onze configurations initiales. Dans la section suivante, nous étendons le processus de conception d'un système interactif pour qu'il produise une réponse adaptée à chaque contexte des chaînes d'équivalence.

4. Modèle pour la conception

La caractérisation de l'espace géographique du système permet de décrire les conditions rencontrées lors de l'exécution. Les intersections des régions d'intérêt déterminent l'ensemble des chaînes de contextes qu'il faut prendre en compte lors de la conception du système. Pour intégrer ces situations, le cadre de conception d'un système d'information interactif est étendu et intègre le contexte d'exécution comme entrée du processus de modélisation.

À notre connaissance, la conception d'un système interactif est principalement centrée sur l'ingénierie de l'interaction homme-machine du tiers client et l'impact d'un contexte dynamique reste cantonné aux utilisateurs et à leur environnement proche. Pour ces systèmes, percevoir le contexte présente un intérêt si l'exécution doit être modifiée pour s'adapter à une situation nouvelle rencontrée par les utilisateurs. Dans le cadre de conception unifié (figure 1), les contraintes contextuelles d'usage sont répercutées lors des dernières étapes de la modélisation, lorsque les niveaux d'interactions avec les utilisateurs sont spécifiés. Les composants métiers et données ne sont pas contraints par l'environnement : à tout moment, ils tiennent leur rôle dans le système. Pour le déploiement d'un système d'information distribué dans un environnement dynamique, il est nécessaire de considérer en plus les variations

du contexte d'exécution au niveau de chaque composant. Le système déployé est fortement contraint par sa situation dans l'environnement, dans lequel l'évolution de la localisation spatiale et temporelle de ses composants joue un rôle prépondérant. Le cadre de conception étendu intègre cette versatilité de l'environnement du système, et caractérise l'influence de l'environnement d'exécution du système sur le déroulement du scénario nominal (figure 9).

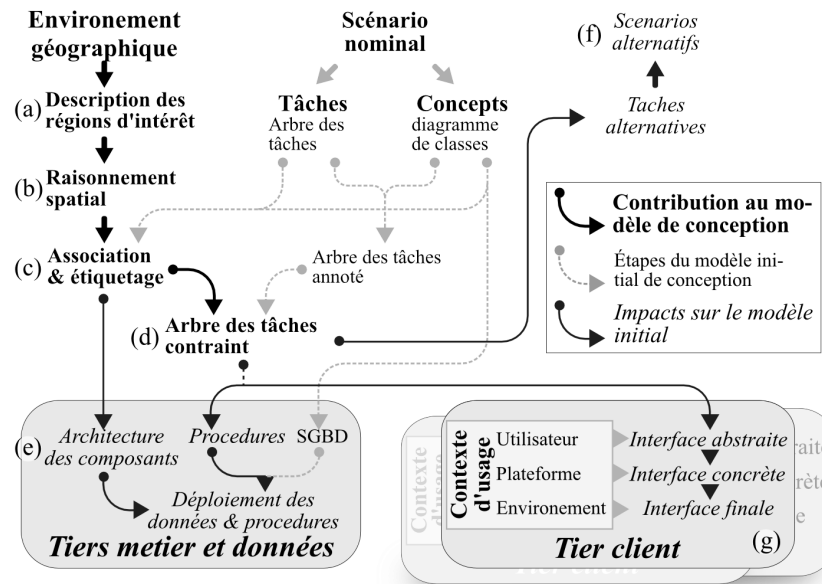


Figure 9. *Modèle de conception étendu*

L'approche géographique modélise la dynamique des composants d'un système mobile par un ensemble de régions (figure 9(a)), où l'étude des contraintes et des équivalences entre configurations topologiques déterminent des groupes de contextes (figure 9(b)). Ces étapes, décrites en section 3.1, caractérisent l'ensemble des situations d'exécution que le système peut rencontrer. Elles sont mises en relation avec les tâches de l'utilisateur et les informations manipulées (figure 9(c)). Le déroulement du scénario nominal va dépendre de la réalisation de ces situations à l'exécution. L'association entre les tâches et les chaînes de contextes d'exécution rapporte ces contraintes au sein du processus de conception (figure 9(d)).

Certaines chaînes de contextes ne contraignent aucune partie du scénario. Il existe dans ce cas des situations d'exécution possibles qui n'ont pas été décrites par le groupe de réflexion à l'origine du scénario nominal. Afin que le système réponde de façon appropriée en toute circonstance, il faut prévoir un comportement adapté à ces situations. Des nouvelles tâches sont imaginées au sein de scénarios alternatifs (figure 9(f)). Elles

seront effectuées à l'initiative du système ou de l'utilisateur lorsque les conditions d'environnement ne permettent pas l'exécution des tâches du scénario nominal.

L'architecture des composants sert de support au système de régions mis en place. En particulier, celle-ci doit permettre la détection des variations dans les intersections des régions d'intérêt pour que les changements de contextes soient répercutés au niveau du système (figure 9(e)). En parallèle, les tâches contraintes sont à l'origine de la partie cliente. La modélisation du tiers client décompose en action d'interaction et de dialogue à la fois les tâches du scénario nominal, mais aussi les tâches alternatives. Comme l'ensemble des situations d'exécution est prise en compte, l'interaction homme-machine est *cohérente* et *robuste*² face aux changements de contexte du système (figure 9(g)).

Dans la suite de cette section, un exemple de conception est présenté en partant du scénario nominal imaginé dans le cas du système de suivi de régate. Nous détaillons particulièrement les étapes d'intégration de la description du contexte d'exécution (figure 9(c) et (d)), avant de présenter des tâches alternatives issues de la combinaison du modèle conceptuel au modèle d'environnement.

4.1. Scénario nominal et décomposition de la tâche

Afin de concevoir le système de suivi, des utilisateurs potentiels et les organisateurs de la régate joignent l'équipe de développement. Ils expriment ensemble les besoins des utilisateurs, corrélés aux impératifs des organisateurs et rédigent un scénario nominal (tableau 1).

*Le système de suivi de régate s'exécute sur le terminal mobile de l'utilisateur et lui permet de **suivre la régate** en temps-réel. Le terminal fournit des outils de manipulation et propose une carte de la zone de compétition où évoluent les bateaux de course. L'utilisateur peut **définir sa zone cartographique d'intérêt** en fonction de **l'intérêt qu'il accorde à certains bateaux** en particulier. Il peut également choisir de suivre les **zones d'intérêt des autres utilisateurs**. Si un utilisateur manifeste son **intérêt pour un bateau spécifique**, les informations de positionnement telles que la (localisation, la vitesse et le cap) sont retournées. Elles sont accompagnées de méta-informations : nom, équipage, année et images du voilier. Lorsque l'utilisateur est proche de la zone de course, il peut **prendre des images du voilier et les partager** avec les autres utilisateurs.*

Tableau 1. Extraits du scénario nominal

Les informations du domaine repérées par les concepteurs à la lecture du scénario (soulignées dans le tableau 1) sont représentées dans un diagramme de classe qui

2. La cohérence et la robustesse sont prises au sens des critères d'évaluation de l'interaction homme-machine (Bastien *et al.*, 1995).

fournit un premier patron pour la conception des tiers métier et données. Par exemple, la notion de bateau est définie par les attributs de « nom », de « type » et de « année » au sein de la classe « Bateau ». Les informations dynamiques comme la vitesse, le cap ou la position d'un bateau sont définis par les attributs de « Vitesse », de « Cap » et de « Localisation » au sein de la classe « DonnéesTempsRéel » associée aux objets de type « Bateau ».

Les tâches décrites dans le scénario (signifiées en gras dans le tableau 1) sont décomposées en une série d'actions de l'utilisateur et peuvent être organisées sous forme d'arbre dont les feuilles représentent les séquences de sous-tâches à effectuer pour accomplir la tâche parente. Nous utiliserons la notation CTT (*Concur Task Tree*) qui propose de différencier les tâches abstraites, les tâches d'interaction et les tâches système et qui les articule par des opérateurs d'ordonnancement (Mori *et al.*, 2002) (figure 10). Les tâches d'interaction et les tâches système sont implémentées par des procédures et des méthodes. Par exemple, la tâche principale « suivre la régate » comprend la sous-tâche « définir la zone cartographique d'intérêt », elle-même effectuée soit par « s'intéresser à certains bateaux » ou par « suivre l'intérêt d'un autre utilisateur ».

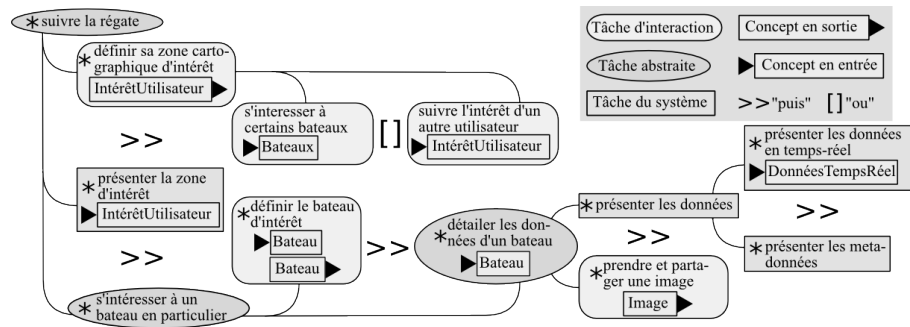


Figure 10. Arbre des tâches annoté du système de suivi de régate

Une fois l'arbre construit, ses nœuds sont annotés avec les informations manipulées en entrée et en sortie à chaque étape :

- $\mathcal{F}out(Task_a)$ retourne l'ensemble des informations $\{Info_b, \dots, Info_n\}$ modifiées en sortie par la tâche $Task_a$;
- $\mathcal{F}in(Task_a)$ retourne l'ensemble des informations $\{Info_b, \dots, Info_n\}$ utilisées en entrée par la tâche $Task_a$.

La tâche d'interaction itérative « définir la zone cartographique d'intérêt » complète par exemple en sortie l'information « IntérêtUtilisateur » à partir d'une entrée constituée soit par un ensemble de bateaux si l'utilisateur effectue la sous-tâche « s'in-

téresser à certains bateaux », soit par la zone d'intérêt d'un autre utilisateur si la sous-tâche « suivre l'intérêt d'un autre utilisateur » est effectuée. Plus formellement :

$$\mathcal{F}out(\text{« définir la zone cartographique d'intérêt »}) = \{\text{« IntérêtUtilisateur »}\}$$

$$\mathcal{F}in(\text{« définir la zone cartographique d'intérêt »}) = \emptyset$$

$$\mathcal{F}out(\text{« s'intéresser à certains bateaux »}) = \{\text{« Bateaux »}\}$$

$$\mathcal{F}out(\text{« suivre l'intérêt d'un autre utilisateur »}) = \{\text{« IntérêtUtilisateur »}\}$$

4.2. Corrélation entre l'environnement et les tâches de l'utilisateur

Le contexte d'exécution du système varie au cours du temps. Ces changements ont été caractérisés par une approche géographique pour la modélisation des composants du système en fonction de régions d'intérêt dynamiques (figure 9(a)). Corréler cette approche géographique de l'environnement avec la modélisation des tâches et des informations d'entrée-sortie permet de définir des prédicats localisés comme « la tâche $Task_a$ est réalisable dans la région P_y » ou « la donnée $Info_b$ est disponible dans la région D_y , et provient de la région S_x » (figure 9(b)).

4.2.1. Étiquetage des tâches et des informations manipulées

Afin d'identifier les régions qui permettent l'exécution d'une tâche donnée, et vérifier le moment et le lieu où elles pourront être effectuées par l'utilisateur, les informations et les tâches sont étiquetées par leurs régions de diffusion. En considérant qu'une tâche $Task_a$ est implémentée par des procédures du composant serveur de traitement C_{P_x} , alors la région d'intérêt P_x supportée par ce composant constitue une étiquette de la tâche $Task_a$. Similairement, une information $Info_b$ est étiquetée par la région D_x si le serveur de données C_{D_x} enregistre les informations définissant $Info_b$. Plus formellement, les régions constituent un ensemble d'étiquettes sur les tâches et les informations :

- $\mathcal{F}tag(Task_a)$ retourne l'ensemble des régions $\{P_1, \dots, P_n\}$ qui étiquettent la tâche $Task_a$;
- $\mathcal{F}tag(Info_b)$ retourne l'ensemble des régions $\{D_1, \dots, D_n\}$ qui étiquettent l'information $Info_b$.

Dans le cas de la décomposition du système de suivi, où seules une région de traitement et une région de diffusion de données sont mises en place, toutes les procédures implémentant les tâches de l'utilisateur sont disponibles dans la région P_1 et toutes les informations du domaine sont enregistrées sur le serveur support de la région des données D_1 . Quelle que soit la tâche $Task_a$ choisie dans l'arbre, $\mathcal{F}tag(Task_a)$ retourne $\{P_1\}$, et quelle que soit l'information $Info_b$, $\mathcal{F}tag(Info_b)$ retourne $\{D_1\}$.

Pour s'exécuter correctement, une tâche a besoin d'accéder aux informations qui lui sont associées en entrée et en sortie. En pratique, les procédures doivent pouvoir recevoir les informations nécessaires depuis un ou plusieurs serveurs de données. Au

niveau des régions, cela se traduit par des intersections entre la région T_x où est effectuée une tâche, et les régions de données $\{D_1, \dots, D_n\}$ des informations en entrée et sortie de la tâche. Plus formellement, $\mathcal{F}exec(Task_a, P_x)$ retourne *vrai* si une tâche $Task_a$ peut s'exécuter dans une région de traitement P_x , et *faux* sinon :

$$\mathcal{F}exec(Task_a, P_x) = \begin{cases} \text{vrai ssi} & \left(\forall Info_b \in (\mathcal{F}out(Task_a) \cup \mathcal{F}in(Task_a)), \exists D_x \right. \\ & \left. \in \mathcal{F}tag(Info_b), P_x \cap D_x \neq \emptyset \right) \wedge P_x \in \mathcal{F}tag(Task_a) \\ \text{faux sinon} \end{cases}$$

Dans le cas du système de suivi, comme $P_1 \cap D_1 \neq \emptyset$, toutes les informations sont accessibles aux procédures de traitement et l'ensemble de la tâche est toujours réalisable dans P_1 . Quelle que soit la tâche $Task_a$, $\mathcal{F}exec(Task_a, P_1) = \text{vrai}$.

Afin d'effectuer une tâche $Task_a$, un utilisateur dont l'espace d'interaction est U_x doit accéder aux procédures qui implémentent cette tâche, et celle-ci doit être exécutable. Ces conditions sont réunies quand $((U_x \cap P_x \neq \emptyset) \wedge \mathcal{F}exec(Task_a, P_x)) = \text{vrai}$. Dans le système de suivi où $\mathcal{F}exec(Task_a, P_1)$ est vrai quel que soit la tâche, il suffit à l'utilisateur de se situer dans la région des traitements pour pouvoir réaliser l'ensemble des tâches du scénario nominal.

4.2.2. Arbre des tâches contraint

En considérant l'arbre des tâches initial (figure 10) et l'association géographique des tâches et des informations, l'accès aux fonctionnalités est garanti dans une région limitée de l'espace de mobilité ζ_{U_1} de l'utilisateur. Celle-ci correspond à l'intersection entre la région des procédures P_1 et l'espace de mobilité ζ_{U_1} (figure 11(a)).

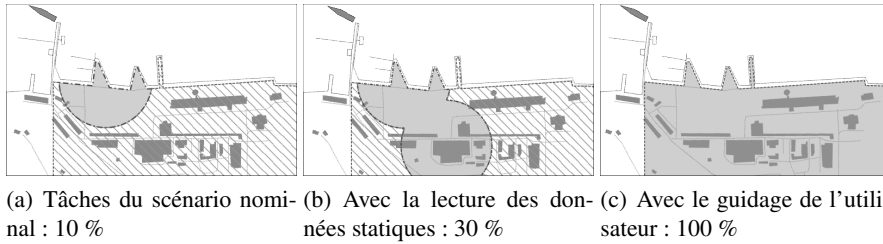


Figure 11. Proportion de l'espace ζ_{U_1} où l'utilisateur accède à des tâches du système

Dans ce cas, seuls trois groupes de contextes parmi cinq caractérisent l'accès aux procédures du système par un utilisateur dans la région des traitements, lorsque $U_1 \cap P_1 \neq \emptyset$ (figure 8(c), 8(d) et 8(e)). Afin de garantir que le système réponde aux attentes des utilisateurs dans l'ensemble de son espace de mobilité, les concepteurs imaginent un comportement du système pour les groupes de contextes restants (figure 8(a) et 8(b)), en enrichissant la tâche initiale par des cas alternatifs :

– lorsque l'utilisateur est situé dans la zone de diffusion des données sans être dans la zone de traitement, il accède à l'ensemble des données enregistrées sur C_{D_1} (figure 8(c)). Le système propose une interrogation directe des données et présente les métadonnées sur les bateaux. La tâche « s'intéresser à un bateau en particulier » reste réalisable. L'emprise spatiale de l'espace de l'utilisateur où le système est accessible couvre $\zeta_{U_1} \cap (P_1 \cup D_1)$ (figure 11(b)) ;

– lorsque l'utilisateur est situé en dehors des régions actives P_1 ou D_1 (figure 8(a)), le système lui indique sur une carte simplifiée l'endroit où se situent les régions exécutant les tâches du scénario nominal. Comme toutes les chaînes de contexte ont été prises en compte, la couverture du système s'étend sur tout l'espace de mobilité de l'utilisateur ζ_{U_1} (figure 11(c)). Cette assistance du système pour la recherche d'une meilleure zone d'utilisation, prévue à la conception, est intégrée à l'arbre des tâches. Une tâche abstraite « trouver le service », alternative à la tâche « suivre la régate », est effectuée à l'initiative du système quant l'utilisateur est hors des régions actives. Cette nouvelle tâche permet de compléter les cas d'usages en imaginant un scénario complémentaire³ (tableau 2) ;

– la position de l'utilisateur par rapport à S_1 précise la notion de proximité à la zone de course, exprimée dans le scénario nominal (« Lorsque l'utilisateur est proche de la zone de course »). Elle est perçue par le système comme une topologie particulière de l'environnement d'exécution, où l'espace de l'utilisateur est en intersection avec la région de provenance des données. Dans ces conditions d'exécution, l'utilisateur peut toujours effectuer chaque tâche du scénario, et le système lui permet de prendre et de partager des photos de la course.

*Un utilisateur marche le long du rivage et souhaite recevoir les informations de course sur son terminal mobile. Lorsqu'il accède au système d'information, il est averti qu'il doit **trouver le service** aux régions P_1 or D_1 afin d'obtenir un système complètement fonctionnel. Les instructions d'orientation vers ces régions lui sont **présentées sur un plan du système**.*

Tableau 2. Scénario alternatif

Ces contraintes d'environnement sont intégrées à l'arbre des tâches, sous la forme de préconditions nécessaires à l'exécution d'une tâche et à l'ensemble de ses sous-tâches (figure 12). Par exemple, les tâches d'enregistrement et de rendu du centre d'intérêt de l'utilisateur peuvent être effectuées quand le système se trouve dans l'un des états des chaînes de contexte où la région de traitement intersecte la région de l'utilisateur (figure 12(b)). La tâche « s'intéresser à un bateau en particulier » peut aussi être effectuée lorsque l'utilisateur intersecte la région de diffusion de données, mais seule la sous-tâche « présenter les métadonnées » de la branche « détailler les données d'un bateau » est effectuable (figure 12(d)).

3. Son énonciation, optionnelle, permet toutefois de prévoir simplement tout les cas d'usages pour des tests du prototype.

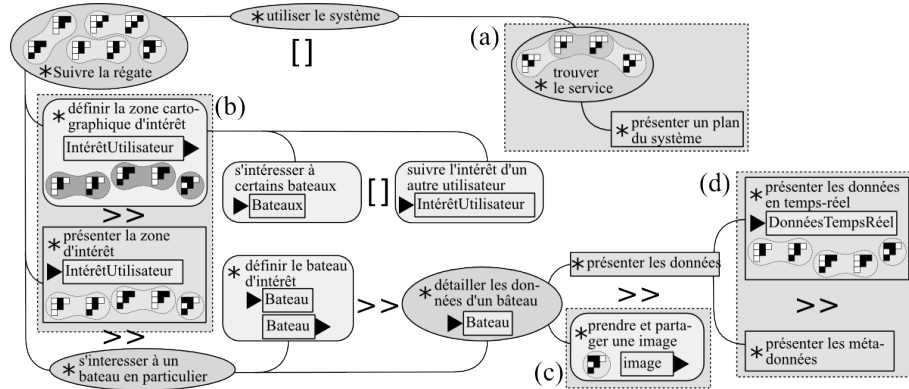


Figure 12. Arbre des tâches contraint

Pour pouvoir partager une photo et effectuer la tâche « prendre et partager une image », l'espace utilisateur doit être dans la zone de course, et partager une intersection avec la zone de traitement (figure 12(c)). Enfin, les tâches du scénario alternatif ajoutent une nouvelle branche à l'arbre effectuée lorsque l'utilisateur est seul (figure 12(a)). Quel que soit le contexte pris parmi les onze possibles, il existe toujours un chemin dans l'arbre menant à une tâche non abstraite. Le système propose bien une réponse à l'utilisateur, indépendamment de la situation encourue.

5. Prototypage

L'arbre des tâches contraint par l'environnement, et augmenté des cas alternatifs, est le point d'entrée de l'implémentation de la partie cliente. Comme chaque état du système est caractérisé dans l'arbre, la présentation des fonctionnalités du cas nominal et des cas alternatifs est unifiée dans une interface homme-machine opérationnelle à tout moment lors de l'exécution du système.

Les contextes d'exécution qui annotent les nœuds de l'arbre des tâches servent d'indicateurs de la façon dont les procédures doivent être distribuées parmi les composants du système. Par exemple, la tâche « présenter un plan du système » s'exécute en intégralité sur la seule plateforme disponible à ce moment : l'assistant personnel de l'utilisateur (composant C_{U_1}). La tâche « détailler les données d'un bateau » s'exécute indifféremment sur les composants C_{P_1} et C_{D_1} supports des régions P_1 et D_1 . Par contre, sa sous-tâche « présenter les données en temps-réel » est implémentée uniquement sur le composant C_{P_1} .

Dans les maquettes de la figure 13, les variations des interfaces de la partie cliente sont illustrées pendant un exemple de parcours d'utilisateur au sein d'un environnement en évolution. Dans cet exemple, à un instant t donné, l'état du système est mis

en correspondance avec l'interface utilisateur permettant d'effectuer les tâches disponibles dans cet état.

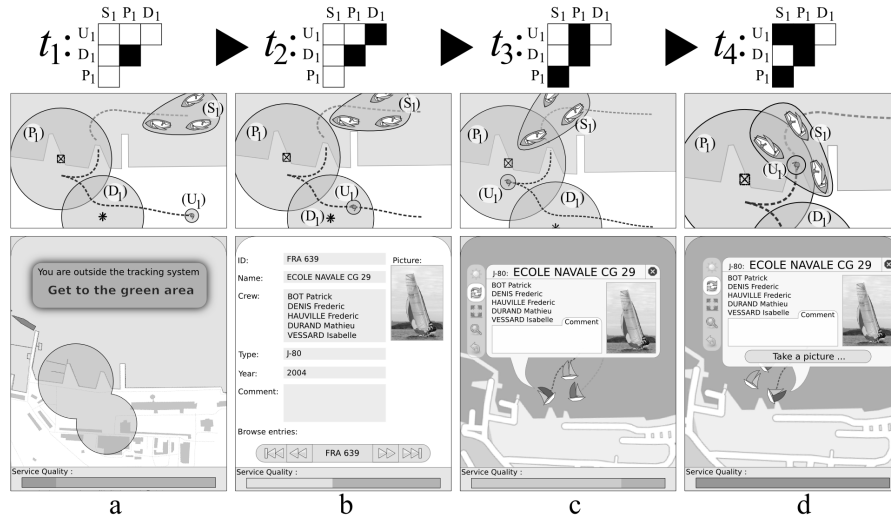


Figure 13. Comportement du système modifié selon le contexte d'exécution rencontré

À l'instant t_1 , l'utilisateur est seul, et le système présente une carte des zones où le système est disponible (tâche « présenter un plan du système »). À t_2 , l'utilisateur s'est rendu dans la zone de diffusion de données. Ce dernier accède à une lecture séquentielle des données statiques des bateaux (tâches entre « détailler les données d'un bateau » et « présenter les métadonnées »). À t_3 , l'utilisateur est arrivé dans la zone des traitements. L'information géographique est rendue en temps réel sur une carte (tâches de la branche « suivre la régate », hormis la tâche « prendre et partager une image »). L'accès aux détails des bateaux n'est plus séquentiel mais direct (e.g., clic sur l'icône d'un bateau). Enfin, à t_4 , l'utilisateur est entré dans la zone de course. Le système considère qu'il est à proximité de la zone de provenance des données et lui permet de prendre et de partager une photo du bateau sélectionné (tâche « prendre et partager une image »). À chaque étape de l'utilisateur, une barre de progression en bas de l'interface indique à l'utilisateur quel est le niveau fonctionnel du système. Elle représente un indicateur de *qualité de service*, mesurant la proportion de l'arbre des tâches réalisable dans le contexte d'exécution courant. Entre t_1 et t_4 , l'utilisateur passe progressivement d'une interface présentant peu de fonctionnalités, à un accès complet au système.

En plus de cette approche conceptuelle, le prototype profite de l'architecture des régions et de la détection des différents contextes pour adapter l'interaction homme-machine au moment de l'exécution (au *niveau des données* du contexte d'usage). En particulier, des utilisateurs associés dans un même contexte d'exécution effectuent les mêmes tâches. Ces utilisateurs contribuent à la définition d'un profil commun qui

adapte l'interaction avec les données manipulées. Un algorithme de partage et de fusion de profils de groupe a été développé et validé par simulation. Il montre qu'avec des utilisateurs en mouvement qui traversent des séquences de contextes d'exécution, les profils des groupes se propagent dans tout l'espace du système. Le niveau de personnalisation des procédures et des interfaces s'améliore avec la quantité de mouvements et le nombre d'utilisateurs (Petit *et al.*, 2007; Petit *et al.*, 2008).

6. Conclusion

Le cadre de conception présenté dans cet article intègre une caractérisation géographique du contexte d'exécution d'un système comme point d'entrée d'une démarche de modélisation d'un système d'information mobile et distribué. Notre approche, centrée sur une spatialisation de régions d'intérêt décrit l'ensemble des contextes système et prévoit pour chacun d'eux un comportement approprié. L'empreinte spatiale des régions est déterminée par les composants du système et par l'origine géographique des données manipulées. Ces régions sont combinées en un ensemble de configurations qui représentent autant de situations d'exécutions possibles au *niveau du système*. Ces situations influencent les procédures disponibles pour l'utilisateur et modifient sa tâche en adaptant les capacités fonctionnelles du système. Les tâches de l'utilisateur, décrites dans un scénario nominal, sont annotées suivant les contextes dans lesquels elles peuvent être effectuées. Afin de caractériser l'ensemble des contextes d'exécution, le scénario nominal est enrichi par de nouvelles tâches. Ces dernières sont associées aux contextes pour lesquels initialement aucune tâche de l'utilisateur n'était exprimée.

Dans le cas d'étude du suivi de régate, développé sur une architecture dynamique à quatre régions d'intérêt, les contextes d'exécution sont groupés en cinq chaînes d'équivalence et corrélés avec la description de la tâche de l'utilisateur. Alors que le fonctionnement initial du système se limiterait à une portion réduite de l'espace de mobilité de l'utilisateur, l'association de tâches à chaque chaîne de contextes garantit l'accès au système depuis l'ensemble de l'espace utilisateur. Le niveau fonctionnel offert à l'utilisateur est adapté en fonction des informations et les procédures à disposition.

Les travaux en cours concernent la formalisation des propriétés d'équivalences entre configuration de régions, et la généralisation des processus de groupement à un nombre arbitraire de régions. Du point de vue du contexte d'usage, la réutilisation des configurations de régions pour la personnalisation du contenu est en cours d'implémentation dans le cas d'un système de visite virtuelle de campus⁴. Enfin, l'analyse géographique proposée peut intégrer une méthodologie conceptuelle qui guiderait l'ingénierie et le développement de systèmes distribués et mobiles.

4. Ces travaux sont menés en partenariat avec le laboratoire d'informatique de l'University College Dublin.

7. Bibliographie

- Bastide R., Navarre D., Palanque P., « A tool-supported design framework for safety critical interactive systems », *Interacting with Computers*, vol. 15, n° 3, p. 309-328, jun, 2003.
- Bastien J., Scapin D., « Evaluating a user interface with ergonomic criteria », *International Journal of Human-Computer Interaction*, vol. 7, n° 2, p. 105-121, 1995.
- Boehm B., « A spiral model of software development and enhancement », *SIGSOFT Softw. Eng. Notes*, vol. 11, n° 4, p. 14-24, 1986.
- Calvary G., Plasticité des Interfaces Homme-Machine, Thèse d'habilitation à Diriger des Recherches, Université Joseph Fourier, November, 2007.
- Calvary G., Coutaz J., Thevenin D., Limbourg Q., Bouillon L., Vanderdonckt J., « A unifying reference framework for multi-target user interfaces », *Interacting with Computers*, vol. 15, n° 3, p. 289-308, 2003.
- Claramunt C., Devoegele T., Fournier S., Noyon V., Petit M., Ray C., « Maritime GIS : From Monitoring to Simulation Systems », in V. Popovitch, M. Schrenk, K. Korolenko (eds), *Proceedings of the 3rd International Workshop on Information Fusion and Geographical Information System*, LNG&C, Springer Verlag, p. 34-44, 27 May, 2007.
- Dey A., Abowd G., Salber D., « A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications », *Human-Computer Interaction*, vol. 16, n° 2, p. 97-166, 2001.
- Eckerson W., « Three Tier Client/Server Architecture : Achieving Scalability, Performance, and Efficiency in Client Server Applications », *Open Information Systems*, vol. 10, n° 1, p. 3-22, January, 1995.
- Edmonds E., « The Pragmatic Roots of Context », *Proceedings of the second International and Interdisciplinary Conference on Modelling and Using Context*, vol. 1999, p. 119-132, 1999.
- Eisenstein J., Vanderdonckt J., Puerta A., « Applying Model-Based Techniques to the Development of UIs for Mobile Computers », *Proceedings of the 6th international conference on Intelligent user interfaces*, ACM Press, p. 69-76, 2001.
- Longley P. A., Goodchild M. F., Maguire D. J., Rhind D. W., *Geographic Information Systems and Science*, 2nd edn, Wiley, chapter Distributed GIS, 2005.
- Luyten K., Laerhoven T. V., Coninx K., Reeth F. V., « Runtime transformations for modal independent user interface migration », *Interacting with Computers*, vol. 15, n° 3, p. 329-347, jun, 2003.
- Mac Aoidh E., Bertolotto M., « Improving Spatial Data Usability By Capturing User Interactions », in S. I. Fabrikant, M. Wachowicz (eds), *The European Information Society*, LNGC, Springer Verlag, p. 389-403, 2007.
- Mori G., Paterno F., Santoro C., « CTTE : support for developing and analyzing task models for interactive system design », *IEEE Transactions on Software Engineering*, vol. 28, n° 8, p. 797-813, aug, 2002.
- Mori G., Paterno F., Santoro C., « Design and development of multidevice user interfaces through multiple logical descriptions », *IEEE Transactions on Software Engineering*, vol. 30, n° 8, p. 507-520, aug, 2004.
- Myers B., « User interface software tools », *ACM Transactions on Computer-Human Interaction*, vol. 2, n° 1, p. 64-103, March, 1995.

- Paterno F., Santoro C., « A unified method for designing interactive systems adaptable to mobile and stationary platforms », *Interacting with Computers*, vol. 15, n° 3, p. 349-366, jun, 2003.
- Petit M., Ray C., Claramunt C., « A contextual approach for the development of GIS : Application to maritime navigation », in J. Carswell, T. Tekuza (eds), *Proceedings of the 6th International Symposium on Web and Wireless Geographical Information Systems*, n° 4295 in LNCS, Springer Verlag, p. 158-169, 4 December, 2006.
- Petit M., Ray C., Claramunt C., « A user context approach for adaptive and distributed GIS », in M. Wachowicz, S. Fabrikant (eds), *Proceedings of the 10th International Conference on Geographic Information Science : AGILE'07*, LNG&C, Springer Verlag, p. 121-133, 10 May, 2007.
- Petit M., Ray C., Claramunt C., « An adaptive interaction architecture for collaborative GIS », *Cartographic and Geographical Information Sciences. Special issue on Modeling and Visualization for Spatial Decision Support*, vol. 35, n° 2, p. 91-102, Apr, 2008.
- Reichenbacher T., « Adaptive methods for mobile cartography », *Proceedings of the 21st International Cartographic Conference*, p. 1311-1322, 2003.
- Rey G., Coutaz J., « Contextor : capture and dynamic distribution of contextual information », *UbiMob '04 : Proceedings of the 1st French-speaking conference on Mobility and ubiquity computing*, ACM, p. 131-138, 2004.
- Satyanarayanan M., « Fundamental Challenges in Mobile Computing », *Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing*, p. 1-7, 1996.
- Schmidt A., Beigl M., Gellersen H.-W., « There is more to context than location », *Computers and Graphics*, vol. 23, n° 6, p. 893-901, dec, 1999.
- Viana W., Bringel Filho J., Jérôme Gensel J., Villanova Oliver M., Martin H., « PhotoMap - Automatic Spatiotemporal Annotation for Mobile Photos », in J. M. Ware, G. E. Taylor (eds), *Proceedings of the 7th International Symposium on Web and Wireless Geographical Information Systems*, LNCS, Springer Verlag, p. 187-201, nov, 2007.