

Conception de systèmes d'information spatio-temporelle adaptatifs avec ASTIS

Bogdan Moisuc, Jérôme Gensel, Hervé Martin

Laboratoire LSR-IMAG

Equipe SIGMA

681, rue de la Passerelle,

BP. 72, 38402 Saint Martin d'Hères Cedex, France

{Bogdan.Moisuc, Jerome.Gensel, Herve.Martin}@imag.fr

Résumé. Les avancées technologiques récentes du Web et du sans fil, conjuguées au succès des applications spatialisées grand public, sont à l'origine d'un accès accru aux systèmes d'information spatio-temporelle (SIST) par une grande diversité d'utilisateurs, munis des dispositifs d'accès et dans des contextes d'utilisation variés. Adapter ces systèmes à l'utilisateur devient donc une nécessité, un gage d'utilisabilité et de pérennité. Cet article présente une approche générique pour la conception et la génération de systèmes d'information spatio-temporelle adaptés à l'utilisateur, appelé ASTIS. ASTIS offre des modalités générales de mise en oeuvre de l'adaptation à l'utilisateur, visant tant le contenu que la présentation des applications. Elle permet aux concepteurs d'intégrer ces modalités d'adaptation dans des applications traitant des données spatio-temporelles. Afin de définir les besoins et types d'adaptation propres à leur application, il suffit aux concepteurs de créer des modèles conceptuels, par spécialisation et instanciation des modèles offerts par notre architecture.

1 Introduction

Dans le domaine des systèmes d'informations géographique et, plus généralement, des systèmes d'informations spatialisées, certaines recherches (Timpf, 2001, Laurini et Servigne, 2007), montrent depuis une dizaine d'années que ces applications doivent prendre en considération une variété plus large d'influences et de limitations que celles utilisées dans la cartographie conventionnelle (de nature matérielle, mais aussi de nature cognitive et sémantique), liées aux utilisateurs, aux dispositifs d'accès, à l'environnement, et de s'adapter afin d'être utilisables par leur différents utilisateurs finaux.

Ces préoccupations se sont multipliées avec l'émergence de l'informatique mobile et ubiquitaire. L'augmentation de l'autonomie des dispositifs mobiles, de leur taille mémoire et de leur puissance de calcul a donné la possibilité à la communauté de chercheurs en géomatique d'explorer de nouvelles applications des systèmes d'information géographique

(Anegg, 2002, Zipf, 2002). De nouvelles perspectives s'ouvrent pour les utilisateurs qui peuvent accéder à des informations spatialisées n'importe quand, n'importe où, à travers des dispositifs d'accès très variés en termes de puissance de calcul et capacité d'affichage (par exemple les services localisés et les systèmes de navigation). Cet accès accru et universel aux applications spatiales accentue les problèmes d'adaptation, liés tant aux limitations intrinsèques des dispositifs mobiles, qu'aux problèmes cognitifs soulevés par l'interaction entre l'utilisateur et ces types de dispositif dans un environnement mobile.

Initialement, les recherches se sont focalisées sur le contournement des limitations techniques des dispositifs mobiles (ou DM), comme les PDA (*Personal Digital Assistant*) et les *smartphones* (téléphones portables de dernière génération dotées d'écrans tactiles). Parmi les objectifs visés nous pouvons citer l'optimisation de l'utilisation de la bande passante dans des réseaux à bas débit (Follin et al., 2005), la généralisation du contenu pour réaliser l'adaptation à la capacité de calcul et à la capacité mémoire des DM (Lee et al., 2003), la généralisation dynamique pour adapter la présentation de cartes sur les DM à affichage réduit (Sester et Brenner, 2004).

Dans des recherches plus récentes, l'attention s'est portée vers les aspects cognitifs de l'adaptation à l'utilisateur (personnalisation) et à son contexte (sensibilité au contexte). Les travaux sur la personnalisation visent à éviter les phénomènes de fatigue et désorientation (Conklin, 1987). En transformant l'application de manière à prendre en compte le profil de l'utilisateur, il devrait avoir le sentiment que le système a été conçu spécialement pour lui. Parmi les caractéristiques des utilisateurs considérées pour l'adaptation, on trouve le plus couramment les préférences et intérêts de l'utilisateur (Malaka et Zipf, 2000), sa nationalité (Sarjakoski et al., 2002) ou son âge (Nivala et Sarjakoski, 2004).

D'autres recherches mettent en évidence l'influence que peut avoir le contexte de l'utilisateur sur l'adaptation (Zipf, 2005, Reichenbacher, 2001) en soulignant que les SIST doivent être capables de présenter à l'utilisateur une information pertinente, au moment opportun et au bon endroit, en fonction de son environnement. Plus précisément, la notion de contexte de l'utilisateur définit l'ensemble d'entités qui peuvent avoir une influence significative sur l'interaction entre l'utilisateur et l'application. Parmi les caractéristiques du contexte le plus souvent prises en compte pour l'adaptation, on peut citer la localisation (Anegg, 2002, Zipf, 2002), l'activité en cours de réalisation par l'utilisateur (Zipf, 2002, Petit et al., 2006), la saison (Nivala et Sarjakoski, 2004) ou encore la direction de mouvement de l'utilisateur (Roth, 2002).

Ces travaux présentent toutefois certains inconvénients. Ainsi, les applications existantes utilisent des mécanismes d'adaptation assez simplistes, qui ne tiennent pas compte des aspects plus complexes de la cartographie (Reichenbacher, 2001) comme, par exemple, les représentations multiples. De même, la dimension temporelle n'est généralement pas prise en compte, tant au niveau de la gestion de données qu'au niveau de l'adaptation (Schwinger et al., 2005). Enfin, les approches existantes n'offrent pas de support pour la conception de SIST adaptatifs. La mise en œuvre des mécanismes d'adaptation se confond avec la logique interne de l'application, ce qui rend ces approches très difficilement réutilisables et extensibles.

En constatant ce manque d'outils et approches génériques pour l'adaptation dans les SIST, notre objectif est d'offrir aux concepteurs de SIST des outils de conception et des logiciels afin de leur faciliter les démarches de conception et de développement de SIST adaptables. Cet article présente un *framework*, appelé ASTIS, dédié à la conception et la

réalisation de systèmes d'information spatio-temporelle adaptatifs et interactifs. ASTIS adopte une approche de conception et de mise en oeuvre d'applications basée sur des modèles. Nous avons préalablement proposé une architecture modulaire appelée GenGHIS (Moisuc et al., 2005), séparant contenu et présentation et permettant aux concepteurs de SIST de générer des applications à partir de modèles de données conçus par eux. GenGHIS est composé de deux modules principaux : l'un assure la gestion du contenu, l'autre la gestion de la présentation, en s'appuyant sur un système de représentation de connaissances par objets spatio-temporel appelé AROM-ST (Moisuc et al., 2004). ASTIS étend GenGHIS avec des mécanismes d'adaptation de contenu (un système de création de points de vue sur les données) et d'adaptation de la présentation (un système de styles graphiques et composants de présentation). Avec ASTIS, après avoir conçu le modèle de données et le modèle de présentation de base pour l'application, les concepteurs peuvent spécifier différentes adaptations de contenu et de présentation, en fonction des caractéristiques des utilisateurs et du contexte. Au démarrage de l'application, un module d'adaptation est chargé de gérer les utilisateurs, de choisir les adaptations nécessaires et de les mettre en œuvre dynamiquement.

L'article est organisé de la manière suivante : la section 2 présente l'architecture générale du *framework* ASTIS. Les sections 3 et 4 détaillent les mécanismes adaptatifs utilisés au sein d'ASTIS pour réaliser l'adaptation aux points de vue et aux niveaux d'expertise des utilisateurs. La section 5 conclut l'article en présentant les perspectives de développement.

2 L'architecture générale de la plate-forme

Notre approche pour la conception et le développement d'applications spatio-temporelles adaptées est basée sur la génération de code à partir de modèles. L'idée est de permettre aux concepteurs de créer des applications adaptatives en se basant sur une approche déclarative, de conception de modèles orientés objet. L'architecture est formée de composants paramétrables qui opèrent une suite de transformations et de conversions, depuis des formats et sources externes de données (aux formats relationnels les plus répandus : mif/mid, excel, mdb, etc.) jusqu'au format final, affichable à l'écran (documents SVG et XHTML avec une structure particulière). Le *framework* contient également des composants visuels génériques, capables d'afficher de manière interactive les documents structurés résultants du processus de transformation.

L'architecture est composée de trois modules principaux : un module de données, un module de présentation et un module d'adaptation. Le module de données est chargé du stockage, de l'interrogation et de l'analyse des données en conformité avec les modèles spécifiés par les concepteurs. Il permet aux concepteurs de décrire leurs modèles de données et de les instancier. Le module de présentation permet aux concepteurs de créer des modèles de présentation adaptés et de générer des présentations conformes à ces modèles. Il contient aussi les composants visuels nécessaires pour afficher ces présentations. Le module d'adaptation est chargé de la gestion du contexte, des utilisateurs, et de l'application dynamique des mécanismes d'adaptation choisis en fonction de ces cibles.

Le fonctionnement de l'architecture est décrit de manière simplifiée dans la FIG. 1. La partie gauche de la figure illustre la partie statique de l'architecture, c'est-à-dire les composants impliqués dans les processus de conception. La partie droite contient la partie

dynamique de l'architecture, c'est-à-dire les composants chargés de la génération à la volée et de l'affichage de contenus SVG adaptés. Les composants représentés par des rectangles à angles droits sont des composants paramétrisables via des modèles. Les composants représentés sous forme de rectangles à coins arrondis sont les composants chargés des conversions de format.

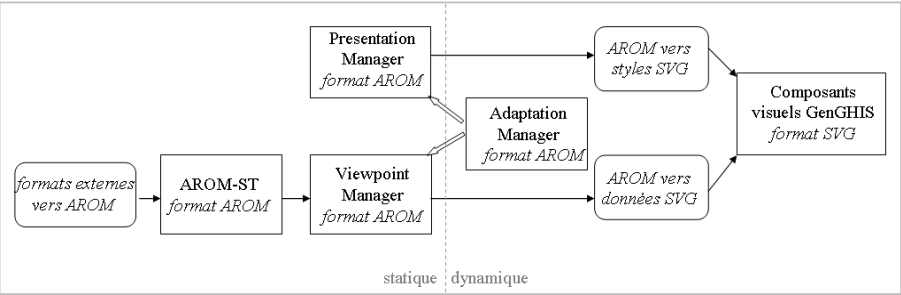


FIG. 1 Schéma du fonctionnement d'ASTIS.

Pour créer une nouvelle application, les concepteurs doivent d'abord spécifier le schéma de données de l'application. Ensuite, ils doivent spécifier différentes transformations de ce schéma afin de l'adapter aux caractéristiques des utilisateurs et du contexte. De la même manière, la troisième étape consiste à spécifier des adaptations de présentation, c'est-à-dire différentes options de style et d'affichage applicables. La dernière étape de conception permet de définir et faire la liaison entre les différentes caractéristiques du contexte de l'utilisateur et les adaptations correspondantes. Cela revient à spécifier un modèle d'adaptation comme celui de la FIG. 2. A présent, il n'existe pas un consensus sur une terminologie de l'adaptation. Nous avons adopté la terminologie de (Reichenbacher, 2003). Ainsi, la cible de l'adaptation (*target*) définit les caractéristiques significatives de l'utilisateur ou de son contexte auxquelles l'application doit s'adapter (par exemple, le niveau d'expertise de l'utilisateur ou sa localisation). Une composante à adapter (*adaptee*) représente une caractéristique du contenu ou de la présentation qui peut être ajustée en adéquation avec le contexte. Le mécanisme d'adaptation (*adapter*) est la fonctionnalité du système qui permet d'ajuster les caractéristiques des composantes à adapter en fonction des caractéristiques connues des cibles.

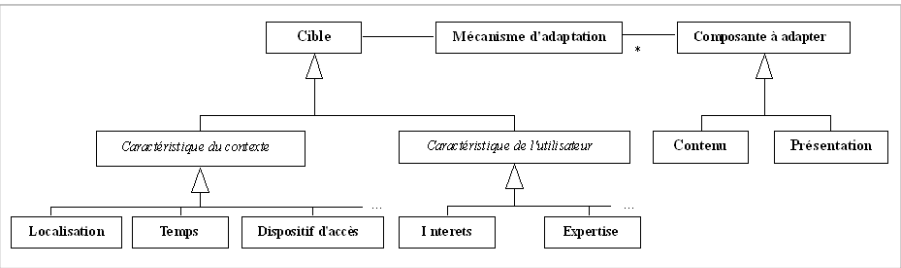


FIG. 2 *Modèle d'adaptation d'ASTIS.*

Afin de finaliser la réalisation d'une nouvelle application les concepteurs doivent procéder à l'acquisition de données, c'est-à-dire peupler avec des instances la base de connaissances AROM-ST de l'application, en utilisant le composant de conversion de formats externes vers AROM-ST. Notre approche de conception s'inscrit donc dans le cadre de l'*adaptativité* (MacEachren, 1998), les concepteurs sont amenés à décrire des adaptations en fonction des caractéristiques de l'utilisateur et de son contexte, qui seront appliquées automatiquement, sans l'intervention des utilisateurs, au moment de l'exécution de l'application. Il est important de noter que, du point de vue technique il est possible de mettre en œuvre de l'*adaptabilité*, c'est à dire de permettre aux utilisateurs de décrire et appliquer eux-mêmes ces adaptations, de manière dynamique, pendant l'exécution de l'application. Nous considérons que, pour que notre approche soit utilisable, il est nécessaire de créer de nouvelles interfaces et des manières simplifiées de contrôler les applications, parce qu'il est raisonnable d'imaginer que les utilisateurs finaux sont moins avisés en ce qui concerne la modélisation de données et de présentations spatio-temporelles. Une des perspectives de notre travail vise la mise en œuvre d'interfaces et modalités de présentation simplifiées pour rendre notre architecture plus utilisable aussi dans une approche basée adaptabilité.

Au moment de l'exécution, le gestionnaire d'adaptation choisit, en fonction des caractéristiques concrètes du contexte, les modalités adéquates d'adaptation à appliquer, décrites par le modèle d'adaptation. Ces modalités sont appliquées au moment de la génération du code SVG de l'application, au niveau de la présentation (structure de la présentation, styles et options d'affichage), et au niveau du contenu (le point de vue sur les données). Les modalités d'application de ces adaptations sont détaillées dans les 2 sections suivantes, et sont illustrées par des exemples d'adaptation visant deux caractéristiques de l'utilisateur : ses intérêts et son niveau d'expertise.

3 Adaptation de contenu

L'adaptation de contenu est réalisée à travers le concept de *point de vue sur les données*. Notre plate-forme utilise pour la modélisation et pour le stockage des données un système de représentation de connaissances par objets, étendu avec des types et opérateurs spatio-temporels (Moisuc et al., 2004). Opérationnalisation d'UML, AROM (Page et al., 2001) modélise la réalité sous forme de *classes* et d'*associations*. Les classes peuvent avoir des propriétés appelées *attributs*. Une association exprime une relation existante entre au moins deux classes, appelées *rôles*. Les instances de classes et associations sont appelées *objets* et, respectivement, *tuples*. AROM dispose d'un langage de modélisation algébrique (LMA) qui permet de formuler des requêtes, d'exprimer des contraintes et de calculer des valeurs d'attributs. Notre plate-forme repose sur les capacités de modélisation d'AROM et du LMA pour la gestion de points de vue sur les données.

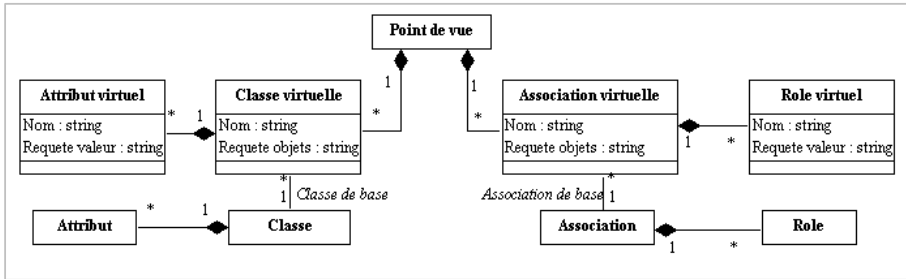


FIG. 3 Structure de points de vue sur les données dans ASTIS.

En ASTIS, un point de vue sur les données est défini en dérivant, à partir du schéma de données initial d'une application, des schémas virtuels, représentant l'information sous différentes perspectives (voir la FIG. 3). Ainsi, un point de vue de données est un schéma de données contenant les classes, associations, attributs et rôles virtuels (représentés respectivement par les classes *Classe virtuelle*, *Association virtuelle*, *Attribut virtuel* et *Rôle virtuel* dans la FIG. 3), et un ensemble de règles de correspondance entre les entités du schéma dérivé et les entités du schéma de base (représentées par les associations *Classe de base* et *Association de base* dans la FIG. 3). Notre approche est similaire à celle d'Abiteboul et Bonner (1991). Elle est ici étendue pour le modèle d'AROM qui adopte une représentation explicite des associations et de leurs instances.

Le concepteur peut employer plusieurs mécanismes pour dériver des points de vue :

- Masquer des classes et des associations, des attributs et des rôles ;
- Masquer les objets et les tuples non pertinents, en employant des requêtes qui permettent de ne retenir que ceux qui correspondent à certains critères de pertinence (l'attribut *Requête objets* pour la classe *Classe virtuelle* et, respectivement, l'attribut *Requête tuples* pour la classe *Association virtuelle*) ;
- Renommer des classes, des associations, des attributs et des rôles ;
- Fusionner en une seule classe des classes qui sont liées par des associations (l'attribut *Requête valeur* de la classe *Attribut virtuel* permet de rechercher la valeur de l'attribut à partir d'attributs des classes liées à la classe de base), ou, au contraire, d'écarter une classe en plusieurs classes (en définissant plusieurs classes virtuelles à partir de la même classe de base) ;
- Définir de nouvelles classes, en utilisant des requêtes pour regrouper des objets du même type (en employant l'attribut *Requête objets* de la classe *Classe virtuelle*).

Nous présentons un exemple d'obtention de point de vue (ou PDV), dans le cas de l'application dédiée au suivi des risques d'éboulement. L'idée est d'obtenir, à partir du schéma présenté en FIG. 4, un schéma dérivé contenant les classes et associations qui intéressent les touristes, en adéquation avec leur perspective sur les éboulements (voir la FIG. 5).

Une manière de définir ce point de vue est la suivante :

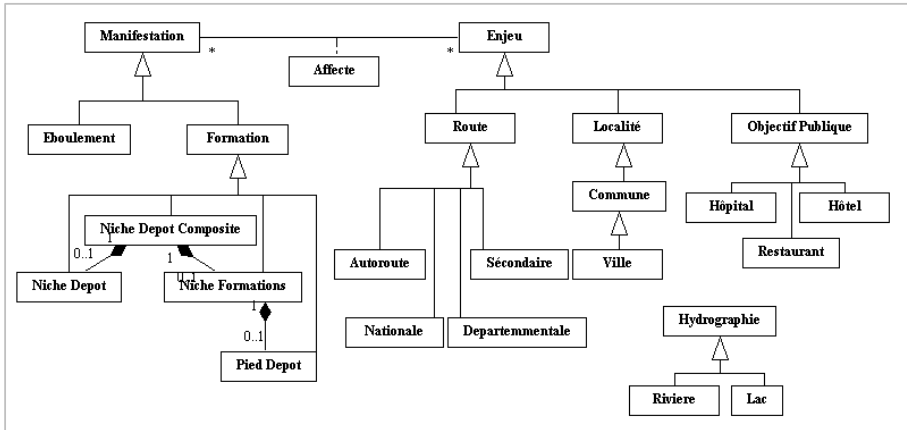


FIG. 4 Exemple de schéma de base dans une application dédiée aux risques d'éboulements.

viewpoint Touriste

viewpoints: BaseRisques

hide: Niche Depot, Niche Depot Composite, Niche Formations, Pied Depot, Formation, Affecte, Enjeu, Hôpital, Objectif Public, Eboulement, Manifestation

class Objectif Touristique

...

class Zone à risque

...

association dessert

...

association menace

...

association se situe

...

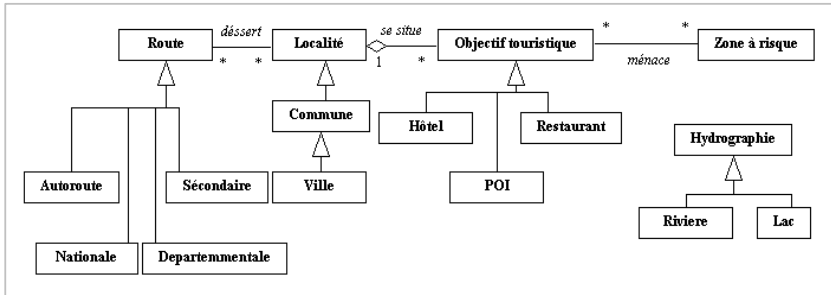


FIG. 5 Exemple de schéma dérivé pour le PDV « Touriste ».

Dans cet exemple, le concepteur a choisi donc d'exclure :

- les classes qui présentent des détails structurels sur la formation des éboulements (*Niche Depot*, *Niche Depot Composite*, *Niche Formations*, *Pied Depot*) ;
- la classe qui permet de connaître les conditions favorisant des éboulements (classe *Manifestation*), comme les pluies torrentielles, les changements brusques et extrêmes de température, etc. ;
- les classes *Enjeu*, *Objectif Publique*, et *Hôpital*, sous l'hypothèse que les seuls objectifs publics qui intéressent les touristes sont les objectifs touristiques ;
- l'association *Affecte*, qui ne peut plus exister, étant donné que les classes qu'elle relie ont été exclues.

Le concept de risque est représenté par une seule classe, *Zone à risque*, en considérant que l'information importante est dans le danger, et non pas dans sa source exacte. La classe *Objectif Touristique* est introduite pour généraliser les classes *Restaurant* et *Hôtel*. Les associations *dessert*, *menace*, *se situe* permettent de rendre compte des différentes relations de voisinage existantes entre les objets. La classe *POI* a été ajoutée afin de permettre aux touristes d'ajouter leurs propres annotations aux cartes, sous la forme de points d'intérêt.

Utilisons l'exemple de la pour illustrer ce type de classe. Par exemple, supposons que la classe *Restaurant*, présente dans le point de vue de base, ne contienne pas d'informations sur les types de cuisine proposés :

```

class: Restaurant
super-class : Enjeu
variables:
  variable: Nom
  type: string
  variable: Position
  type: point
    
```

Afin de permettre l'ajout de cette information, le concepteur peut redéfinir la classe *Restaurant* dans le PDV « touriste » ainsi :


```

class: Restaurant
super-class : Objectif Touristique
base-class : BaseRisques.Restaurant
variables:
  variable: Types Cuisine
  type: list-of-string
instances: set(r in BaseRisques.Restaurant: true)

```

Le concepteur définit la nouvelle classe comme classe à extension virtuelle, avec comme instances un ensemble d'instances virtuelles équivalent à l'ensemble d'instances de la classe de base. Il ajoute aussi l'attribut *Types Cuisine* qui permet aux touristes de maintenir la liste des types de cuisine proposée par chaque restaurant.

Lorsque la valeur de l'attribut *Types Cuisine* est mise à jour, le nouvel attribut est présent seulement au niveau du nouveau point de vue. Nous donnons ci-dessous un exemple d'instance réelle, stockée dans le PDV originel, suivi d'un exemple d'instance dérivée.

```

instance: Restaurant154869753
is-a: Restaurant
Nom: "Le Lantier"
Position: "-0.375769, 0.219575"

instance: RestaurantRestaurant154869753
is-a: Restaurant
base-instance: BaseRisques.Restaurant154869753
Types Cuisine: ["Française", "Italienne"]

```

4 Adaptation de la présentation

Les points de vue peuvent influencer non seulement les données à présenter aux utilisateurs, mais également la manière dont les données sont présentées. La plate-forme ASTIS permet de générer des interfaces de visualisation qui rendent compte des dimensions diverses de l'information (dimensions spatiale, temporelle et attributaire). Celles-ci sont affichées dans des fenêtres spatiales (cartes), temporelles (diagrammes temporels) ou attributaires (tableaux), dont les contenus sont synchronisés. Les mécanismes de synchronisation (ou *linking*, voir Buja et al., 1991) permettent de réaliser des requêtes visuelles et de visualiser leurs résultats sur toutes les fenêtres simultanément.

ASTIS permet aux concepteurs de spécifier des adaptations de la présentation qui sont appliquées dynamiquement à l'exécution de l'application, en fonction des caractéristiques des cibles (l'utilisateur et son contexte). Un modèle de présentation, dont un schéma simplifié est présenté en FIG. 6, permet de définir et de regrouper une série de transformations à appliquer sous la forme d'une visualisation. Le modèle de présentation d'ASTIS reprend la spécification SLD (*Styled Layer Descriptor*) de l'OGC (2002) et l'étend sur plusieurs aspects.

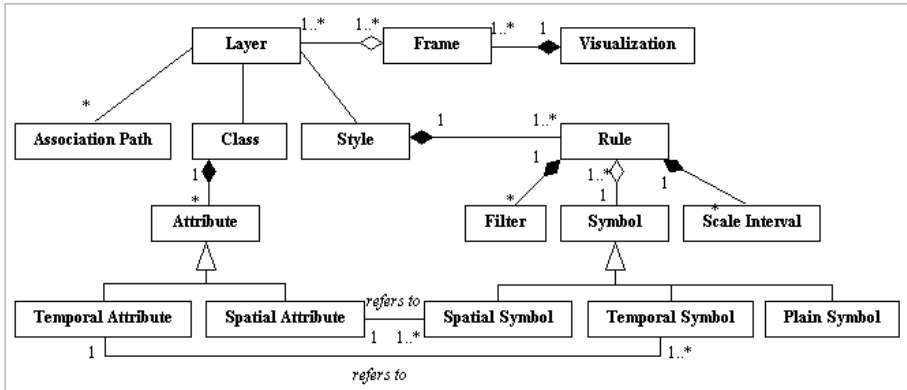


FIG. 6 Modèle de présentation simplifié d'ASTIS.

SLD est un standard de présentation qui permet de décrire des transformations de style (classe *Style* en FIG. 6) à appliquer à des couches de données (classe *Layer*). Une couche de données fait référence à une classe d'objets spatiaux (classe *Class* dans la FIG. 6, appelée aussi "feature type" dans la terminologie OGC). Un style est composé d'un ensemble de règles, qui permettent de décrire le type exact de symbole graphique correspondant à chaque objet spatial, en fonction de ses valeurs d'attributs (classe *Filter*). Par exemple, pour une couche représentant les rivières dont le symbole graphique est un trait bleu, un ensemble de règles permet de spécifier différentes épaisseurs du trait en fonction du débit de chaque rivière. Une règle peut aussi spécifier l'intervalle d'échelles pour lequel l'ensemble d'objets contenus est visible sur la carte.

Le modèle de présentation d'ASTIS étend le standard SLD sur deux aspects : la prise en compte des présentations temporelles et attributaires, et non pas seulement spatiales, ainsi que la prise en compte des éléments dynamiques de présentation, afin de permettre des présentations interactives. ASTIS permet de définir des visualisations complexes et multidimensionnelles (classe *Visualisation*) contenant plusieurs fenêtres (classe *Frame*) de type spatial, temporel ou attributaire, chaque fenêtre affichant un certain nombre de couches (de type spatial, temporel ou attributaire). De nouveaux types de symboles graphiques temporels (1D, classe *Temporal Symbol*) et attributaires (0D, classe *Plain Symbol*) ont été ajoutés aux types existants en SLD (2D, classe *Spatial Symbol*). Nous pouvons apercevoir un exemple de visualisation multidimensionnelle en FIG. 6 à gauche, contenant une fenêtre temporelle, une fenêtre attributaire et deux fenêtres spatiales, chacune des fenêtres étant peuplée de symboles de type correspondant : symboles spatiaux (polygones colorées en aplats, cercles proportionnels, signes conventionnels, texte ancré, etc.), symboles temporels (histogrammes), symboles attributaires (texte simple).

Afin de permettre la génération de présentations dynamiques et interactives, le modèle de présentation d'ASTIS étend la notion de couche décrite en SLD en ajoutant des informations concernant les liens existants entre les différents types d'objet (classe *Association Path*). Par exemple, dans une application de gestion d'avalanches (voir FIG. 7), en cliquant sur le contour d'une commune sur la carte (la commune avec un remplissage rouge) il est possible

de visualiser (toujours sur la carte) quels sont les sites avalancheux susceptibles de l'affecter (en violet), quelles sont les caractéristiques temporelles des événements avalancheux qui se sont produits sur ces sites (sur le diagramme temporel, en bas) ainsi que leurs caractéristiques attributaires (fenêtre attributaire, à gauche).



FIG. 7 Exemple de synchronisme interactif dans une application de visualisation de risques avalancheux.

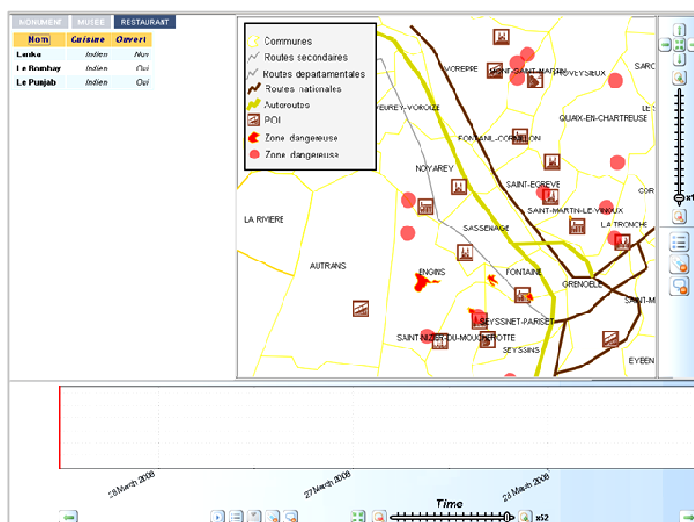


FIG. 8 Exemple de visualisation pour le point de vue « touriste ».

Nous reprenons l'exemple de l'application dédiée au suivi des risques d'éboulements pour montrer comment peuvent être adaptées les présentations aux besoins de l'utilisateur pour un certain point de vue. Considérons la conception des présentations selon un point de vue « touristique » (voir la FIG. 8) et selon un point de vue « cyndinicien » (voir la FIG. 9). La

visualisation pour le point de vue « cyndinicien » est conçue pour un groupe d'utilisateurs habitués aux outils de type SIST, alors que la visualisation selon le point de vue « touriste » s'adresse sans doute à des utilisateurs moins avisés.

Certaines différences de contenu entre les deux fragments de cartes sont visibles. Le point de vue « touristique » contient, par exemple, des informations sur des objectifs touristiques (musées, monuments, restaurants). Le point de vue « cyndinicien » contient plus d'informations liées à l'occupation des sols et au réseau hydrographique.

Au niveau de la structure, les deux visualisations contiennent des fenêtres attributaires, spatiales et temporelles. Pourtant, si pour le PDV « touriste » comporte plus de détails attributaires sur les points d'intérêt touristique, les tableaux attributaires pour les cyndiniciens apportent plus d'informations sur les risques mêmes. Au niveau de la fenêtre temporelle, les graphiques temporels permettent aux cyndiniciens d'avoir une perspective historique sur plusieurs décennies des phénomènes liés aux risques. La fenêtre temporelle de la visualisation dédiée aux touristes leur permet tout simplement d'accéder aux prévisions des dangers éventuels menaçant leurs points d'intérêt pour les prochains jours.

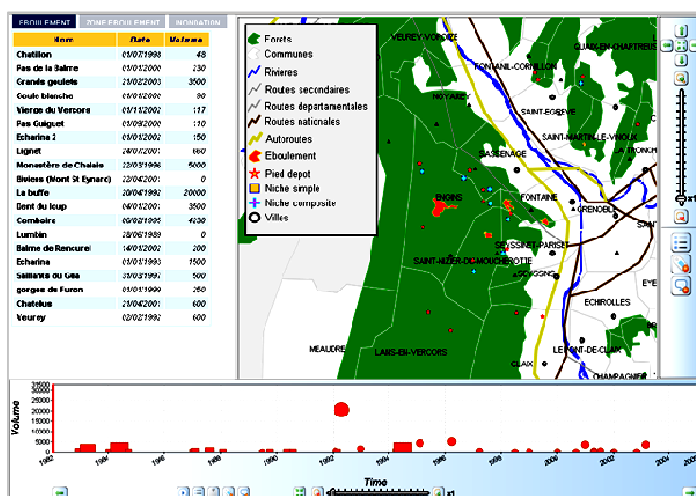


FIG. 9 Exemple de visualisation pour le point de vue « cyndinicien ».

La représentation du concept de risque est très simplifiée dans l'application dédiée aux touristes. Une zone rouge permet de mettre en évidence les différents types de zones d'éboulements qui sont à éviter par les touristes, permettant ainsi de les informer des risques existants lors de la préparation de leurs balades en montagne. Les différentes composantes d'une zone d'éboulements (pied de dépôt, niche de dépôt, niche composite, etc.) sont, au contraire, présentées en détail par des signes conventionnels familiers aux cyndiniciens.

En ce qui concerne les niveaux d'expertise, la visualisation s'adressant aux touristes est plus simple : la densité visuelle affichée (et, respectivement, la quantité totale d'information) est plus faible. Son style de présentation s'appuie sur l'utilisation de symboles visuels. Au contraire, la visualisation pour les cyndiniciens est plus dense et utilise des signes

conventionnels qui permettent d'afficher une quantité plus importante d'information, au détriment de la lisibilité.

5 Conclusions et perspectives

Nous avons présenté dans cet article une plate-forme pour la conception et la génération de SIST adaptables à l'utilisateur. Cette architecture implémente une approche conceptuelle de l'adaptation à l'utilisateur qui permet aux concepteurs de SIST de générer des applications adaptables en décrivant simplement des modèles conceptuels. Ces modèles doivent décrire la transformation applicable à la composante à adapter (les éléments du système qui sont transformés afin de répondre aux besoins de l'utilisateur) et la cible de l'adaptation (les caractéristiques de l'utilisateur ou de son environnement que le système doit prendre en compte pour l'adaptation). Afin d'aider les concepteurs dans la modélisation, la plate-forme offre des modèles génériques pour chaque type de cible et de composante à adapter, qu'il suffit aux concepteurs d'instancier. Cela permet aux mécanismes génériques d'adaptation implémentés au sein de notre plate-forme de réaliser l'adéquation aux types d'utilisateurs au niveau de l'application finale. Notre approche met en œuvre tant l'adaptation du contenu (les données), que l'adaptation de la présentation (les interfaces de visualisation).

Plusieurs perspectives sont possibles, en vue d'améliorer notre proposition et de la rendre plus utile pour la conception et la réalisation d'applications pour des SIST mobiles. La première vise la réalisation d'un module d'acquisition du contexte. Ce module devrait utiliser un modèle compréhensif de contexte et devrait permettre aux applications d'acquérir un maximum de paramètres du contexte (localisation, temps, paramètres du dispositif d'accès, historique de l'utilisateur, agenda, etc.) sans aucune intervention de la part de l'utilisateur ou du concepteur. La deuxième perspective est de compléter la plateforme avec un module d'adaptation dynamique, chargé de détecter les changements de contexte et de déclencher (en utilisant un mécanisme de type ECA – Event Condition Action) les adaptations nécessaires, c'est-à-dire régénérer de manière adéquate les parties de l'application influencées par le changement de contexte.

Références

- Abiteboul S. and Bonner A., Objects and Views, Proc. ACM-SIGMOD'91, 1991.
- Anegg H., Kunczier, H. Michlmayr, E. Pospischil G. and Umlauf M., LoL@: Designing a Location Based UMTS Application, ÖVE-Verbandszeitschrift e&i, Springer, 2002.
- Buja A., McDonald J. A., Michalak, J. et Stuetzle, W., Interactive data visualization using focusing and linking, Proc. of the 2nd Conference on Visualization '91, 1991.
- Conklin J., Hypertext: An introduction and survey. IEEE Computer 20 (9), pp. 17-41, 1987.
- Follin J.-M., Bouju, A., Bertrand, F. and Stockus, A., An Increment Based Model for Multi-resolution Geodata Management in a Mobile System, Proc. W2GIS 2005, LNCS, 2005.
- Lee C.-H., Chen L., Lee J.-D. and Bae H.-Y., Content Adaptation and Transmission Strategy of Spatial Information for WWW and Mobile Applications, HSI'03, LNCS, 2003.

- Laurini R. and Servigne S., Visual Access to City's websites: A Challenge for PDA's GUI. *Journal of Visual Languages and Computing*, 18 (3), p. 339-355, June 2007.
- MacEachren A.M., *VISUALIZATION - Cartography for the 21st century*. 7th Annual Conference of Polish Spatial Information Association. Warsaw, Poland, 1998.
- Malaka R. and Zipf A., DEEP, MAP Challenging IT Research in the Framework of a Tourist Information System. In: D. Fesenmaier, *Proceedings of ENTER 2000*. Wien, New York, Springer, 15-27, 2000.
- Moisuc B., Davoine P. A., Gensel J. and Martin H., Design of Spatio-Temporal Information Systems for Natural Risk Management with an Object-Based Knowledge Representation Approach, *Geomatica*, Vol. 59, No. 4, 2005.
- Moisuc B., Gensel J. et Martin H., Représentation de connaissances par objets pour les SIG à représentations multiples, *Proc. Cassini-SIGMA 2004*.
- Nilvala A-M. and Sarjakoski L.T., Preventing Interruptions in Mobile Map Reading Process by Personalisation. *Proc. 3rd Workshop on "HCI in Mobile Guides"*, in adjunction to: *MobileHCI'04*, 2004.
- OGC, *Styled Layer Descriptor Implementation Specification*, version 1.0.0, 2002.
- Page M., Gensel J., Capponi C., Bruley C., Genoud P., Ziebelin D., Bardou D. and Dupierris V., A New Approach to Object-Based Knowledge Representation: the AROM System, *LNAI*, 2001.
- Petit M., Ray C. and Claramunt C., A contextual approach for the development of GIS: Application to maritime navigation. *Proceedings W2GIS'06, LNCS*, 2006.
- Reichenbacher T., Adaptive Concepts for a Mobile Cartography, *Supplement Journal of Geographical Sciences*, December, 43-53, 2001.
- Reichenbacher T., Adaptive Methods for Mobile Cartography. *Proceedings of the 21st International Cartographic Conference (ICC) 2003*, Durban, 2003.
- Roth J., Context-Aware Web Applications Using the PinPoint. *Proceedings IADIS International Conference WWW/Internet 2002*, Lissabon (Portugal), IADIS Press, 3-10, 2002.
- Sarjakoski T., Sarjakoski L. T., Lehto L., Sester M., Illert A., Nissen F., Rystedt R. and Ruotsalainen R., Geospatial Info-mobility Services - A Challenge for National Mapping Agencies. *Proceedings Joint International Symposium on GeoSpatial Theory, Processing and Applications*, Ottawa, Canada, 2002.
- Schwinger W., Grün Ch., Pröll B., Retschitzegger W. and Schauerhuber. A., Context-awareness in Mobile Tourism Guides - A Comprehensive Survey, *Technical Report*, July 2005.
- Sester M. and Brenner C., Continuous Generalization for Fast and Smooth Visualization on Small Displays, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXV (B4:IV): 1293-1298, 2004.

Timpf S., Geographic Task Models for geographic information processing. Meeting on Fundamental Questions in Geographic Information Science, Duckham, M. and Worboys M.F. (eds.), Palace Hotel, Manchester, UK, pp.217-229, 2001.

Zipf A., User-Adaptive Maps for Location-Based Services (LBS) for Tourism, Proc. ENTER 2002, Innsbruck, 2002.

Zipf A., Location aware mobility support for tourists. Trends & Controversies. In: IEEE Intelligent Systems. Journal. Special Issue on "Intelligent Systems for Tourism". November/December 2002. 57-59, 2002.

Zipf A., Using Styled Layer Descriptor (SLD) for the dynamic generation of user- and context-adaptive mobile maps - a technical framework, In Proc. W2GIS'05, LNCS, 2005.

Summary. The joint development of the Web and of ubiquitous computing increases constantly the potential diversity of the users, access devices and contexts of use of spatio-temporal information systems accessed via these means. Adapting these systems to their users becomes a necessity and a guarantee of usability and longevity. This paper presents a generic approach for the design and generation of adaptive spatio-temporal information systems. The presented framework integrates generic user adaptation mechanisms, aimed at adapting both the content and the presentation of the applications. It allows designers to integrate these mechanisms into adaptable applications. In order to define the needs and the adaptations of their applications, designers only have to create conceptual models, by instantiating generic models supplied by our framework.

