

SOLIPSIS

Un monde virtuel massivement partagé

Gwendal Simon

France Telecom R&D - IRISA

Préambule

Environnement virtuel massivement partagé :

- **monde virtuel** généré par ordinateur (jeux multi-utilisateurs, entraînement militaire. . .)
- **entité virtuelle** :
 - > avatar d'un utilisateur ou représentation virtuelle d'un objet,
 - > position et déplacements dans le monde virtuel,
 - > interactions simulant la réalité.
- **présence** dans le monde virtuel :
 - > être perçu par d'autres entités,
 - > percevoir les entités proches virtuellement.

Préambule

- **Cohérence :**

- > définition : deux entités ont la même perception d'une même scène virtuelle,
- > objectif : trouver un compromis entre la quantité de messages échangés et le maintien de la cohérence dans un environnement dynamique.

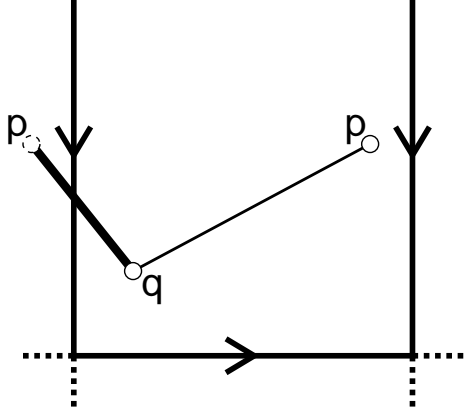
- Solutions proposées :

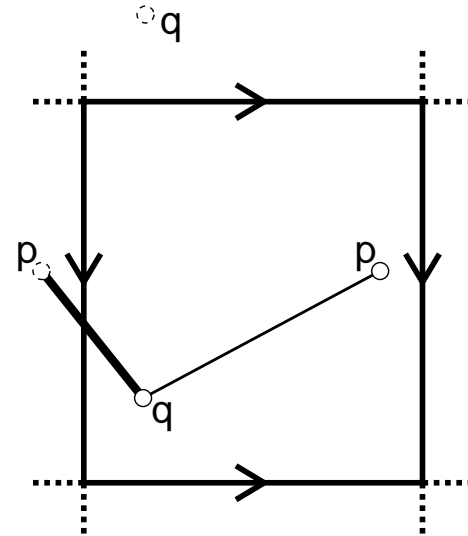
- > utilisation de **serveurs** :
 - ▷ réception et retransmission des événements (déplacements, modifications d'un avatar. . .)
- > utilisation du **multicast** :
 - ▷ découpage du monde en zones,
 - ▷ une entité responsable et un groupe multicast par zone.

Motivation

- Principe :
 - > Associer environnement virtuel partagé et réseaux de pairs :
 - ▷ le monde virtuel est une ressource partagée entre les utilisateurs (peer-to-peer),
 - ▷ chaque utilisateur est responsable de la ressource “lui-même” qu’il rend disponible,
 - ▷ chaque utilisateur collecte les informations lui permettant de reconstituer son environnement virtuel local.
- Objectifs :
 - > **scalabilité** : un nombre *illimité* de participants,
 - > **cohérence** : maintenir une cohérence *suffisante* (espace de rencontres virtuelles).

Modèle

- Le **monde virtuel** est un *tore* :
 - > un espace sans limite,
 - > un ensemble fini de coordonnées:
$$T = \{(x, y) \in \mathbb{N}_{size_x} \times \mathbb{N}_{size_y}\}$$
 - Un **nœud** est un élément logiciel :
 - > associé à une entité virtuelle,
 - > représenté par un identifiant unique id_e ,
 - > ayant une position dans le monde virtuel $(x_e, y_e) \in T$,
 - > connecté à d'autres nœuds (ensemble des adjacents $k(e)$),
 - > libre de se déplacer virtuellement à chaque instant,
 - > susceptible de quitter le monde à chaque instant.
- 
- The diagram illustrates a torus world, which is a 2D space with periodic boundaries. It is represented as a rectangle with two vertical lines on the left and right sides, each with a downward-pointing arrow, indicating that the space wraps around horizontally. A horizontal line at the bottom has a rightward-pointing arrow, indicating that the space wraps around vertically. A node 'q' is located in the center of the rectangle. Two nodes 'p' are located on the top edge, one on the left and one on the right, connected to node 'q' by lines, representing connections across the periodic boundaries.



Modèle

- Une **connexion** entre deux nœuds :
 - > est un lien bidirectionnel (connaissance mutuelle)

$$e' \in k(e) \Leftrightarrow e \in k(e')$$

- > permet de :
 - ▷ communiquer sa position à intervalle régulier,
 - ▷ s'échanger des flux multimédia,
 - ▷ s'informer à propos d'autres nœuds.
- L'**environnement virtuel local** est un disque $A(e)$:
 - > centré sur l'entité e ,
 - > de rayon r_e dépendant :
 - ▷ des capacités de l'entité e ,
 - ▷ de la densité d'entités.

Propriété

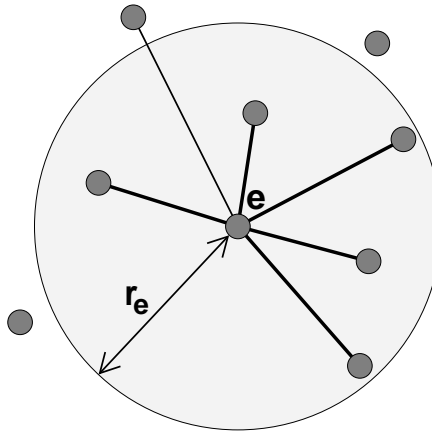
Principale propriété : **Local Awareness**

- Idée : une entité est connectée à tous ses plus proches voisins,
- Énoncé : la propriété est vérifiée par l'entité e quand :

$$a(e) \subseteq k(e)$$

avec :

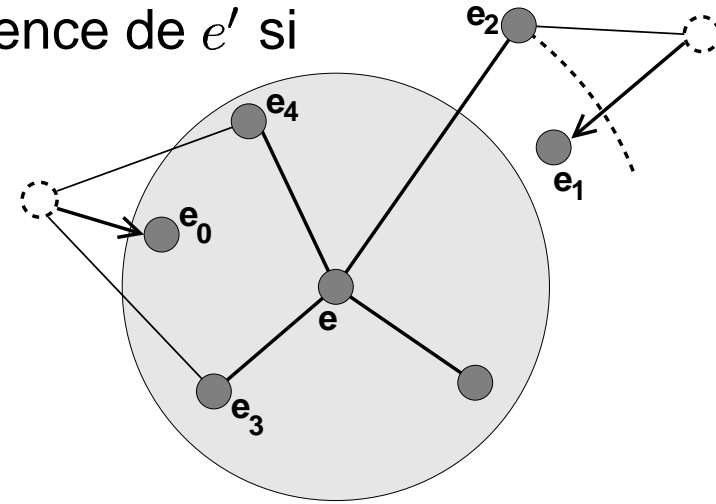
$a(e)$: l'ensemble des entités situées dans le disque $A(e)$



Maintien de la propriété

Collaboration spontanée :

- $\forall e', e'' \in k(e)$, l'entité e indique à e'' la présence de e' si
 - > e' entre dans $A(e'')$,
 - > $d(e'', e') < d(e'', e)$,

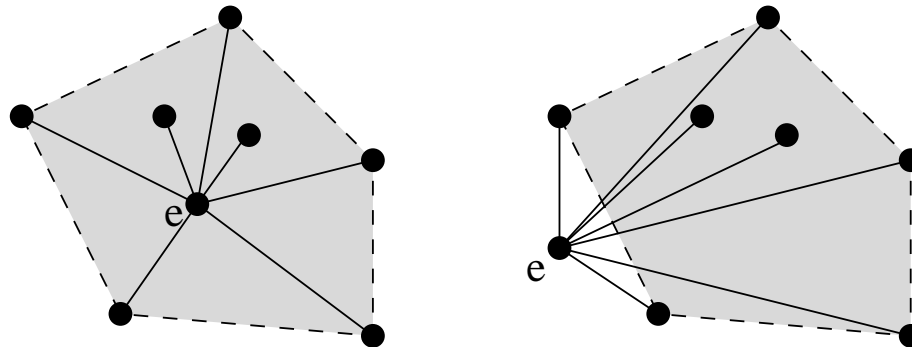


Collaboration requête/réponse :

- collecter des entités pour vérifier

$$pos_e \in CH(k(e))$$

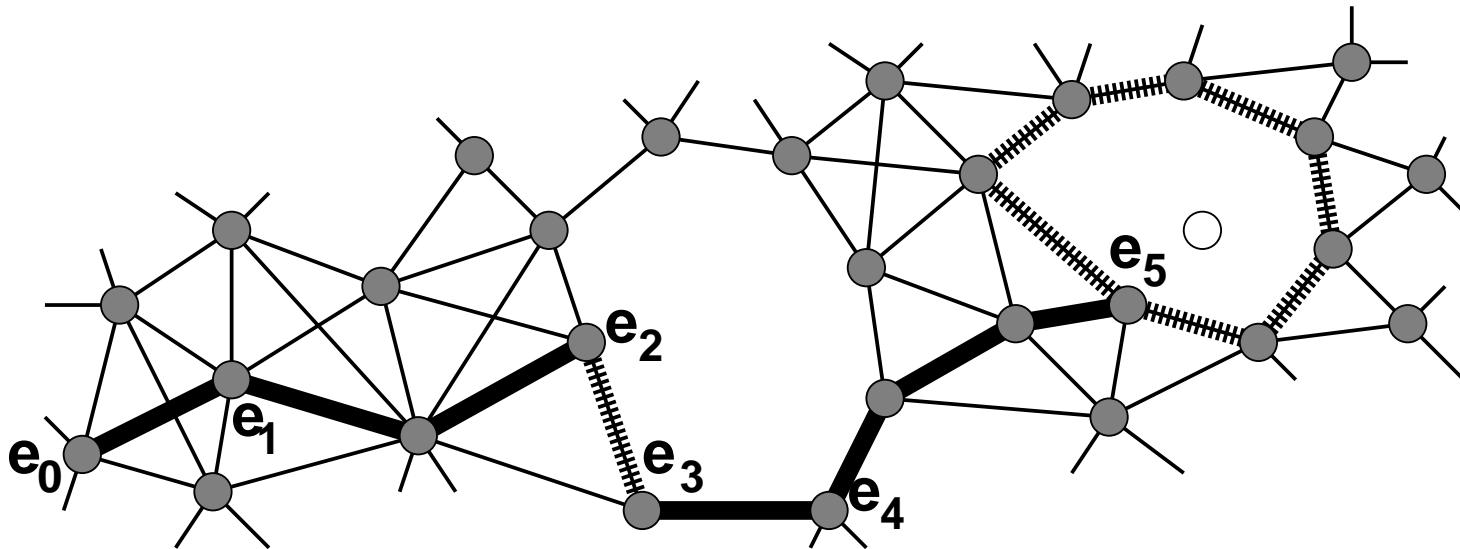
$CH(k(e))$: l'enveloppe convexe (*convex hull*) des adjacents de e .



Connexion & Téléportation

Pour une entité e qui se connecte ou qui se “téléporte” dans le monde virtuel :

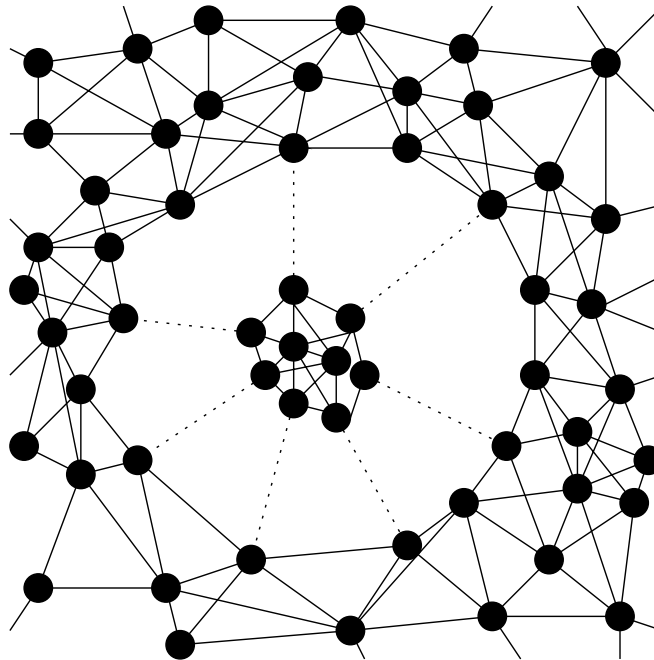
- e doit connaître une entité e_0 connectée et sa future position pos_e ,
- algorithme de localisation inverse :
 - > trouver l'entité la plus près de pos_e ,
 - > tourner autour de pos_e .
- définir r_e .



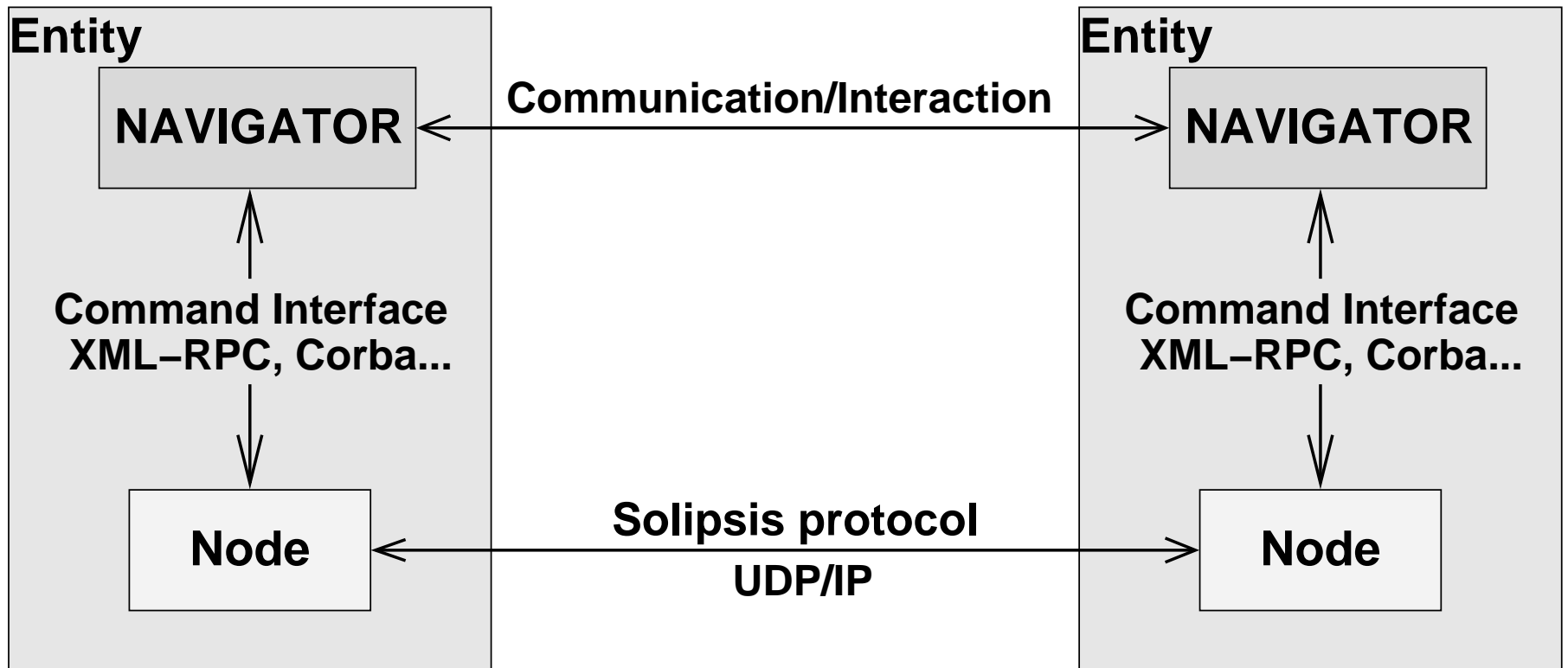
Résultats

Nos algorithmes garantissent:

- la connaissance des plus proches voisins,
- la connectivité en évitant la formation d'îles,
- le maintien des propriétés dans un contexte de mobilité, i.e. d'instabilité.



Architecture Générale

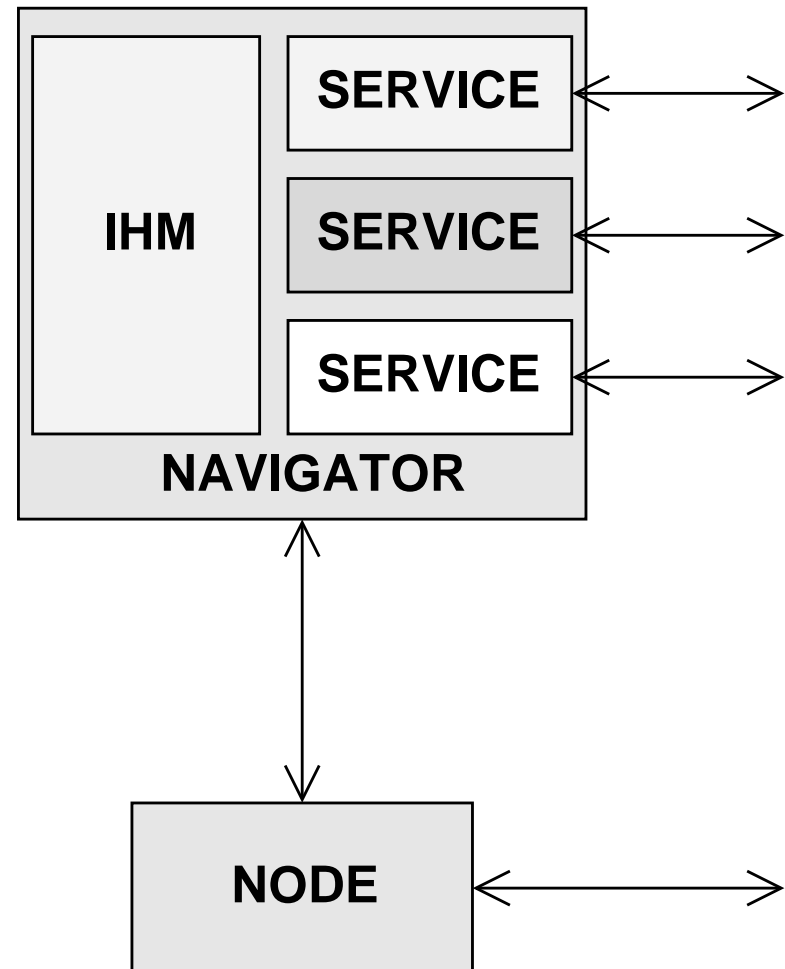


Tâches périodiques

- **gestion du voisinage :**
 - > détection de défaillance
 - ▷ message `Heartbeat`
 - > nombre de voisins constant (dans un intervalle)
 - ▷ message `Close`
- **maintien des propriétés topologiques :**
 - > aucun demi-plan n'est vide
 - ▷ messages `Search` et `Found`
 - > nombre constant de voisins dans le disque virtuel local
 - ▷ message `Delta;AR`
- **récupération d'informations pour le système :**
 - > gestion du fichier *bootstrap*

Événements Utilisateurs

- **ordre IHM :**
 - > déplacements :
 - ▷ message `Delta;POS`
 - > téléportation
 - > ...
- **gestion des services :**
 - > message `Service`
 - > message `EndService`



Événements du Voisinage

- nouvelles connexions :
 - > réaction à une détection
 - ▷ message Detect
 - > ouverture spontanée
 - ▷ messages Hello et Connect
- mobilité :
 - > message Delta;POS
 - > message Delta;AR
- informations pour les navigateurs
 - > messages Service et Endservice



Conclusion & travaux futurs

- aujourd'hui :
 - > création et animation d'une communauté Open-Source
 - > présentation à la conférence internationale *CodeCon'04* en février
 - ▷ **<http://solipsis.netofpeers.net>**
- demain :
 - > optimisation du protocole
 - > gestion des collisions
 - > utilisation de DHTs pour la connexion
 - > confiance distribuée pour argent virtuel
 - > de nouveaux services et de nouveaux navigateurs ...