Méthodes formelles et auto-assemblage

Fabien Tarissan

PPS, Université Paris 7

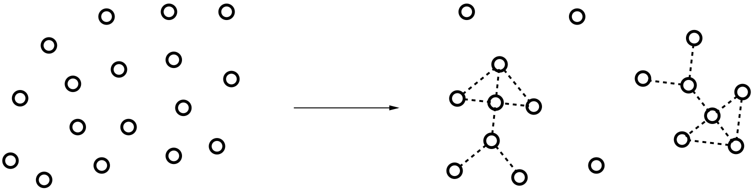
23 Janvier 2006

Le problème

- L'auto-assemblage : Comment un phénomène collectif peut surgir d'intéractions élémentaires ? (analyse et synthèse)
- Problème récurrent
  - Biologie moléculaire (analyse)
  - Ingénierie génétique (synthèse)
  - Robotique distribuée (synthèse)

Le problème

- L'auto-assemblage : Comment un phénomène collectif peut surgir d'intéractions élémentaires ? (analyse et synthèse)
- Problème récurrent
  - Biologie moléculaire (analyse)
  - Ingénierie génétique (synthèse)
  - Robotique distribuée (synthèse)



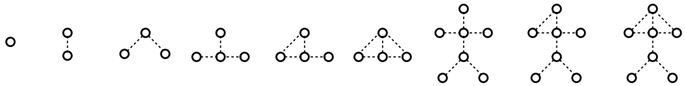
Approche formelle

- Modèle de liaison, espace, ...
  Quels sont les objets construits ?
- Capacité combinatoire des composants :
  Qu'est-ce qu'un élément de base peut calculer ?
  L'hypothèse raisonnable dépend du contexte.
- Modèle de communication :
  Comment interagissent les agents ?
  Quel est le type d'information échangé ?

On recherche une syntaxe capable de décrire à la fois les éléments et l'assemblage.

Travail préliminaire

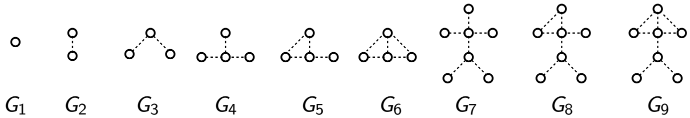
- $\mathcal{G}$: Ensemble de graphes intermédiaires :



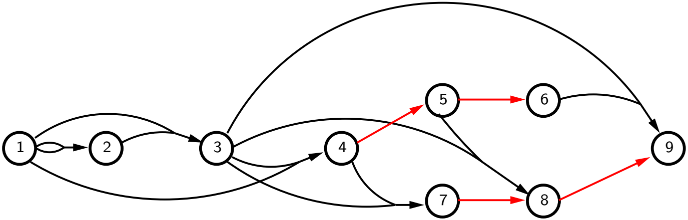
G_1 G_2 G_3 G_4 G_5 G_6 G_7 G_8 G_9

Travail préliminaire

- $\mathcal{G}$: Ensemble de graphes intermédiaires :



- Graphe d'assemblage du motif final :



Une syntaxe pour les graphes ...

- $Canaux = x, y, \ldots$: ensemble dénombrable.
- $\mathcal{RA} = R, S, \ldots$: ensemble des réseaux d'agents défini par

$$\begin{array}{rcll} R & := & \emptyset & \\ & \mid & < C > & \qquad (C \subseteq Canaux) \\ & \mid & R, R & \\ & \mid & (\nu x)R & \end{array}$$

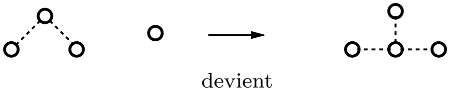
- Nœuds $=$ agents
- Arcs $=$ partages de noms

devient $< x >$, $< x, y >$, $< y >$

$RTK < rtk^x >$, $RTK < rtk^x, shs^y >$, $SHS < rtk^y >$

... et pour l'assemblage

- Règles de construction = paire de réseaux : $R \rightarrow R'$.
- Exemple



$$\langle x \rangle \ , \ \langle x, y \rangle \ , \ \langle y \rangle \ , \ \langle \rangle \quad \longrightarrow \quad (\nu z)\,(\langle x \rangle \ , \ \langle x, y, z \rangle \ , \ \langle y \rangle \ , \ \langle z \rangle)$$

- Évolution des réseaux = réécriture de graphes.

Modélisation

- Au départ : $(Agents, Réactions)$ où
  $Agents = <>, <>, \ldots, <>$
  $Réactions = \{\underbrace{<>, \ldots, <>}_{\|V\| \ \mathit{fois}} \rightarrow [(V, E)]_{\mathcal{RA}}\}$

- Étape intermédiaire : $(Agents, Réactions')$ où
  $Réactions'$ représente les règles du graphe d'assemblage.

Ne résoud toujours pas le problème d'auto-assemblage

Formalisation du problème

- Extraction d'un fragment du langage :
  $\langle x \rangle \ , \ \langle x \rangle \ , \ \langle \rangle \ \not\twoheadrightarrow \ (\nu y)(\langle x \rangle \ , \ \langle x, y \rangle \ , \ \langle y \rangle)$
  $\Longrightarrow$ restriction sur les capacités de synchronisation
- Propriété voulue : comportement identique

Formalisation du problème

- Extraction d'un fragment du langage :
  $\langle x \rangle \ , \ \langle x \rangle \ , \ \langle \rangle \ \not\rightarrow \ (\nu y)(\langle x \rangle \ , \ \langle x, y \rangle \ , \ \langle y \rangle)$
  $\Longrightarrow$ restriction sur les capacités de synchronisation
- Propriété voulue : comportement identique

Que veut dire même comportement ?

- Comparaison des transitions
- Comparaison des états

Formalisation du problème

- Extraction d'un fragment du langage :
  $\langle x\rangle \ , \ \langle x\rangle \ , \ \langle\rangle \ \not\rightarrow \ (\nu y)(\langle x\rangle \ , \ \langle x, y\rangle \ , \ \langle y\rangle)$
  $\Longrightarrow$ restriction sur les capacités de synchronisation
- Propriété voulue : comportement identique

Que veut dire même comportement ?

- Comparaison des transitions
- Comparaison des états

$\Longrightarrow$ Outil mathématique : bisimulation

Bisimulation

[*Observation*]
La plus petite relation $\downarrow$ vérifiant $S \downarrow C$ si la composante connexe C apparaît dans le système S.

[*Bisimulation avec observables (faible)*]
Relation binaire symétrique $\mathfrak{R}$ telle que si $S \ \mathfrak{R} \ S'$ alors

1. Si $S \rightarrow T$ alors $\exists \ T'$ t.q. $S' \rightarrow^* T'$ et $T \ \mathfrak{R} \ T'$
2. Si $S \downarrow C$ alors $S' \downarrow C$.

Une idée de l'algorithme

- Un agent actif par composante connexe.
- Connaissance locale de la structure du complexe
- Chaque agent connaît le rôle qu'il joue dans le complexe
- Propagation des changements liés à une interaction le long d'un arbre de recouvrement

Des agents structurés

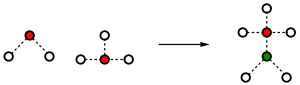
$$\text{Agent} = < S, C, g, r, e >$$

- S : noms partagés par les voisins dans l'arbre de recouvrement.
- C : autres noms privés.
- g : identifiant de la composante connexe.
- r : rôle joué dans la composante connexe.
- e : état de l'agent (actif, passif, mise à jour)

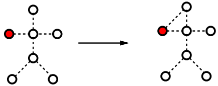
Traduction des reactions

Familles de réactions :

- Création d'une connexion
  - entre 2 complexes disjoints



  - dans un même complexe



- Propagation des mises à jour
- Échange d'activité
- Mécanisme pour gérer les échecs

Règle de connexion

$\forall G_1, r_1, G_2, r_2$ such that $[G_1.r_1 \oplus G_2.r_2]^{abs} \in \mathcal{G}$

$$
\begin{array}{c}
\langle\ S_1\ ,\ C_1\ ,\ g_1\ ,\ r_1\ ,\ Act(G_1)\ \rangle \\
\langle\ S_2\ ,\ C_2\ ,\ g_2\ ,\ r_2\ ,\ Act(G_2)\ \rangle \\
\downarrow \\
(\nu\ com) \\
\langle\ S_1 \cup \{com\}\ ,\ C_1\ ,\ g_1\ ,\ r_1\ ,\ Act(G_1.r_1 \oplus G_2.r_2)\ \rangle \\
\langle\ S_2 \cup \{com\}\ ,\ C_2\ ,\ g_1\ ,\ r_2 + \|G_1\|\ ,\ Up(S_2, \|G_1\|)\ \rangle
\end{array}
$$

Connexion cyclique

$\forall G, r_1, r_2$ such that $[r_1 \overset{G}{\frown} r_2]^{abs} \in \mathcal{G}$

$$\begin{array}{c} \langle\ S_1\ ,\ C_1\ ,\ g\ ,\ r_1\ ,\ Act(G)\ \rangle \\ \langle\ S_2\ ,\ C_2\ ,\ g\ ,\ r_2\ ,\ \quad P \quad\ \rangle \\ \downarrow \\ (\nu\ com) \\ \langle\ S_1\ ,\ C_1 \cup \{com\}\ ,\ g\ ,\ r_1\ ,\ Act(r_1 \overset{G}{\frown} r_2)\ \rangle \\ \langle\ S_2\ ,\ C_2 \cup \{com\}\ ,\ g\ ,\ r_2\ ,\ \quad P \quad\ \rangle \end{array}$$

Mises à jour

- Propagation

$$\begin{array}{c}
\langle \; S_1 \; , \; C_1 \; , \; g_1 \; , \; r_1 \; , \; Up(L + \{com\}, Num) \; \rangle \\
\langle \; S_2 + \{com\} \; , \; C_2 \; , \; g_2 \; , \; r_2 \; , \; P \; \rangle \\
\downarrow \\
\langle \; S_1 \; , \; C_1 \; , \; g_1 \; , \; r_1 \; , \; Up(L, Num) \; \rangle \\
\langle \; S_2 + \{com\} \; , \; C_2 \; , \; g_1 \; , \; r_2 + Num \; , \; Up(S_2, Num) \; \rangle
\end{array}$$

- Fin de la phase de mise à jour

$$\langle S, C, g, r, Up(\emptyset, Num) \rangle \longrightarrow \langle S, C, g, r, P \rangle$$

Échange d'activité

$$\begin{array}{c} \langle\ S_1\ ,\ C_1\ ,\ g\ ,\ r_1\ ,\ Act(G)\ \rangle \\ \langle\ S_2\ ,\ C_2\ ,\ g\ ,\ r_2\ ,\ \quad P \quad\ \rangle \\ \downarrow \\ \langle\ S_1\ ,\ C_1\ ,\ g\ ,\ r_1\ ,\ \quad P \quad\ \rangle \\ \langle\ S_2\ ,\ C_2\ ,\ g\ ,\ r_2\ ,\ Act(G)\ \rangle \end{array}$$

Début de la dislocation

$$\langle S, C, g, r, Act(G)\rangle \longrightarrow \langle S, C, g, r, AI\rangle$$

Propagation

- Aux voisins dans l'arbre de recouvrement

$$\begin{array}{c} \langle\ S_1 + \{com\}\ ,\ C_1\ ,\ g_1\ ,\ r_1\ ,\ Al\ \rangle \\ \langle\ S_2 + \{com\}\ ,\ C_2\ ,\ g_2\ ,\ r_2\ ,\ P\ \rangle \\ \downarrow \\ \langle\ S_1\ ,\ C_1\ ,\ g_1\ ,\ r_1\ ,\ Al\ \rangle \\ \langle\ S_2\ ,\ C_2\ ,\ g_2\ ,\ r_2\ ,\ Al\ \rangle \end{array}$$

- Aux voisins venant de connexions cycliques

$$\begin{array}{c} \langle\ S_1\ ,\ C_1 + \{com\}\ ,\ g_1\ ,\ r_1\ ,\ Al\ \rangle \\ \langle\ S_2\ ,\ C_2 + \{com\}\ ,\ g_2\ ,\ r_2\ ,\ P\ \rangle \\ \downarrow \\ \langle\ S_1\ ,\ C_1\ ,\ g_1\ ,\ r_1\ ,\ Al\ \rangle \\ \langle\ S_2\ ,\ C_2\ ,\ g_2\ ,\ r_2\ ,\ Al\ \rangle \end{array}$$

Retour à l'état initial

$$\langle \emptyset, \emptyset, g, r, Al \rangle \longrightarrow (\nu g)\langle \emptyset, \emptyset, g, 1, Act(G_1) \rangle$$

Une présention plus intuitive

Sortir des impasses

Compétition pour les ressources $\Longrightarrow$ impasses.

Plusieurs mécanismes possibles :

- Méthode brutale : dislocation de la composante (agents en bleu).
- Solution plus naturelle : revenir en arrière.
  1. On conserve la partie incrémentale.
  2. On précise quelles sont les étapes irréversibles.
  3. On plonge l'algorithme dans $\mathcal{R}\pi$.

Conclusions

- Ce qu'on a fait :
  - Un langage formel pour décrire le problème de l'auto-assemblage
  - Un outil mathématique pour résoudre ce problème
- Extensions :
  - Espace et mobilité
  - Algorithme plus proche de la biologie
- Systèmes complexes, multi-échelles