Disponibilité de services dans des composants logiciel

DISPO

Thomas Jensen

.

IRISA/CNRS

Organisation

Partenaires :

- École des Mines de Nantes, équipe OBASCO
- ENSTB dépt. RSM
- IRIT/SVF
- IRISA, projet Lande.

Objectifs :

Intégrer la sécurité dans la conception de logiciels.

Sécurité de systèmes informatiques

- Confidentialité
- Intégrité
- **Disponibilité**

Conception modulaire de logiciels :

- composants et interfaces,
- conception et programmation par aspects

Technique de validation :

- analyse statique,
- vérification de modèles,
- tissage de moniteurs.

Disponibilité

Disponibilité :

> *Aptitude à répondre à une demande d'un service, d'une ressource en garantissant des contraintes de performances*

Politiques de disponibilité :

- Yu et Gligor
  - Fondé sur la logique temporelle (*finite waiting time*)
  - *User requirements* : des contraintes sur le comportement des utilisateurs.
- Millen
  - moniteurs d'allocations de ressources
  - modèle basé sur des systèmes de transitions
- Cuppens et Saurel
  - logiques temporelles et déontiques
  - cohérence d'une politique de disponibilité

Analyses de politiques de disponibilité

Politique de disponibilité :

R1 : Le sujet S doit pouvoir commencer à réaliser la tâche T au plus tard 6 unités de temps après sa requête.

R2 : Le sujet S doit pouvoir disposer de la ressource nécessaire à T au plus tard 4 unités de temps après l'avoir demandé

R3 : La durée maximale de réalisation de la tâche T est de 3 unités de temps.

- Vérifier la cohérence d'une politique de disponibilité (est-ce qu'il y a un modèle ?)
- Extraire des exigences sur les processus à vérifier ou à imposer ("un processus doit relâcher une ressource immédiatement après avoir fini de l'utiliser").

Analyse de disponibilité

À l'intérieur d'un composant, vérifier une propriété par analyse statique et model checking

- Extraire des modèles abstraits d'un composants (flot de contrôle, flot de données) par interprétation abstraite
- Vérification de propriétés comportementales sur les modèles extraits
- Inférence de pre-/post conditions pour des fragments de programme

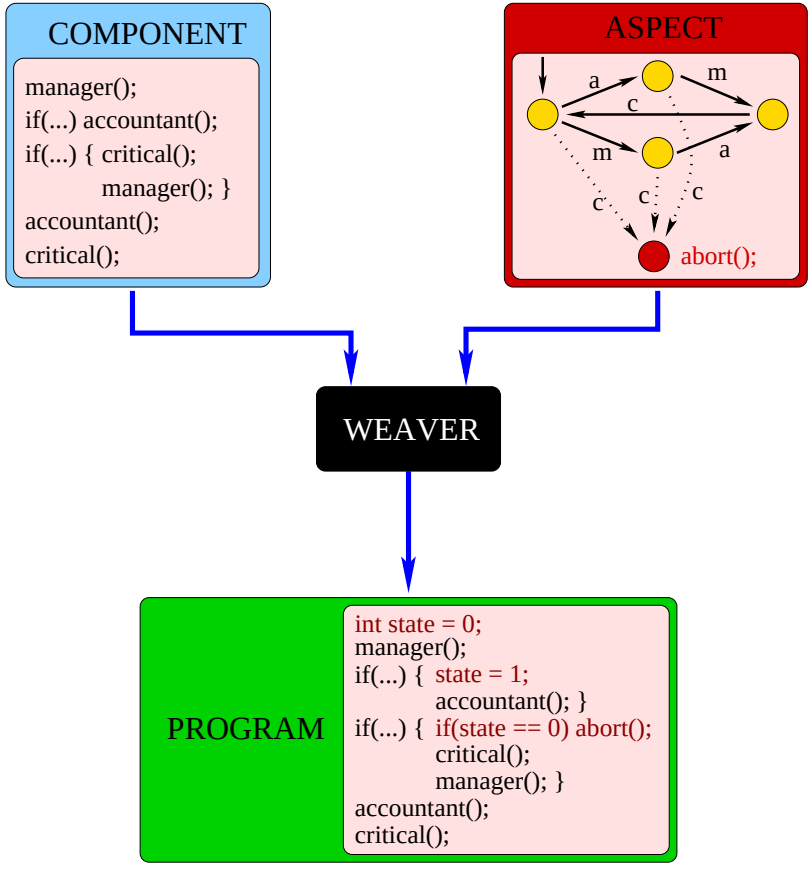
$$\mathbf{f_1 : P_1, \ldots f_n : P_n \vdash C.s_1 : R_1 \ldots C.s_k : R_k}$$

où $P_1, \ldots, P_n, R_1, \ldots, R_k$ sont des propriétés de consommation de ressources.

- raisonnement *assume-guarantee* sur une collection de composants
- vers une notion d'interface de disponibilité.

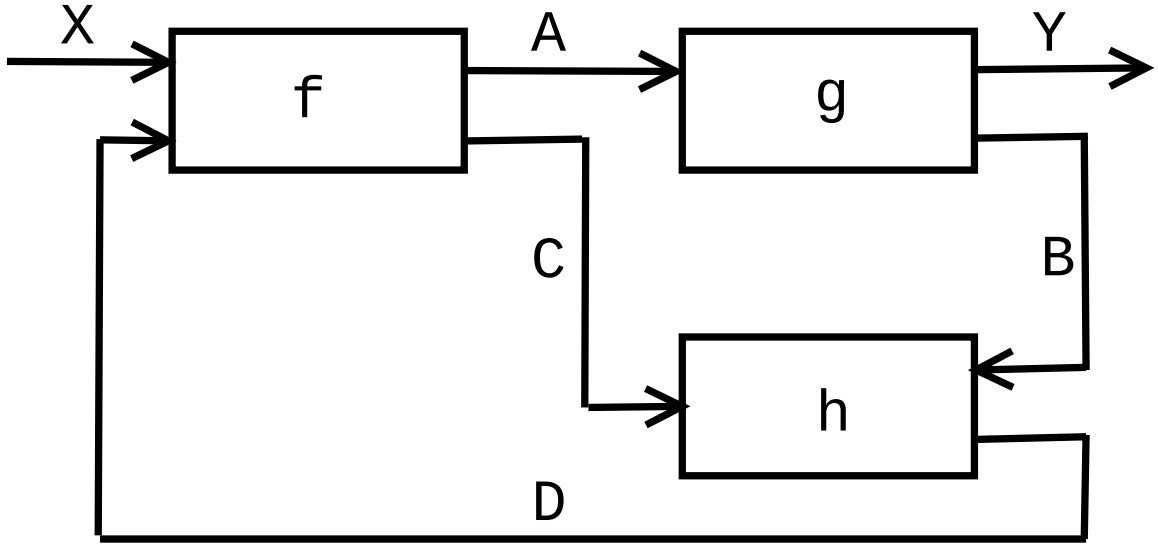
Aspects et tissage

Assurer/imposer une propriété par tissage d'aspects (insertion d'un moniteur de référence)



Composants et interfaces de disponibilité

- Interfaces : ports + protocoles de communication + ...
- Des propositions industrielles. Modèle «minimal» basé sur les réseaux de Kahn :



Sémantique dénotationnelle «flot de donnée»

- Langages de composants : ArchJava
- Raisonnement compositionnel sur les propriétés de composants.
- Extraction et composition d'interfaces.
- Tissage au niveau des composants

DISPO

Partenaires :

- École des Mines de Nantes, équipe OBASCO
- ENSTB dépt. RSM
- IRIT/SVF
- IRISA, projet Lande.

Thèmes :

- Politiques de disponibilité
- Analyse et tissage de propriétés de disponibilité
- Intégré dans un modèle de composants de logiciels