Systèmes Dynamiques, Opportunistes et Distribuées

DISTRIMAN :   
Distribution et Déploiement

## Objectif

Permettre à des applications s’exécutant sur une machine Apam de requérir et/ou fournir des instance/composant s’exécutant sur des machines Apam distante. Les machine Apam distante sont non connues à l’avance du point de vue des applications et peuvent évoluer au cours du temps.

Dans un deuxième temps, permettre à des applications s’exécutant sur une machine Apam de déployer ces dépendances (instances/composant) sur des machines distante.

Un système distribué est une application à services constitué de services qui communiquent. Un modèle de système distribué exprime l’évolution, la distribution et le déploiement dynamique d’une application à service.

## Éléments

**Distriman** est le manager qui permet à une machine Apam d’utiliser des services d’une autre machine Apam, et/ou de déployer des services en distant. Distriman découvre et communique avec les ExpAm.

**ExpAm** est un gestionnaire présent (singleton) sur toute machine Apam qui veut exporter des services. ExpAm représente sa machine Apam auprès des Distriman. ExpAm et Distriman ont un protocole (décrit ci-dessous) permettant de rendre interopérables les services Apam.

Hypothèses[[1]](#footnote-1)

* Distriman réifie tous les ExpAm visibles sous la forme d’un composite ayant le nom et les propriétés de la machine distante. Ce composite représente la machine distante et contient les implémentations distantes actuellement utilisées.
* Distriman interprète un modèle exprimant, pour chaque composite local, quelles sont les dépendances qu’il est possible de résoudre en distant.
* ExpAm interprète un modèle exprimant, pour chaque composite local, quels sont les composants qui **peuvent** être exportés vers d’autres machines Apam.
* Importer un service est similaire à déployer ce service (via OBRMAN par exemple). Importer un service revient à créer :
  + Une implémentation (Apam), ayant le meme nom et propriétés de l’implémentation originale, mais « instantiable=false ». Cette implémentation étant « déployée », elle est contenue dans le composite type appelant, et également dans le composite Machine représentant la machine distante. Les implémentations distantes et leurs clones sont immutables (il n’est pas possible, en local ou en distant, de modifier les propriétés ou les définitions).
  + Une instance (Apam), ayant le meme nom et propriétés que l’instance originale. L’instance étant créée, appartient au composite appelant (et aura donc la visibilité définies par ce composite). L’instance est modifiable, en local comme en distant. La modification des propriétés d’une instance provoque la destruction/recréation des proxys et end-point correspondants pour synchroniser les valeurs[[2]](#footnote-2).
  + Un proxy (en local) et un end-point (en distant).
  + Le proxy est le serviceObject associé à l’instance Apam.

## Modèle de distribution

Un modèle Distriman contient la description des imports et exports d’un ou plusieurs composites.

<distriman>   
 <composite type-name=”Expr”>  
 <import specification=*"true"* machExp*=”Exp”* install*=”Exp” /*>  
 <import implementation=*"A\*-lib"* machExp*=”Exp”* install*=”Exp” /*>  
 <export specification=*"A\*-lib" |* implementation=”Exp”/>  
 </composite>  
 <composite type-name= …..  
</distriman>

Expr est une expression LDAP, ou « true » ou « false ».

Le modèle précédent est associé à un bundle et décrit les stratégies de Distriman (et ExpAm) pour les divers composites présents dans ce bundle.

If, for a given composite, no export is provided, that composite is not visible from outside the current machine. If, for a given composite, no import is provided, that composite dependencies will be resolved only inside the current machine.

**<import**  This tag expresses that a distant resolution is required if

* The source of the resolution pertains to the current composite, and
* **specification=  “Exp” | implementation= “Exp” | interface= “Exp” | message= “Exp”**  
  The resolution target is matching the content (i.e. the target is respectively of the type specification, implementation, interface of message) and its name matches the expression.
* **machExp =  « exp »** expresses that the target resolution must be intended on the remote machines that satisfies the expression. The expression is evaluated locally against the properties available on that machine representative. If the selected machine does not owns a component satisfying the dependency constraints, another machine is selected, until a satisfactory component is selected or all machines are tried. If no solution is found, returns null. If more than one machine match the expression, they are tried in a random order.   
  If *machExp* is missing, *machExp=“true”* is assumed (i.e. all visible ExpAm).
* **Install =  « Exp »**. If no resolution is found, *install* expresses the condition under which a remote deployment can (and must) be intended. The expression is evaluated against the properties of the machine representative. If the expression is satisfied, the resolution is intended, OBRMAN enabled, on the corresponding distant machine.   
  If *install* is missing, *install=false* is assumed (no remote deployment).

**<export specification=  “Exp” | implementation= “Exp”**

This tag is interpreted par ExpAm. It indicates which components of the current composite are visible from (exported to) other Apam machines. Any implementation that matches one or the other expressions is visible. By default all implementations are visible. Seules les instances ayant une visibilité globale sont exportées.

Figure 1 Exemple d'export d'une instance.

DM

Client (MC)

Server (MS)

MS

MC

p

e

DM

MS

Composite Server (MS)

MC

Composite Client (MC)

e

p

ServiceObject Proxy

Endpoint du ServiceObject

1. This is a minimal specification; it will be extended later on, if required. [↑](#footnote-ref-1)
2. Solution couteuse mais simple et déjà disponible dans Rose. On supposera que les modifications de propriétés ne sont pas trop fréquentes sur des services distants. [↑](#footnote-ref-2)