ONERA

Journées Altarica, Marseille, 22-23 octobre 2003

Safety Architecture Patterns (SAP)

P. BIEBER, C. CASTEL, C. KEHREN, C. SEGUIN

Email : {bieber,castel,kehren,seguin}@cert.fr

ONERA-CERT
BP 4025, 2 Avenue Edouard-Belin
31055 Toulouse - FRANCE

Plan

1. Contexte

2. Présentation des *SAP*

3. Exemple d’utilisation

4. Conclusion

Plan

1. Contexte

2. Présentation des *SAP*

3. Exemple d’utilisation

4. Conclusion

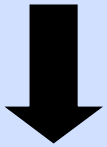
Contexte

ESACS

→ Projet européen ESACS (*Enhanced Safety Assessment for Complex Systems)* : 2001 – 2003

► Utilisation de notations formelles (Altarica, Scade, …)

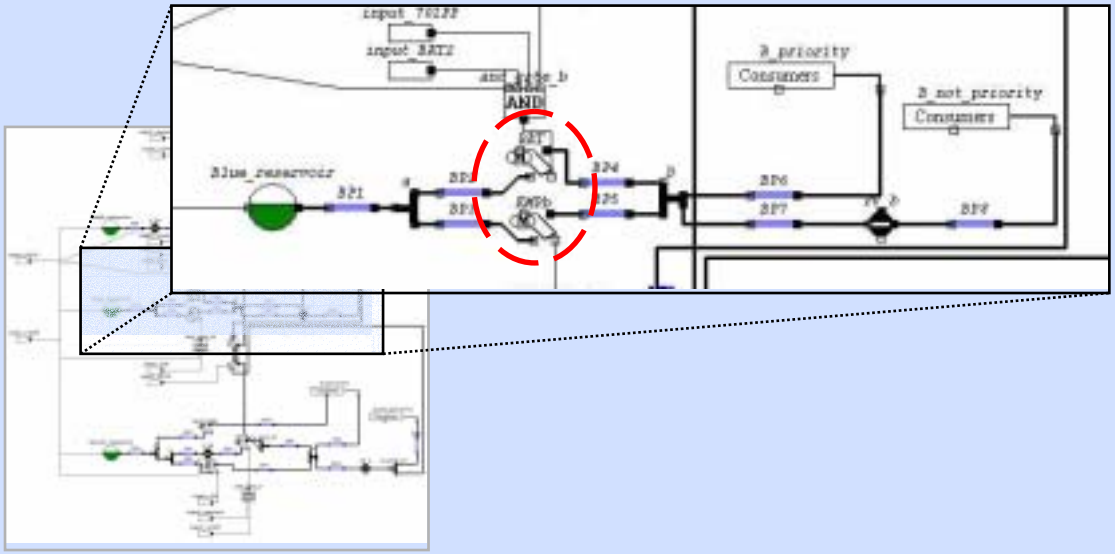
→ Modélisation de plusieurs systèmes Airbus (hydraulique, électrique)



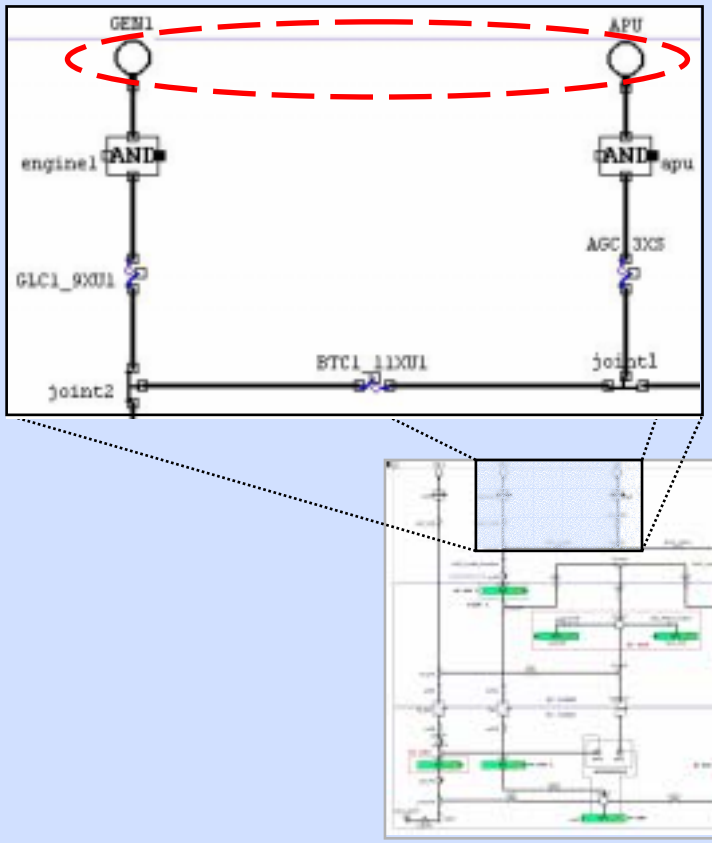
Mise en évidence d'architectures de sûreté récurrentes

Contexte

Exemples



HYD



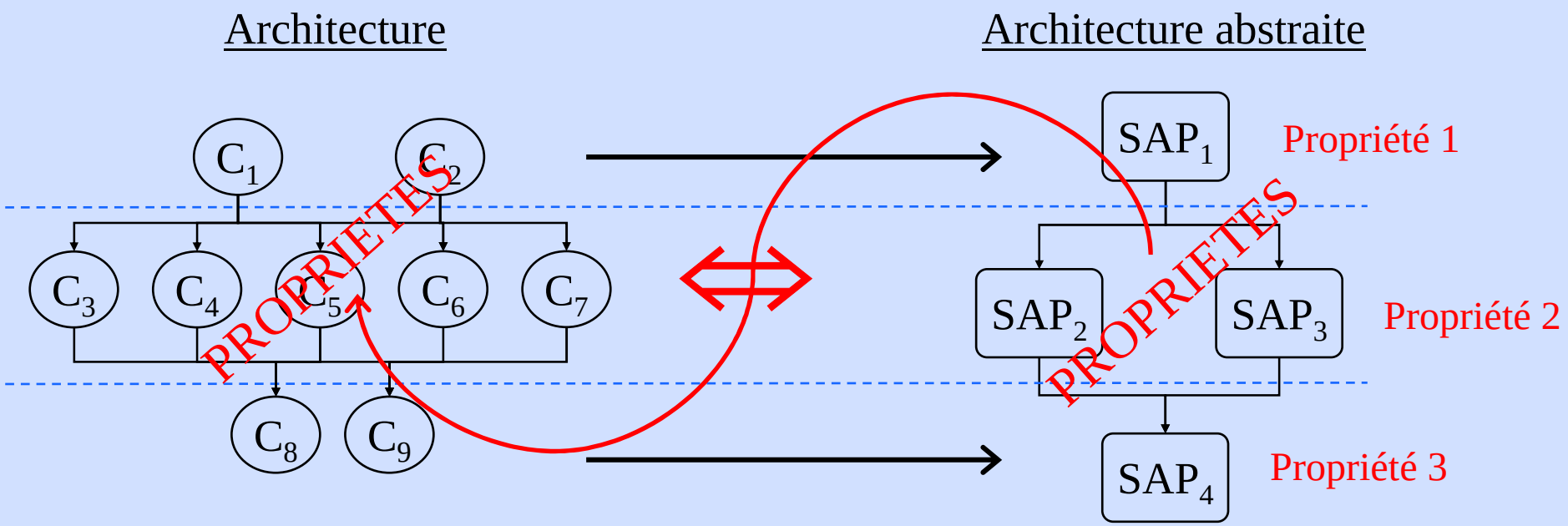
ELEC

→ Exemple d'architecture de sûreté (= *SAP*) commune :

redondance passive

ONERA

Contexte



→ Les propriétés démontrées sur M_A doivent être valides sur M_C

$$M_A \overset{\mathcal{R}}{\leftrightarrow} M_C$$

ONERA

Contexte

Objectifs des *SAP*

→ Capitaliser les solutions actuelles de sûreté, récurrentes et matures (gain de temps en conception)

→ Les *SAP* possèdent des propriétés pré-prouvées (analyse compositionnelle du système plus rapide), conception d'architectures directement en fonction des exigences

→ Vue du système plus macroscopique (1 *SAP* = { composants }), vue fonctionnelle synthétique du système

→ Permettre d'exhiber les propriétés satisfaites par le modèle et de comparer ces propriétés aux exigences

Contexte

Pourquoi Altarica ?

Le langage [Arnold et al., 2000]

→ formel

→ adapté à la SdF

→ capable de traiter les gros systèmes car il permet de réaliser des modèles :

- ► hiérarchiques
- ► compositionnels

Les outils

→ Editeur graphique

→ Outils de validation

Plan

1. Contexte

2. Présentation des *SAP*

3. Exemple d'utilisation

4. Conclusion

Présentation des *SAP*

SAP = vue abstraite du modèle

Caractérisation d'un *SAP*

→ Définition du contenu utile du point de vue de la SdF

→ Étude de leurs propriétés (hypothèses + architecture = propriétés pré-validées)

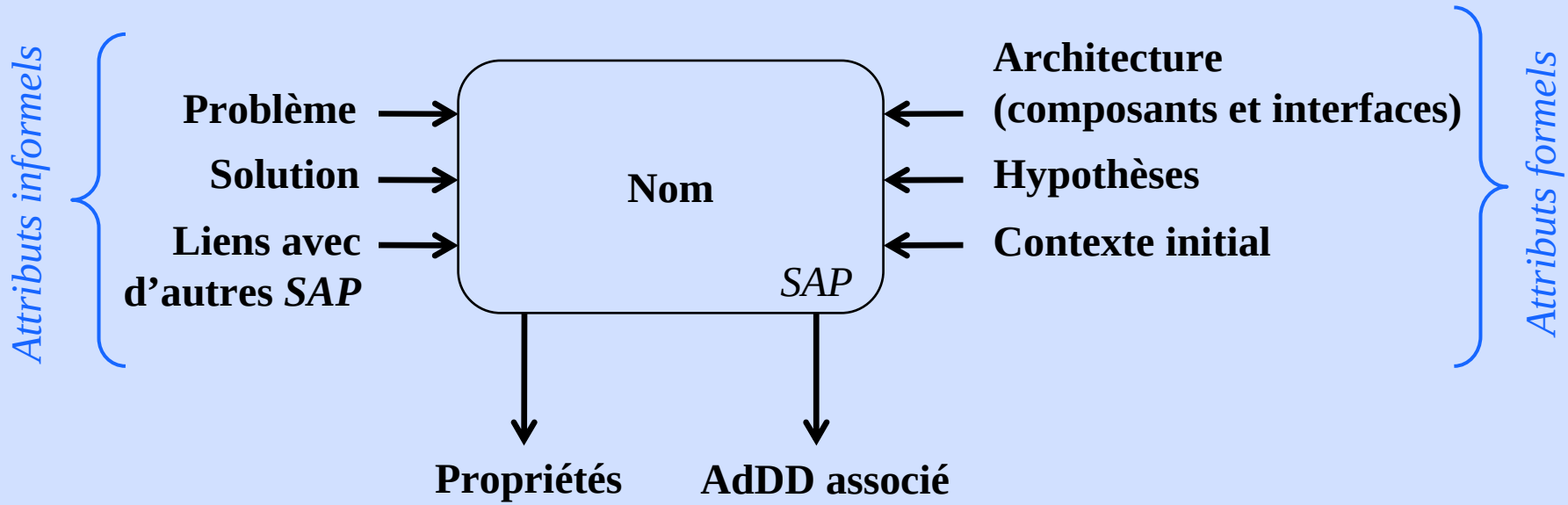
→ Étude de leur combinaison et de la combinaison de leurs propriétés

Exploitation de ces modèles généraux

→ Validation de ces modèles généraux par rapport à une architecture réelle

Présentation des *SAP*

Attributs



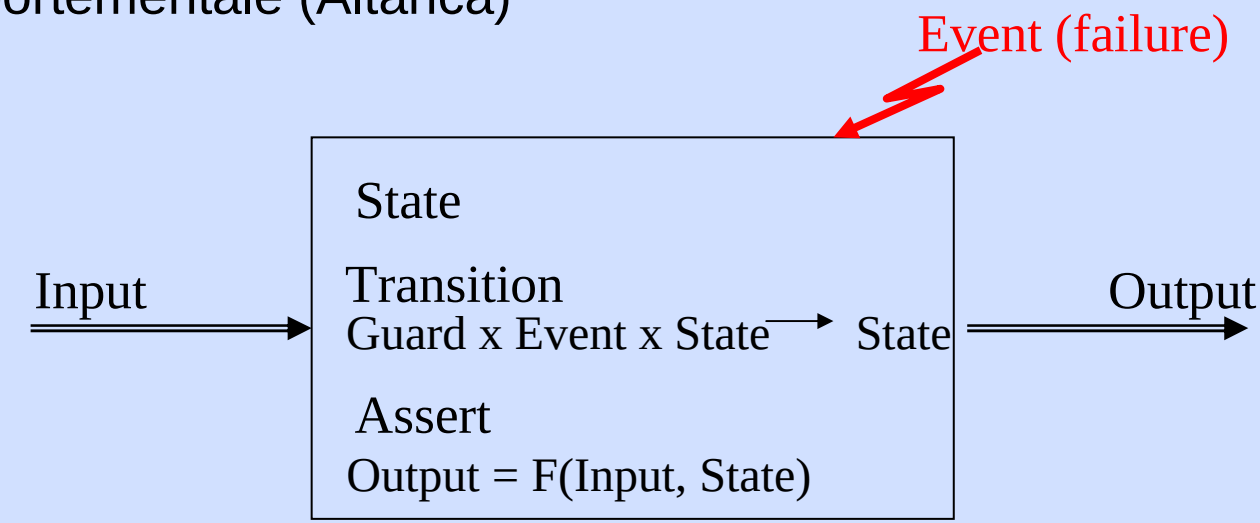
Présentation des *SAP*

Formalisation

→ déclarative (propriétés)

Si P_1, … , P_n alors P_m

→ comportementale (Altarica)

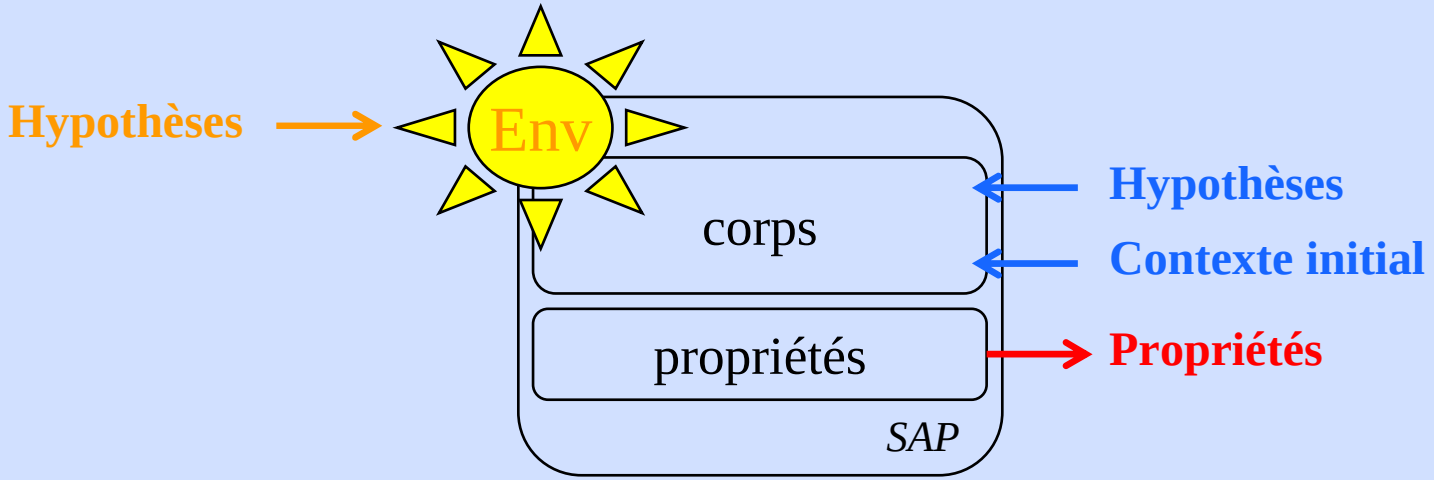


comportement ⇨ propriété

Présentation des *SAP*

Validation : preuve modulaire

→ Distinction entre hypothèses requises et internes



→ Intérêts

- ► Explicitation des exigences importantes pour la preuve
- ► Allocation des responsabilités

Présentation des *SAP*

Exemple de *SAP* (formalisation déclarative) : redondance passive $C_1 \hookrightarrow C_2$

Nom : COld Spare SAP (cossap)

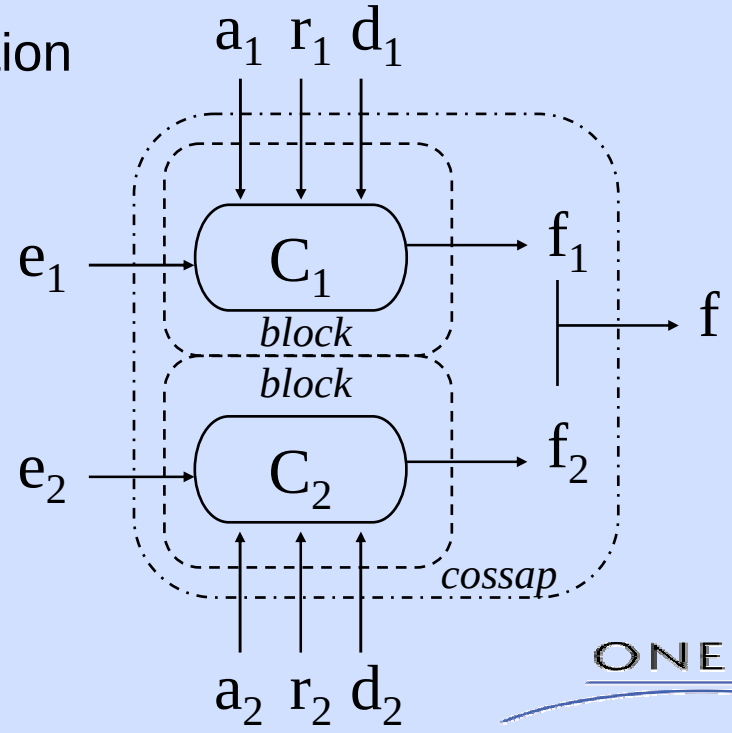
Problème : Tolérance à la faute d'une fonction

Solution : Dupliquer la fonction

Architecture : 2 composants C_1 et C_2

interfaces : a : activation

r : ...

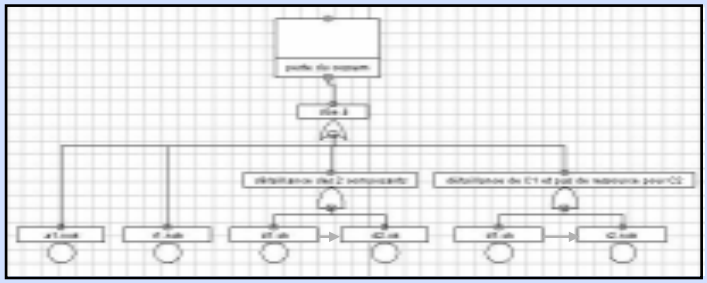


ONERA

Présentation des *SAP*

Contexte initial : $e_i = 1, a_1 = 1, a_2 = 0, r_i = 1, d_i = 0$

Sous-arbre dynamique associé :



Hypothèses :

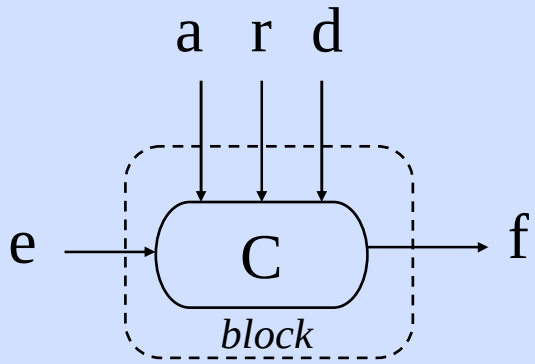
ressource $G(a_i \wedge \neg d_i \Rightarrow r_i)$

fonction
$$\begin{cases} G(f_1 \vee f_2 \Leftrightarrow f) \\ G(e_i \wedge a_i \wedge r_i \wedge \neg d_i \Leftrightarrow f_i) \end{cases}$$

activation
$$\begin{cases} G(\neg d_1 \Rightarrow a_1) \\ G(d_1 \Rightarrow X a_2) \end{cases}$$

défaillance $G(\neg(d_1 \wedge d_2))$

Liens avec d'autres SAP : SAP Block



Propriété : $G\,F(f)$

Présentation des *SAP*

Exemple de *SAP* (formalisation comportementale) : redondance active

trans

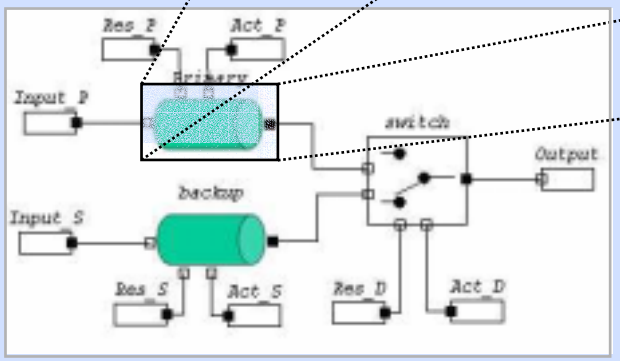
state_ = ok |- failure -> state_ := nok;

assert

$e_i \wedge a_i \wedge r_i \wedge \neg d_i$ $\qquad$ f_i

*(if (input **and** act **and** res **and** (state_ = ok)) **then** output = true **else** output = false);*

…



+

hypothèses externes + propriétés
(automate Altarica / formules)

ONERA

Plan

1. Contexte

2. Présentation des *SAP*

3. Exemple d’utilisation

4. Conclusion

Exemple d'utilisation

→ **Création d'architectures à partir de bibliothèques de *SAP***

→ **Validation d'architectures par substitution avec des *SAP***

► Recherche des *SAP* dans l'architecture réelle
i.e. redondances doubles, triples, … (analyse fonctionnelle du système)

► Vérification de la validité des hypothèses du *SAP* sur l'architecture réelle
e.g.

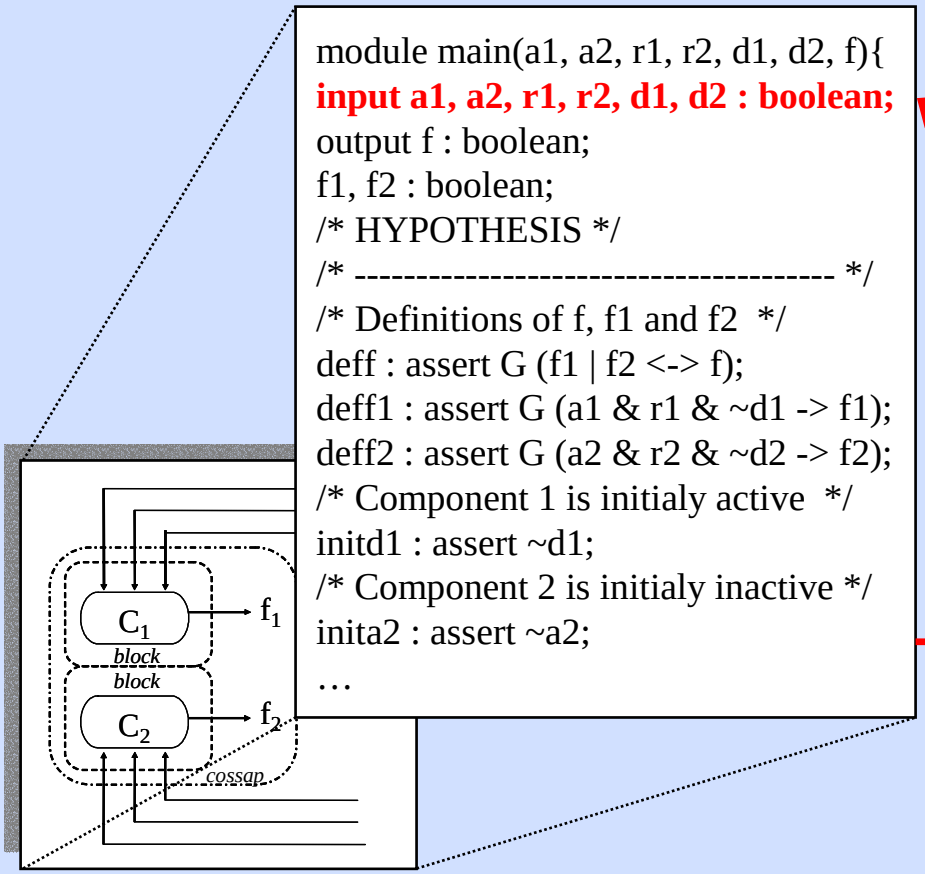
- Le *SAP* prend en entrée l'architecture réelle
- Traduction globale vers SMV (passerelle *Alta2smv*)
- Vérification des hypothèses

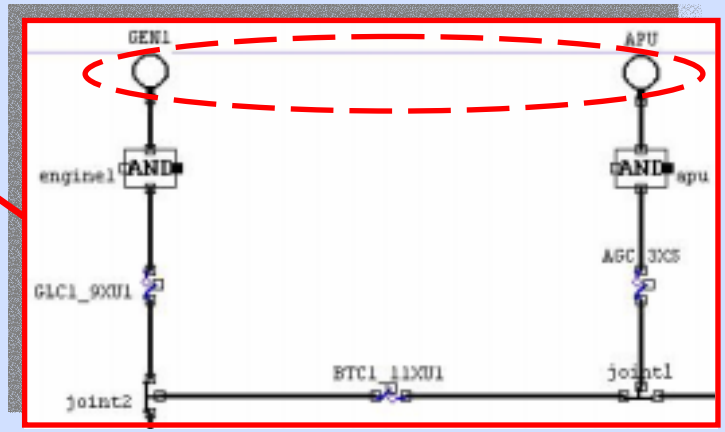
► Substitution de l'architecture par le *SAP*

Exemple d'utilisation

Formalisation déclarative : propriétés


```
module main(a1, a2, r1, r2, d1, d2, f){
input a1, a2, r1, r2, d1, d2 : boolean;
output f : boolean;
f1, f2 : boolean;
/* HYPOTHESIS */
/* ------------------------------------ */
/* Definitions of f, f1 and f2  */
deff : assert G (f1 | f2 <-> f);
deff1 : assert G (a1 & r1 & ~d1 -> f1);
deff2 : assert G (a2 & r2 & ~d2 -> f2);
/* Component 1 is initialy active  */
initd1 : assert ~d1;
/* Component 2 is initialy inactive */
inita2 : assert ~a2;
…
```

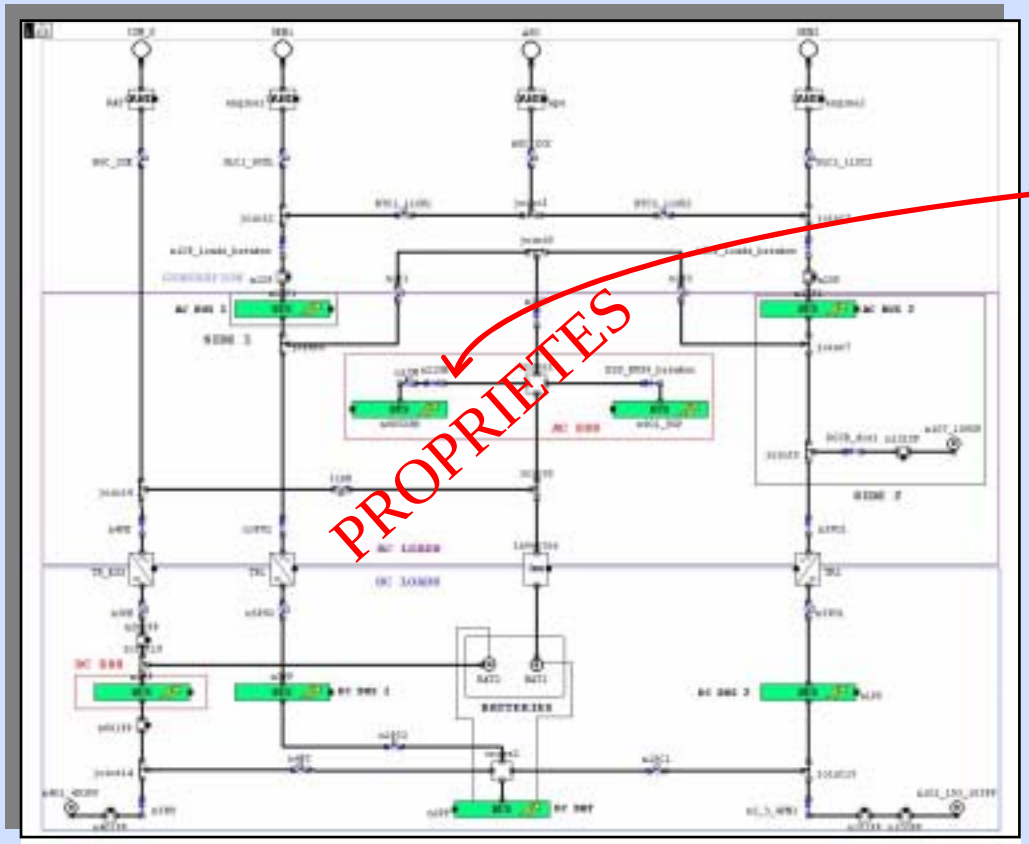


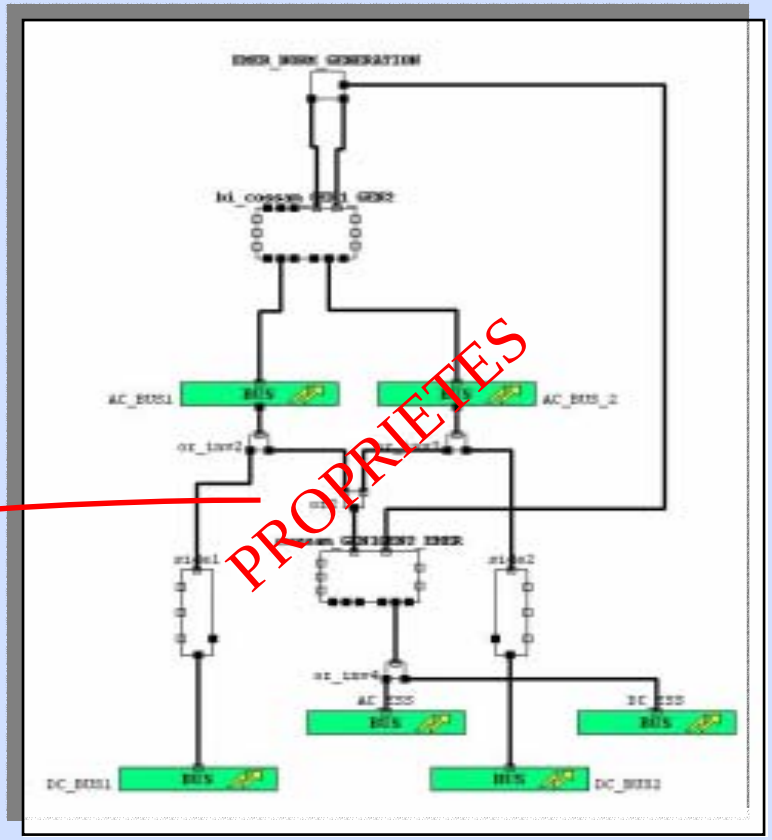


archi_satisfies_sap_properties : true

ONERA

Exemple d'utilisation





ONERA

Plan

1. Contexte

2. Présentation des *SAP*

3. Exemple d’utilisation

4. Conclusion

Conclusion

Intérêts

→ Vue synthétique de l'architecture

→ Vérification de la tenue des exigences

→ Test rapide de nouvelles architectures

Perspectives

→ Conception

- ► Assister le processus de proposition d'optimisations d'architectures

→ Validation

- ► Définition d'une relation de raffinement entre *SAP* et architecture réelle
- ► Automatisation (*SAP matching*)
- ► Evaluation des performances

MERCI

ONERA