AIDE A LA CONCEPTION DE NOYAU APPLICATIF POUR LE LOGICIEL SYNDEX

QUENTIN QUADRAT

Stage de fin de tronc commun 1er Septembre au 31 Décembre 2005

Supervisé par Yves Sorel, Directeur de Recherche à l'INRIA





- 1. Sommaire
- 2. INRIA. AOSTE
- 3. Probl. STR
- 4. Méthodo, AAA
- 5. SynDEx
- 6. Graphe...
- 7. Graphe...
- 3. Diagramme...
- 9 Exécutif
- 10. Noyau . . .
- 11. Génération . . .
- 12. Premier travail
- 13. Editeur de code
- 14. Exemple
- 5. Deuxième . . .
- 6. Diag. avant...
- 17. Diag. après...
- 18. Conclusion
- 19. FIN



1. Sommaire

- Structure d'accueil : INRIA, le projet AOSTE.
- Problématiques des systèmes temps réel embarqués.
- Méthodologie Adéquation Algorithme Architecture.
- Les exécutifs et les noyaux d'exécutifs.
- Le logiciel SynDEx.
- Présentation du travail : l'Éditeur de Code, création d'une application Automatique.

2.	INRIA	A, AO.	57	
3.	Probl	. STR	?	
4.	Méth	odo. /	4 <i>A</i>	
5.	SynD	Ex		
6.	Graph	ne		
7.	Graph	ie		
8.	Diagr	amme	٠	
9.	Exécu	ıtif		
10.	Noya	au		
11.	Géne	ératio	n .	
12.	Pren	nier tı	av	
13.	Edit	eur de	C	
14.	Exer	nple		
15.	Deux	xième		
16.	Diag	r. ava	nt	
17.	Diag	r. apre	ès .	
18.	Con	clusio	7	
19.	FIN			
Page 2 sur 20				
Pre	écéd.	Suiv	/ar	
44	•	•	•	

2. INRIA, AOSTE

2.1. INRIA.

- Institut National de Recherche en Informatique et en Automatique (créé en 1967)
- Il est placé sous la double tutelle du ministre chargé de la Recherche et de l'Industrie.

2.2. Le projet AOSTE.

- Analysis Optimisation of Systems with real-Time and Embedded constraints,
- Méthodologie AAA (Adéquation Algorithme Architecture),
- SynDEx : logiciel de CAO pour l'aide à l'implantation de systèmes distribués temps réel embarqués.

1. Sommaire 3. Probl. STR Méthodo, AAA 5. SynDEx Graphe . . . Graphe . . . Diagramme . . . Exécutif Noyau . . . 10. Génération . . . 11. 12. Premier travail

13. Editeur de code

Deuxième . . .

Page 3 sur 20

Suivant

Diag. avant...

Diag. après...

Conclusion

14. Exemple

19. FIN

Précéd.

3. Probl. STR

Le temps est la problématique des systèmes temps réel embarqués :

- Une opération doit pouvoir s'exécuter dans un temps prévu à l'avance.
- Réagir trop tard peut conduire à des conséquences catastrophiques pour le système lui-même ou son environnement.
- Il faut donc recourir à des calculateurs à architecture parallèle ou à des circuits intégrés spécialisés pour augmenter la puissance de calcul.

DONC, Il faut pouvoir affecter les tâches sur les différents processeurs et assurer la synchronisation de leur exécution.

1. Sommaire INRIA. AOSTE 4. Méthodo, AAA SynDEx Graphe . . . Graphe . . . Diagramme . . . Exécutif 10. Noyau . . . 11. Génération . . . 12. Premier travail 13. Editeur de code 14. Exemple Deuxième . . . Diag. avant... Diag. après... Conclusion 19. FIN Page 4 sur 20 Précéd. Suivant



- **EXÉCUTIF** Algorithme : graphe d'ordonnancement des tâches.
- Architecture : graphe des ressources et des moyens de communication.
- Exécutifs : affectation temporelle des tâches aux ressources.
- Adéquation : générer l'exécutif.

- 1. Sommaire
- 2. INRIA, AOSTE
- 3. Probl. STR

4 Méthodo A

- 5. SynDEx
- 6. Graphe...
- 7. Graphe...
- 3. Diagramme . . .
- 9. Exécutif
- 10. Noyau...
- 11. Génération...
- 12. Premier travail
- 13. Editeur de code
- 14. Exemple
- 15. Deuxième . . .
- 16. Diag. avant...
- 17. Diag. après...
- 18. Conclusion
- 19. FIN



5. SYNDEX

- Met en oeuvre la méthodologie AAA.
- Entrée : graphe algorithme et graphe architecture.
- Sortie : les exécutifs distribués temps réel, sans interblocage et statiques écrits en macro-code m4.
- Visualise la prédiction des performances temps réel pour le dimensionnement de l'architecture.
- Utilise des heuristiques pour la distribution et l'ordonnancement de l'algorithme d'application sur l'architecture.
- Des noyaux d'exécutifs génériques sont fournis pour : SHARC, i80C196, TMS320C40, i80386, MC68332 et stations de travail Unix ou Linux.

1.	Sommaire
2.	INRIA, AOSTE
3.	Probl. STR
4.	Méthodo. AAA
5.	SynDEx
6.	Graphe
7.	Graphe
8.	Diagramme
9.	Exécutif
10.	Noyau
11.	Génération
12.	Premier trava
13.	Editeur de cod
14.	Exemple
15.	Deuxième
16.	Diag. avant
17.	Diag. après
18.	Conclusion

19 FIN

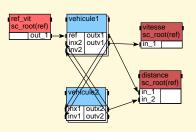
Précéd.

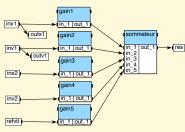
Page 6 sur 20

Suivant

6. Graphe algorithme

- Un graphe hiérarchique définit l'algorithme.
- Les noeuds sont les tâches (opérations) à réaliser et les arcs définissent les relations de précédence.



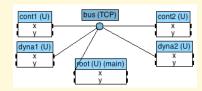


- 1. Sommaire
- 2. INRIA. AOSTE
- 3. Probl. STR
- 4. Méthodo. AAA
- 5. SynDEx
- 6. Graphe.
- 7. Graphe...
- 3. Diagramme...
- 9. Exécutif
- 10. Noyau . . .
- 11. Génération . . .
- 12. Premier travail
- 13. Editeur de code
- 14. Exemple
- 5. Deuxième . . .
- ... Diag. avant...
- Diag. après...
 Conclusion
- 16. CONCIUSION
- 19. FIN



7. Graphe architecture

- Un graphe non hiérarchique définit l'architecture.
- Les noeuds sont les ressources (soit des opérateurs soit des médias de communication).
- Les arcs les communications entre les ressources.



- 1. Sommaire
- 2. INRIA. AOSTE
 - . Probl. STR
- 4. Méthodo. AAA
- 5. SynDEx
- 6. Graphe...
- 7. Graphe..
 - 3. Diagramme...
- 9. Exécutif
- 10. Noyau . . .
- 11. Génération...
- 12. Premier travail
- 13. Editeur de code
- 14. Exemple
 - 5. Deuxième . . .
 - 6. Diag. avant...
- 7. Diag. après...
- 18. Conclusion
- 19. FIN



8. Diagramme temporel

La synchronisation des exécutions des tâches peut être visualisée sous forme d'un diagramme temporel :

root	cont2	cont1	dyna2	bus	dyna1
ref_vit vitesse distance	gain4 gain3 gain2 gain1 sommateur	gain4 gain3 gain2 gain1 gain5 sommateur	pas sommateur puissancemoteur pas sommateur retard retard		pas sommateur puissancemoteur pas sommateur
					retard

- Colonnes : ressources disponibles.
- Contenu des colonnes : agenda des tâches à exécuter.
- Epaisseur des opérations : durée des opérations.

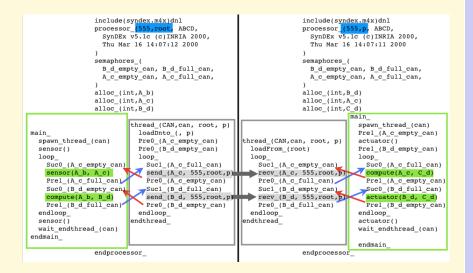
- 1. Sommaire
- 2. INRIA. AOSTE
- 3. Probl. STR
- 4. Méthodo, AAA
- 5. SynDEx
- 6. Graphe...
- 7. Graphe...

8. Diag

- 9. Exécutif
- 10. Noyau...
- 11. Génération...
- 12. Premier travail
- 13. Editeur de code
- 14. Exemple
- 15. Deuxième . . .
- 16. Diag. avant...
- 17. Diag. après...
- 18. Conclusion
- 19. FIN



9. Exécutif



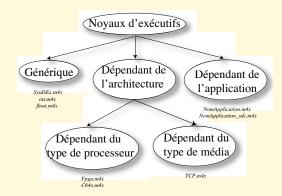
Les opérations et les synchronisations sont spécifées en macrocode m4.

- 1. Sommaire
- 2. INRIA. AOSTE
 - 3. Probl. STR
- 4. Méthodo, AAA
- 5. SynDEx
- 6. Graphe...
- 7. Graphe...
- 8. Diagramme...
- 9 Exécutif
- 10. Noyau . . .
- 11. Génération . . .
- 12. Premier travail
- 13. Editeur de code
- 14. Exemple
- 15. Deuxième . . .
- 16. Diag. avant..
- 17. Diag. après...
- 18. Conclusion
- 19. FIN



10. Noyau d'exécutif

Noyau d'exécutif : ensemble de définitions des différentes macros utilisées par les exécutifs, dans un langage haut niveau (C, ...) ou assembleur.



Travail du stage a concerné la génération de noyau applicatif.

- 1. Sommaire
- 2. INRIA. AOSTE
- 3. Probl. STR
- 4. Méthodo, AAA
 - 5. SynDEx
- 6. Graphe...
- 7. Graphe...
- 8. Diagramme . . .
- 9. Exécutif

10. Novau

- 1. Génération . . .
- 2. Premier travail
- 13. Editeur de code
- 14. Exemple
- 15. Deuxième . . .
- 17. Diag. après...
- 18. Conclusion
- 19. FIN



11. GÉNÉRATION DE L'EXÉCUTABLE

Macro-processeur M4 : Exécutifs \times Noyaux d'exécutifs \rightarrow Codes sources (C)

Compilateur (GCC) : Codes sources (C) \rightarrow Exécutables

- Sommaire
 INRIA, AOSTE
 Probl. STR
 Méthodo, AAA
- 5. SynDEx
- 6. Graphe...
- 7. Graphe...
- 8. Diagramme...
- 9. Exécutif
- 10. Noyau...

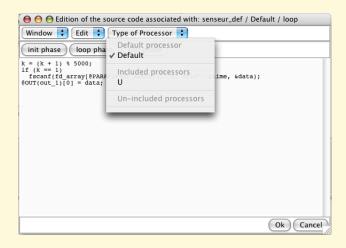
11. Génération...

- 12. Premier travail
- 13. Editeur de code
- 14. Exemple
- 15. Deuxième...
- 16. Diag. avant...
- 17. Diag. après...
- 18. Conclusion
- 19. FIN



12. Premier travail

Développer un éditeur de noyau applicatif Éditeur de Code dans l'IHM de SynDEx.



INRIA. AOSTE Probl. STR Méthodo, AAA SynDEx Graphe . . . Graphe . . . Diagramme . . . Exécutif Noyau . . . 10. Génération . . . Editeur de code Exemple Deuxième . . . Diag. avant... Diag. après... Conclusion 19. FIN Page 13 sur 20 Suivant Précéd.

1. Sommaire

13. Editeur de code

Avant le stage, un utilisateur devait spécifier manuellement le noyau applicatif en en terme de macros instructions m4.

Après le stage, un utilisateur édite le code de chaque opérations en langage naturel.

L'Éditeur de Code :

- Génére les macros m4 correspondantes.
- Automatise la traduction des noms de ports en arguments m4.
- Complète les noms de ports.
- Communique avec Emacs.

Pr	écéd. Suivant
Page 14 sur 20	
19.	FIN
18.	Conclusion
17.	Diag. après
16.	Diag. avant
15.	Deuxième
14.	Exemple
13.	Editeur de code
12.	Premier travail
11.	Génération
10.	Noyau
9.	Exécutif
8.	Diagramme
7.	Graphe
6.	Graphe
5.	SynDEx
4.	Méthodo. AAA
3.	Probl. STR
2.	INRIA, AOSTE
1.	Sommane

14. Exemple

• Le code que l'on devait écrire dans un fichier:

• s'écrit dans l'Éditeur de Code :

```
QOUT(out)[0] = QIN(in)[0] + QPARAM(P);
```

- 1. Sommaire
- 2. INRIA, AOSTE
- 3. Probl. STR
- 4. Méthodo. AAA
- 5. SynDEx
- 6. Graphe...
- 7. Graphe...
- 3. Diagramme . . .
- 9. Exécutif
- 10. Noyau...
- 11. Génération . . .
- 12. Premier travail
- 13. Editeur de code

14. Exemple

- 15. Deuxième...
- 6. Diag. avant...
- 17. Diag. après...
- 18. Conclusion
- 19. FIN



15. Deuxième travail

- Documenter l'Éditeur de Code.
- Créer une application d'automatique où une voiture suit une autre et maintient une distance fixe alors que la première choisit une vitesse.
- Apprentissage : introduction à l'automatique, logiciels Scilab et Scicos
- Simulation : Scicos et SynDEx.

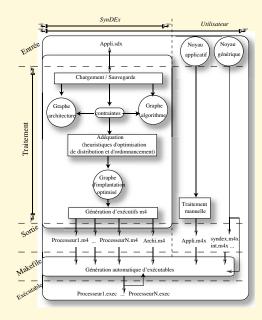
- 1. Sommaire
- 2. INRIA. AOSTE
 - B. Probl. STR
- 4. Méthodo. AAA
- 5. SynDEx
- 6. Graphe...
- 7. Graphe...
- 3. Diagramme . . .
- 9. Exécutif
- 10. Noyau...
- 11. Génération . . .
- 12. Premier travail
- 3. Editeur de code
- 14. Exemple

5. Deuxième . . .

- 16. Diag. avant...
- 7. Diag. après...
- 18. Conclusion
- 19. FIN



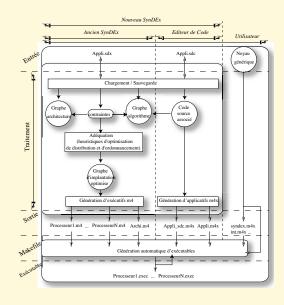
16. DIAG. AVANT LE STAGE



- 1. Sommaire
- 2. INRIA, AOSTE
- 3. Probl. STR
- 4. Méthodo. AAA
- 5. SynDEx
- 6. Graphe...
- 7. Graphe...
- 8. Diagramme...
- 9. Exécutif
- 10. Noyau...
- 11. Génération . . .
- 12. Premier travail
- 13. Editeur de code
- 14. Exemple
- 5. Deuxième . . .
- 16 51
- 17. Diag. après...
- 18. Conclusion
- 19. FIN



17. Diag. Après le stage



- 1. Sommaire
- 2. INRIA, AOSTE
 - B. Probl. STR
- 4. Méthodo. AAA
- 5. SynDEx
- 6. Graphe...
- 7. Graphe...
- 8. Diagramme...
- 9. Exécutif
- 10. Noyau...
- 11. Génération . . .
- 12. Premier travail
 - 3. Editeur de code
- 4. Exemple
- 5. Deuxième . . .
- 16. Diag. avant...
- 18. Conclusion
- 19. FIN

Page 18 sur 20
Précéd. Suivant







18. CONCLUSION

Appréciations des utilisateurs de l'Éditeur de Code. Ce travail était une demande par l'un (MBDA) des principaux utilisateur industriel de SynDEx dont les commentaires après utilisation sont positifs.

Expérience acquise. Ce stage m'a permis de :

- Découvrir les problèmes d'affectation pour réaliser des systèmes temps réel.
- Découvrir des notions en automatique.
- De manipuler les logiciels SynDEx, Scilab, Scicos,
- De progresser dans la connaissance du langage OCaml.

- 1. Sommaire INRIA. AOSTE Probl. STR Méthodo, AAA SynDEx Graphe . . . Graphe... Diagramme . . . Exécutif Noyau . . . 10. Génération . . . 11. Premier travail 12. 13. Editeur de code 14. Exemple Deuxième . . . Diag. avant...
 - Diag. après... 19. FIN Page 19 sur 20 Précéd. Suivant

19. FIN

Questions?

- 1. Sommaire INRIA, AOSTE Probl. STR 4. Méthodo. AAA 5. SynDEx Graphe . . . Graphe . . . Diagramme . . . Exécutif 10. Noyau . . . Génération . . . 11. 12. Premier travail 13. Editeur de code
- 14. Exemple
- 15. Deuxième . . .
 - 6. Diag. avant...
- 17. Diag. après...
- 18. Conclusion

19. FIN

Page 20 sur 20
Précéd. Suivant