Développement, amélioration et intégration d'outils de génération de code

pour une plateforme de prototypage de fonctions de contrôle moteur.

Mathieu SOUM
Université Paul Sabatier
Master 2 – Développement Logiciel

Stage réalisé chez Aboard Engineering
Maître de stage : Sébastien RICHE
Tutrice universitaire : Isabelle FERRANÉ

Année universitaire 2014 - 2015

CONTEXTE

OBJECTIFS

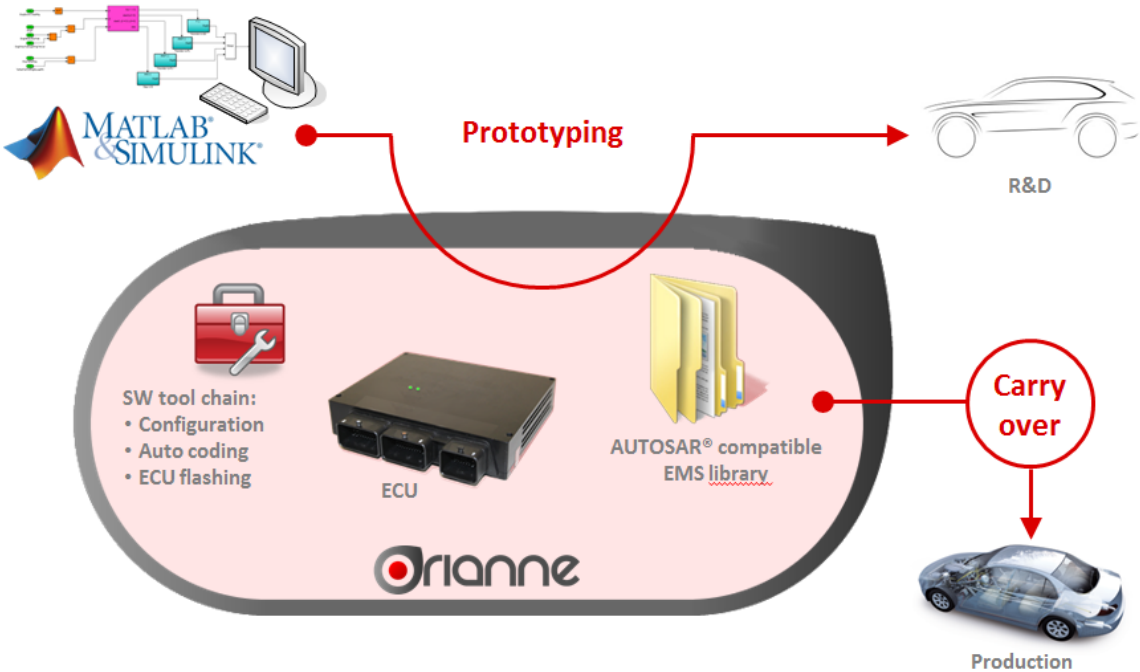
TRAVAIL RÉALISÉ

BILAN

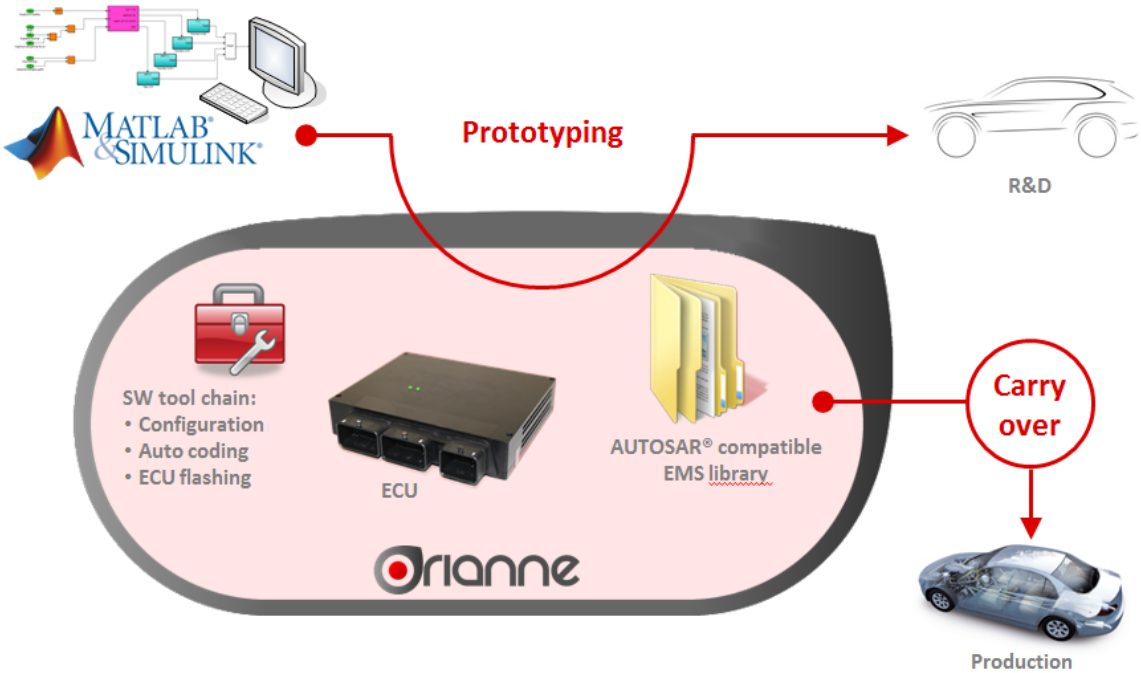
Automobile, Aéronautique, Marine, Loisir, Industriel de la R&D à la série

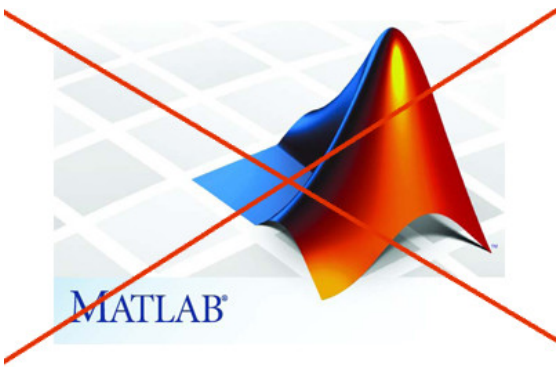
Domaines	Connaissances & expérience
Moteurs thermiques (essence , Diesel)	• Combustion • Stratégies de contrôle moteur, incl. dépollution • Technologie des capteurs et actionneurs • Problématiques de production série (diagnostic, adaptatif…) • Mise au point moteur / véhicule
Machines électriques	• Modélisation et simulation électronique de puissance et machine électrique • Commande vectorielle • Pilotage de machines électriques • Mise au point et calibration
Hybrides	• Contrôle & supervision d'énergie
Outils	• Développement d'outils génériques et spécifiques client • Instrumentation et essais pour tous les domaines

Orianne®


MATLAB® & SIMULINK®
Prototyping
R&D
SW tool chain:
• Configuration
• Auto coding
• ECU flashing
ECU
AUTOSAR® compatible
EMS library
Carry over
Orianne
Production

Orianne®



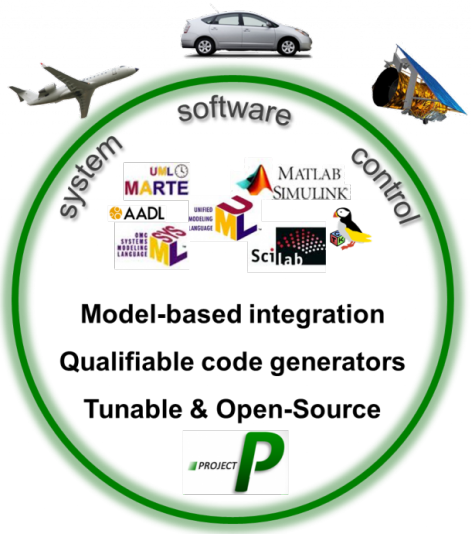


⟹

QGen

QGen

Générateur de code embarqué C/Ada depuis Matlab Simulink.



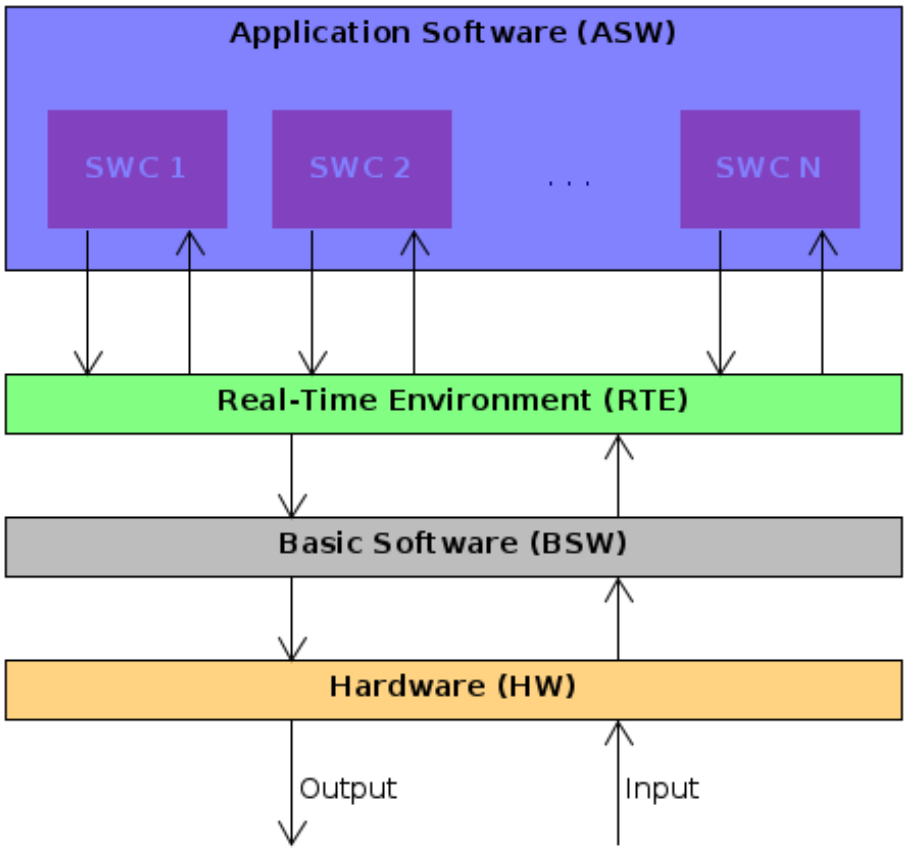
AdaCore
The GNAT Pro Company

Avantages

- Prix
- Maîtrise

Orianne®

« coopération sur les standards,
concurrence sur la mise en œuvre »



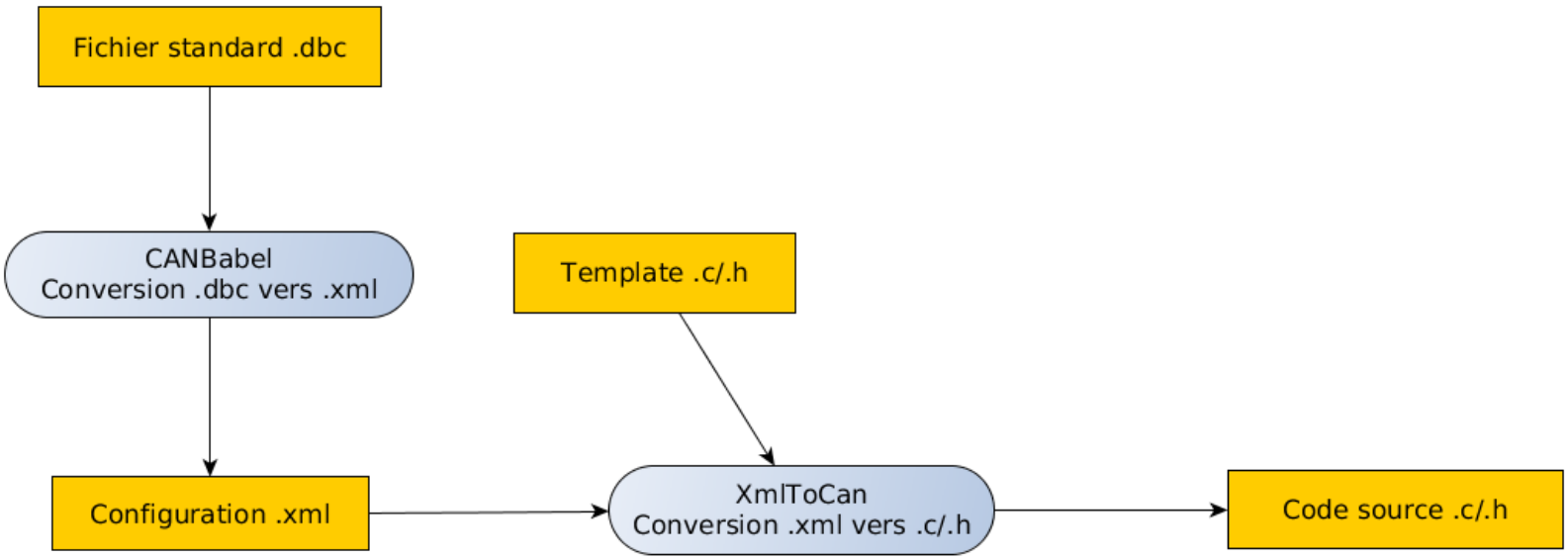
ASW
L'application issue du code généré.
Contient la glue ansi que la gestion
de la communication CAN

RTE
La couche de laison entre le BSW et l'ASW.
Lie également les SoftWare Components
(SWC) entre eux.

BSW
Le microgiciel embarqué dans le calculateur.
Ce sont les pilotes et API pour utiliser les
entrées/sorties de la couche HW.

HW
Le matériel du calculateur

CONTROLLER AREA NETWORK



Processus de convertion des fichiers .dbc vers du code source C

OBJECTIFS

QGen

- Analyse du code
- Identification des problèmes
- Collaboration avec AdaCore

Plateforme Orianne

- Génération CAN fonctionnelle et viable
- Intégration à l'IHM Orianne
- Conception des outils manquants

QGEN – ANALYSE

Analyse statique (OCLint)

- Complexité
- Propreté du code

Analyse à l'execution

- Espace de stockage
- Mémoire
- Temps d'execution

QGEN – PREMIERS RÉSULTATS

Analyse statique : codes équivalents

Analyse à l'exécution :

- Moins performant
- Plus volumineux

Facteur 2

QGEN – HYPOTHÈSES D'AMÉLIORATIONS

- Extrapolation et interpolation
- Variables intermédiaires
- Déclaration des entrée/sorties

QGEN – DERNIÈRES MÉTRIQUES

Résultats équivalents entre QGen et Matlab RTW-EC.

- Temps d'execution : $800\mu s \Rightarrow 400\mu s$
- Espace de stockage grandement diminué

Résultats très concluants
Première version majeure annoncée

ORIANNE – DÉCOUVERTE DE LA PLATEFORME

Architecture : Composants OSGi

Non pertinent

- Manque de modularité
- Redondance
- Linéarité

→ Maven

ORIANNE – GÉNÉRATEUR CAN

Trop de mémoire utilisée ⇒ code inutilisable

Étude de l'existant :

- Redondance
- Linéarité
- Complexité

Factorisation, abstraction ⇒ Clarté, maintenabilité

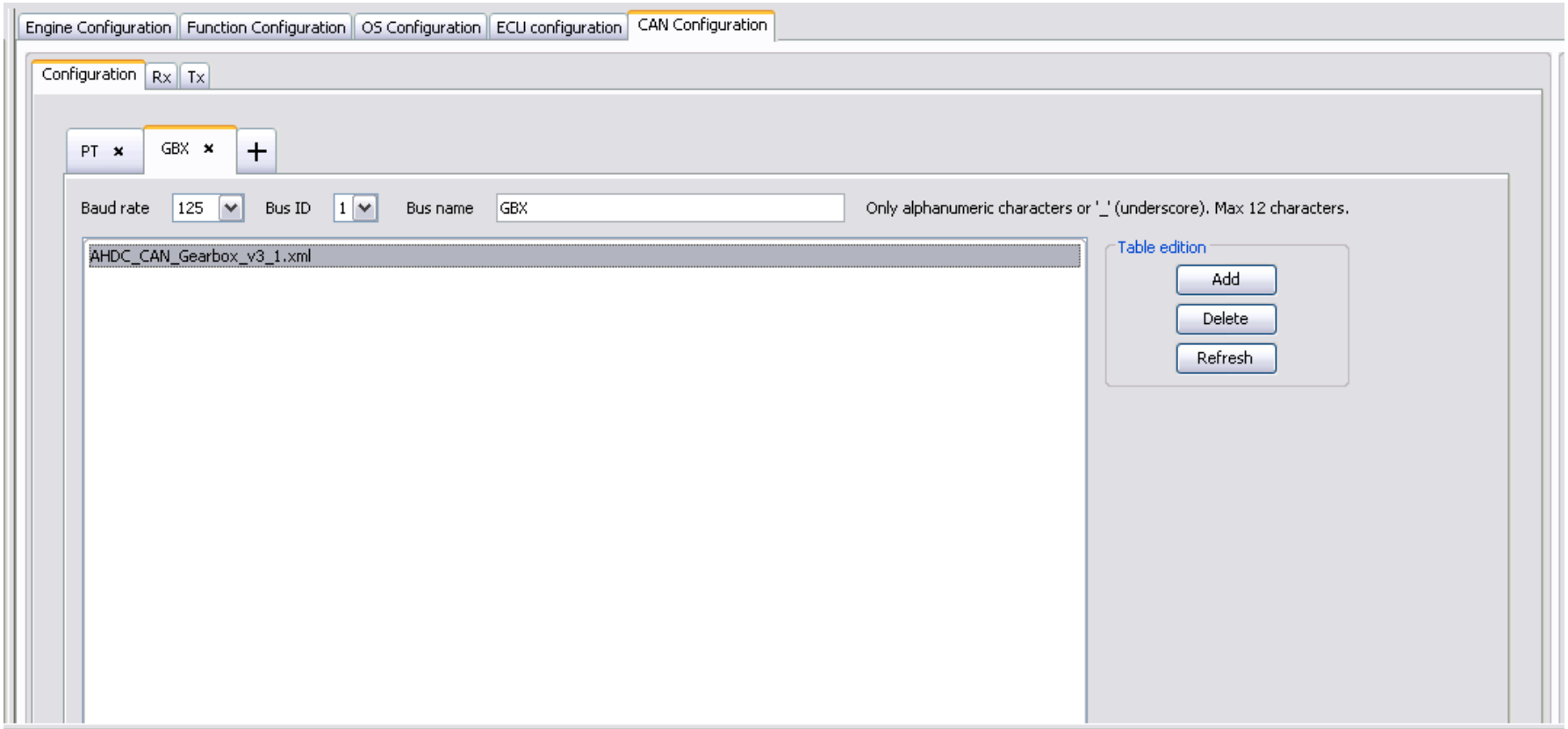
ORIANNE – GÉNÉRATEUR CAN

Adaptation des templates

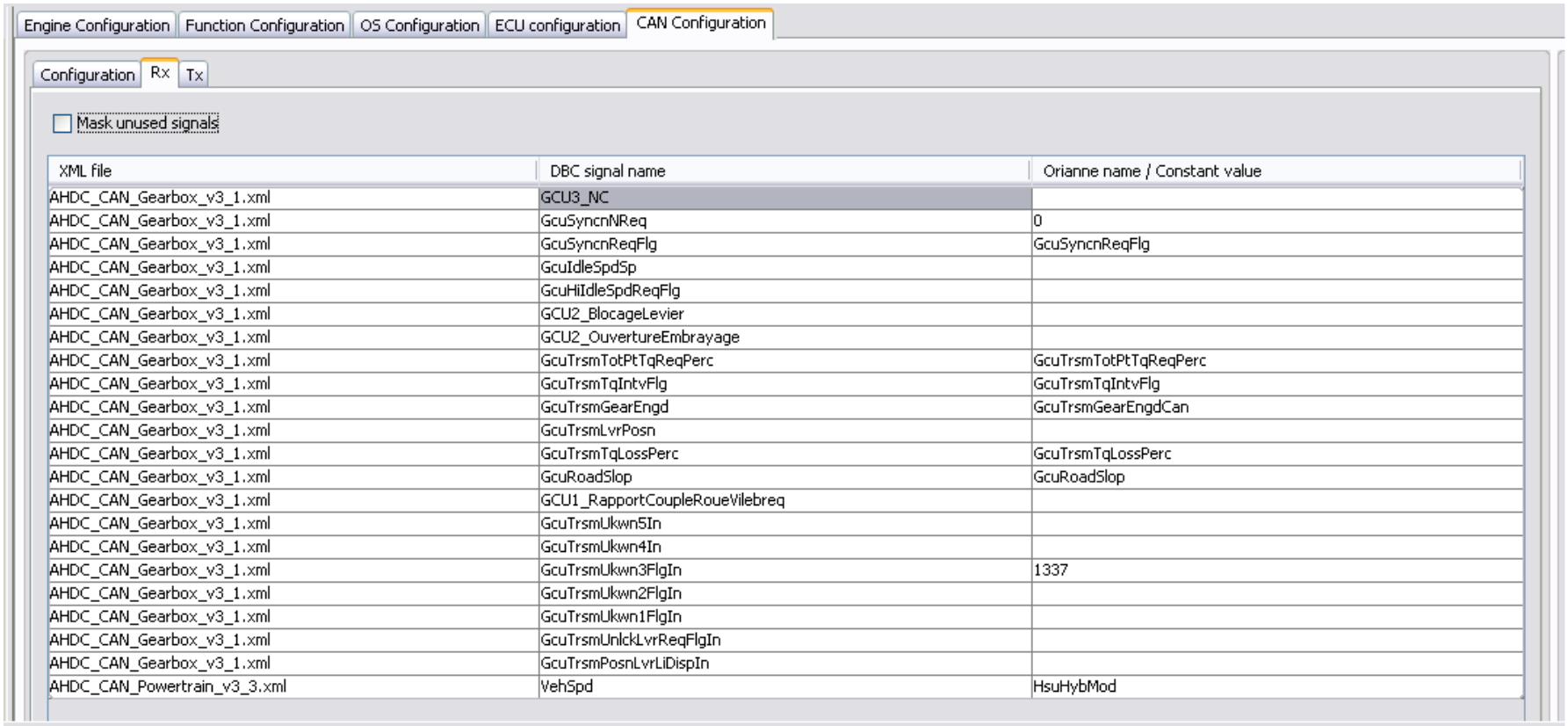
- Suppression des données inutilisées
- Passage en constante des configurations
- Factorisation du code généré

```
39  const SGN_CFG signals_$IDTRAME-0x$_$suffix$[$signal_count$] = {/$decl_sgn_config$/
40      {   /* Signal $NomSignal$ */
41          &$NomSignal$,               // ASWVariable
42          $typeSignal$,               // ASWVariable_type
43          $StartBit$,                 // start_bit // Donnee issue de la DBC
44          $SignalLength$,             // lenght // Donnee issue de la DBC
45          $SignedValueOfSignal$,      // signed_value // Donnee issue de la DBC
46          $EndianessOfSignal$,        // endianess // Donnee issue de la DBC
47          $SignalFactor$,             // factor // Donnee issue de la DBC
48          $SignalOffset$,             // offset // Donnee issue de la DBC
49          $SignalInitValue$,          // init_value
50          $StartBitModulo$,           // StartBit_modulo8
51          $StartOctet$,               // StartOctet
52          $IDTRAME$,                  // id
53          $OffsetCalcul$              // OffsetSignedValue
54      }/$decl_sgn_config$/
55  };
```

ORIANNE – INTÉGRATION À L'IHM



ORIANNE – INTÉGRATION À L'IHM



XML file	DBC signal name	Orianne name / Constant value
AHDC_CAN_Gearbox_v3_1.xml	GCU3_NC	
AHDC_CAN_Gearbox_v3_1.xml	GcuSyncnNReq	0
AHDC_CAN_Gearbox_v3_1.xml	GcuSyncnReqFlg	GcuSyncnReqFlg
AHDC_CAN_Gearbox_v3_1.xml	GcuIdleSpdSp	
AHDC_CAN_Gearbox_v3_1.xml	GcuHiIdleSpdReqFlg	
AHDC_CAN_Gearbox_v3_1.xml	GCU2_BlocageLevier	
AHDC_CAN_Gearbox_v3_1.xml	GCU2_OuvertureEmbrayage	
AHDC_CAN_Gearbox_v3_1.xml	GcuTrsmTotPtTqReqPerc	GcuTrsmTotPtTqReqPerc
AHDC_CAN_Gearbox_v3_1.xml	GcuTrsmTqIntvFlg	GcuTrsmTqIntvFlg
AHDC_CAN_Gearbox_v3_1.xml	GcuTrsmGearEngd	GcuTrsmGearEngdCan
AHDC_CAN_Gearbox_v3_1.xml	GcuTrsmLvrPosn	
AHDC_CAN_Gearbox_v3_1.xml	GcuTrsmTqLossPerc	GcuTrsmTqLossPerc
AHDC_CAN_Gearbox_v3_1.xml	GcuRoadSlop	GcuRoadSlop
AHDC_CAN_Gearbox_v3_1.xml	GCU1_RapportCoupleRoueVilebreq	
AHDC_CAN_Gearbox_v3_1.xml	GcuTrsmUkwn5In	
AHDC_CAN_Gearbox_v3_1.xml	GcuTrsmUkwn4In	
AHDC_CAN_Gearbox_v3_1.xml	GcuTrsmUkwn3FlgIn	1337
AHDC_CAN_Gearbox_v3_1.xml	GcuTrsmUkwn2FlgIn	
AHDC_CAN_Gearbox_v3_1.xml	GcuTrsmUkwn1FlgIn	
AHDC_CAN_Gearbox_v3_1.xml	GcuTrsmUnlckLvrReqFlgIn	
AHDC_CAN_Gearbox_v3_1.xml	GcuTrsmPosnLvrLiDispIn	
AHDC_CAN_Powertrain_v3_3.xml	VehSpd	HsuHybMod

ORIANNE – BILAN CAN

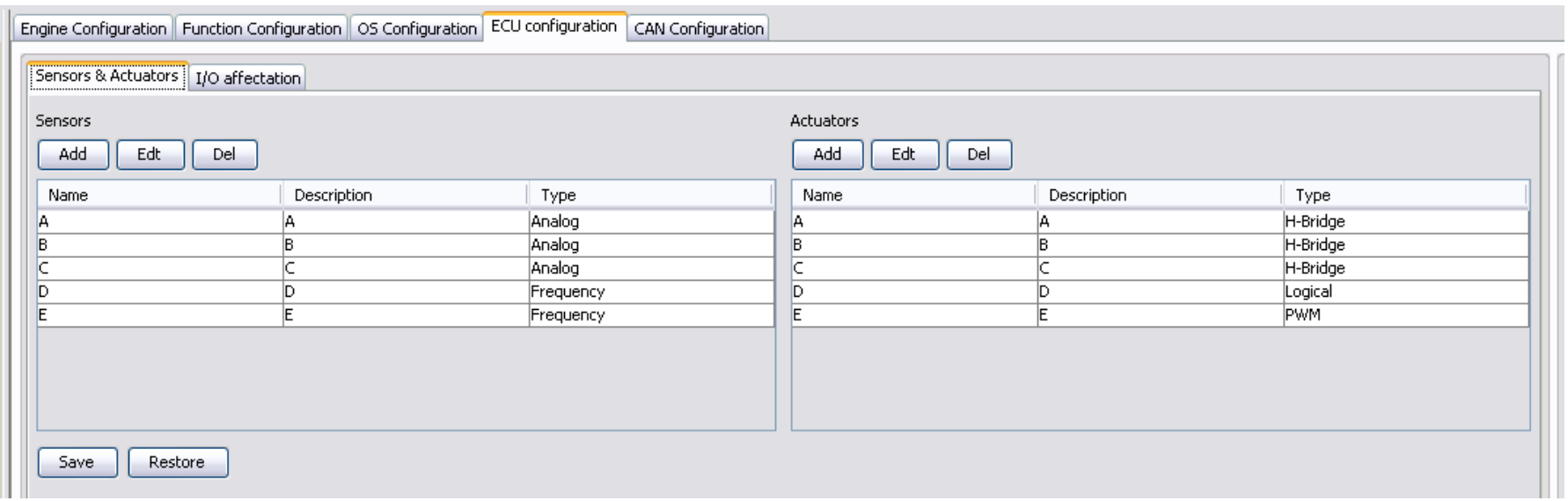
Objectif de performance atteint

Facteur 7

Code généré testé et validé
IHM de configuration complète

ORIANNE – PARTIE RTE

Configuration complète via l'IHM
Développement du générateur



Contexte

Objectifs

Travail réalisé

Bilan

BILAN

Expérience professionnelle

- Découverte du domaine automobile
- Expérience en alternance
- Atteinte des objectifs

Apports personnels

- Intérêt pour l'informatique embarquée
- Approfondissement technique
- Confort de mes compétences en anglais

Développement, amélioration et intégration d'outils de génération de code

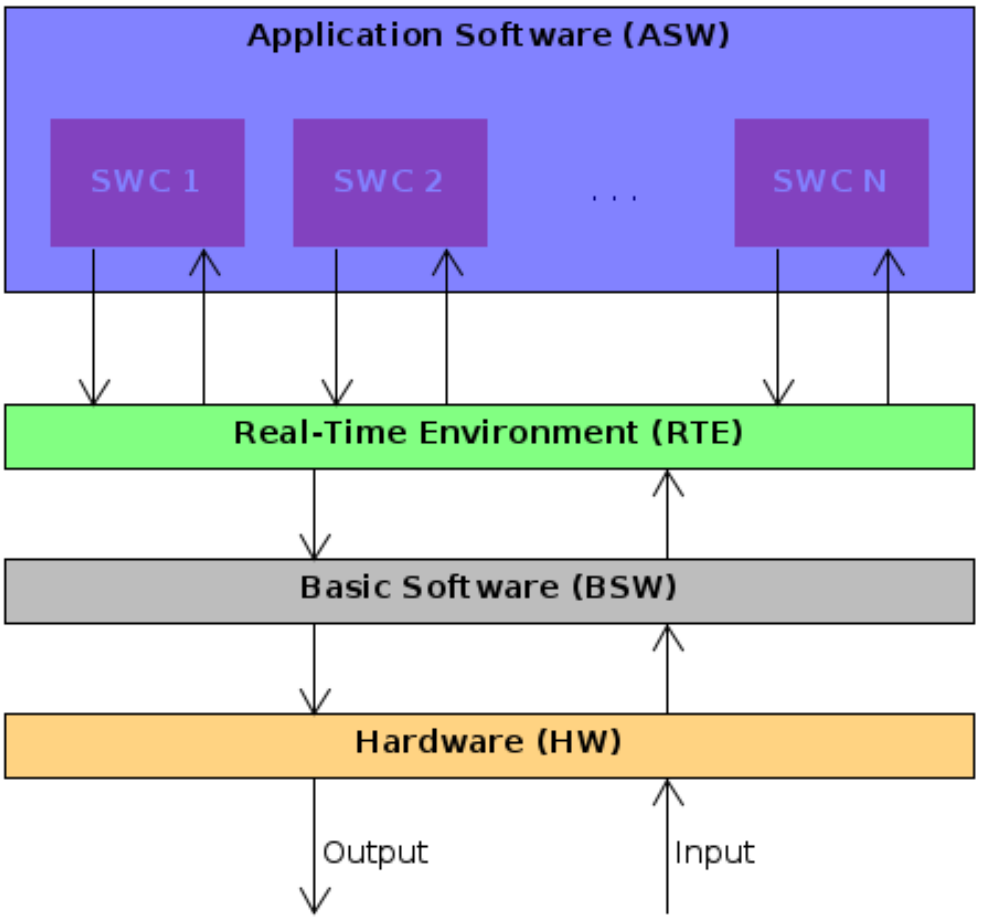
pour une plateforme de prototypage de fonctions de contrôle moteur.

Mathieu SOUM
Université Paul Sabatier
Master 2 – Développement Logiciel

Stage réalisé chez Aboard Engineering
Maître de stage : Sébastien RICHE
Tutrice universitaire : Isabelle FERRANÉ

Année universitaire 2014 - 2015

AUTOSAR



ASW
L'application issue du code généré.
Contient la glue ansi que la gestion
de la communication CAN

RTE
La couche de laison entre le BSW et l'ASW.
Lie également les SoftWare Components
(SWC) entre eux.

BSW
Le microgiciel embarqué dans le calculateur.
Ce sont les pilotes et API pour utiliser les
entrées/sorties de la couche HW.

HW
Le matériel du calculateur