CLEARSY
Safety Solutions Designer

Séminaire
SEPT2020

AIX
LYON
PARIS
STRASBOURG

WWW.CLEARSY.COM

Méthodes Formelles, Développement Logiciel et Systèmes Critiques

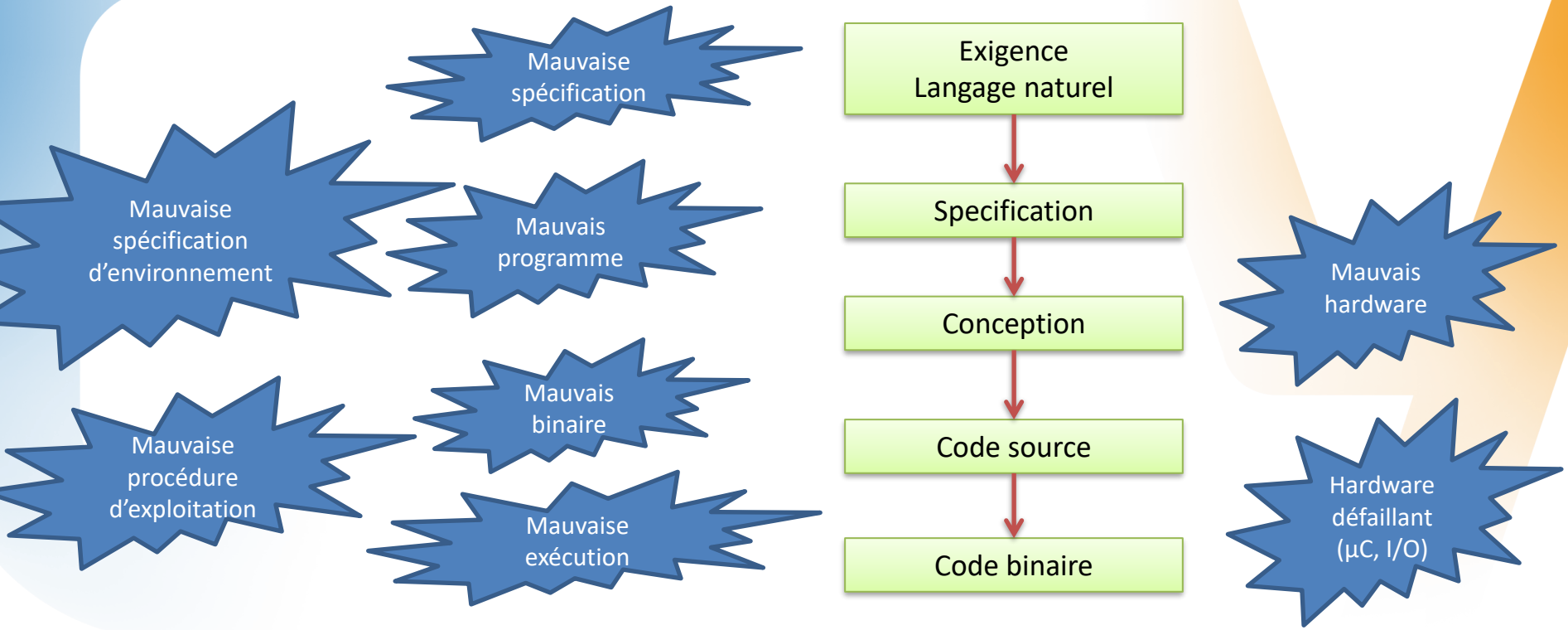
Sherbrooke, 15 Septembre 2020

Thierry Lecomte
Directeur R&D

THIERRY.LECOMTE@CLEARSY.COM

Système défaillant

« Ça compile donc ça marche »



Systèmes Critiques

- Systèmes où la vie est en jeu
  - Trains
  - Avions
  - Voitures
  - Centrales nucléaires
  - Machines industrielles
  - Etc.

Standards pour Systèmes Critiques

- Spécifiques aux domaines applicatifs
- Recommandations
  - Pas de recette définitive pour la production de système sûr
  - Couvre les développements HW/SW et le processus de dév.
- Démonstration de sûreté (*safety case*)
  - Evénement redouté pas plus fréquent qu'exigé

Niveaux de sûreté

- Safety Integrity Level (*SIL*)
  - Niveau 3: 1 défaillance tous les 100 ans (10^{-7}/h)
  - Niveau 4: 1 défaillance tous les 10 000 ans (10^{-9}/h)
- Système = SW + HW + Environnement
  - Erreur de spécification
  - Erreur de développement, programmation, compilation
  - Mauvaise exécution
  - Système redondé (2+ processeurs, mécanismes de protection, etc.)

Méthodes Formelles pour Systèmes Critiques

- Acceptées pour certification
  - Hautement recommandées (EN50128, IEC61508)
  - Voire obligatoires (Critères Communs)
- Certification tierce
  - Indépendance
  - Responsabilité engagée

IEC 61508: Software design and dev. (table A.2)

Technique/Measure	Ref	SIL1	SIL2	SIL3	SIL4
1 Fault detection and diagnosis	C.3.1	---	R	HR	HR
2 Error detecting and correcting codes	C.3.2	R	R	R	HR
3a Failure assertion programming	C.3.3	R	R	R	HR
3b Safety bag techniques	C.3.4	---	R	R	R
3c Diverse programming	C.3.5	R	R	R	HR
3d Recovery block	C.3.6	R	R	R	R
3e Backward recovery	C.3.7	R	R	R	R
3f Forward recovery	C.3.8	R	R	R	R
3g Re-try fault recovery mechanisms	C.3.9	R	R	R	HR
3h Memorising executed cases	C.3.10	---	R	R	HR
4 Graceful degradation	C.3.11	R	R	HR	HR
5 Artificial intelligence - fault correction	C.3.12	---	NR	NR	NR
6 Dynamic reconfiguration	C.3.13	---	NR	NR	NR
7a Structured methods including for example, JSD, MASCOT, SADT and Yourdon	C.2.1	HR	HR	HR	HR
7b Semi-formal methods	Table B.7	R	R	HR	HR
7c Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	C.2.4	---	R	R	HR
8 Computer-aided specification tools	B.2.4	R	R	HR	HR

a) Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. Only one of the alternate or equivalent techniques/measures has to be satisfied.

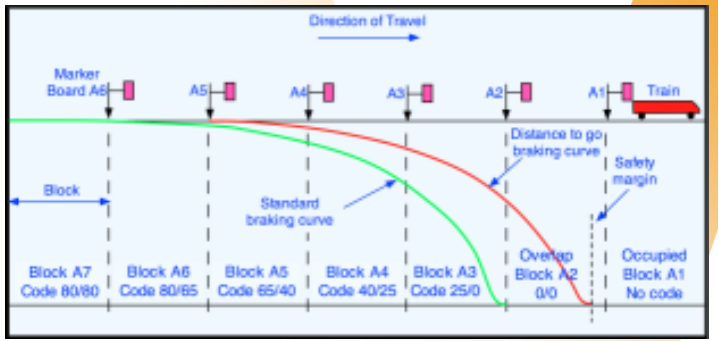
b) The measures in this table concerning fault tolerance (control of failures) should be considered with the requirements for architecture and control of failures for the hardware of the programmable electronics in part 2 of this standard.

Engineers say Boeing pushed to limit safety testing in race to certify planes, including 737 MAX

May 5, 2019 at 6:00 am | *Updated May 5, 2019 at 1:24 pm*

Méthodes Formelles et Ferroviaire

- La conduite n'est pas de sécurité
  - Pas besoin de méthodes formelles pour piloter un train
- Des garde-fous formels (méthode B)
  - Localisation (graphes)
  - Contrôle d'énergie cinétique (calcul entiers)
  - Freinage d'urgence (équations booléennes)



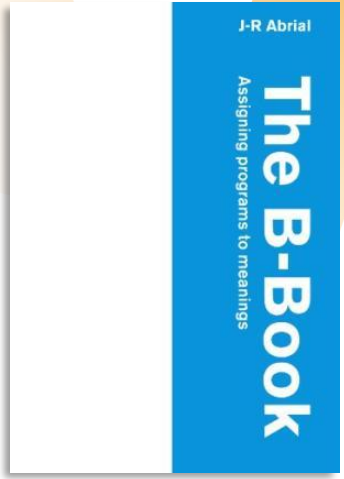
Braking curves

https://www.methode-b.com/

https://www.atelierb.eu/

Méthodes Formelles et Ferroviaire

- Développement de logiciels de sûreté
  - Modèle formel de logiciel (spécification et implémentation)
  - Preuve mathématique de cohérence



REFERENCES

[1] *Météor: A Successful Application of B in a Large Project*
FM 1999, Toulouse
Patrick Behm, Paul Benoit, Alain Faivre, Jean Marc Meynadier

[2] Using B as a High Level Programming Language in an Industrial Project: Roissy VAL
ZB 2005, Guildford
Arnaud Amelot, Frédéric Badeau

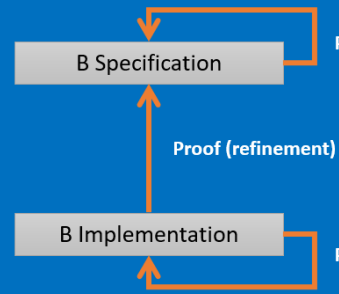
Méthodes Formelles et Ferroviaire

« Only inactive sequences can be added to the active sequences execution queue. »

Natural language requirement

```
activation_sequence = /* Activation d'une séquence non active */
PRE ¬(sequences = sequences_actives) THEN
    ANY sequ WHERE
        sequ ∈ sequences - sequences_actives
    THEN
        sequences_actives := sequences_actives ∪ {sequ}
    END
END;
```

```
activation_sequence = /* Activation d'une séquence non active */
VAR sequ IN
    sequ <-- indexSequenceInactive;
    activeSequence(sequ)
END;
```



```
void M0__activation_sequence(void)
{
    CTX__SEQUENCES sequ;

    sequence_manager__indexSequenceInactive(&sequ);
    sequence_manager__activeSequence(sequ);
}
```

C generated code

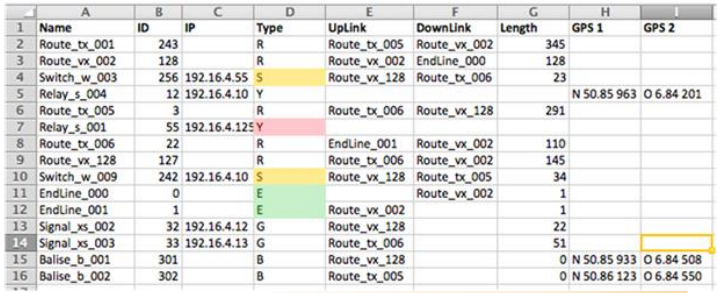
Cyclic software single-thread

0x01F970	FFFF	8B4C	2440	89C5	8D7D	0C8B	4110	89CE
0x01F980	83C6	0C8D	1485	0000	0000	8D42	0883	F807
0x01F990	7617	F7C7	0400	0000	740F	8B41	0C8D	7D10
0x01F9A0	83C6	0489	450C	8D42	04FC	89C1	C1E9	02F3

Binary code

Méthodes Formelles et Ferroviaire

- Vérification formelle de données
  - Modèle formel de données (100k cellules)
  - Conformité par model-checking
  - Utilisé par la plupart des industriels



	A	B	C	D	E	F	G	H	I
1	Name	ID	IP	Type	UpLink	DownLink	Length	GPS 1	GPS 2
2	Route_tx_001	243		R	Route_tx_005	Route_vx_002	345		
3	Route_vx_002	128		R	Route_vx_002	EndLine_000	128		
4	Switch_w_003	256	192.16.4.55	S	Route_vx_128	Route_tx_006	23		
5	Relay_s_004	12	192.16.4.10	Y				N 50.85 963	O 6.84 201
6	Route_tx_005	3		R	Route_tx_006	Route_vx_128	291		
7	Relay_s_001	55	192.16.4.125	Y					
8	Route_tx_006	22		R	EndLine_001	Route_vx_002	110		
9	Route_vx_128	127		R	Route_tx_006	Route_vx_002	145		
10	Switch_w_009	242	192.16.4.10	S	Route_vx_128	Route_tx_005	34		
11	EndLine_000	0		E		Route_vx_002	1		
12	EndLine_001	1		E	Route_vx_002		1		
13	Signal_xs_002	32	192.16.4.12	G	Route_vx_128		22		
14	Signal_xs_003	33	192.16.4.13	G	Route_tx_006		51		
15	Balise_b_001	301		B	Route_vx_128		0	N 50.85 933	O 6.84 508
16	Balise_b_002	302		B	Route_tx_005		0	N 50.86 123	O 6.84 550

REFERENCES

[1] *Formally Checking Large Data Sets in the Railways*
ICFEM 2012, Kyoto
Thierry Lecomte, Lilian Burdy, Michael Leuschel

[2] Formal Data Validation in the Railways
SSS'16, Brighton
Erwan Mottin, Thierry Lecomte

Méthodes Formelles et Ferroviaire

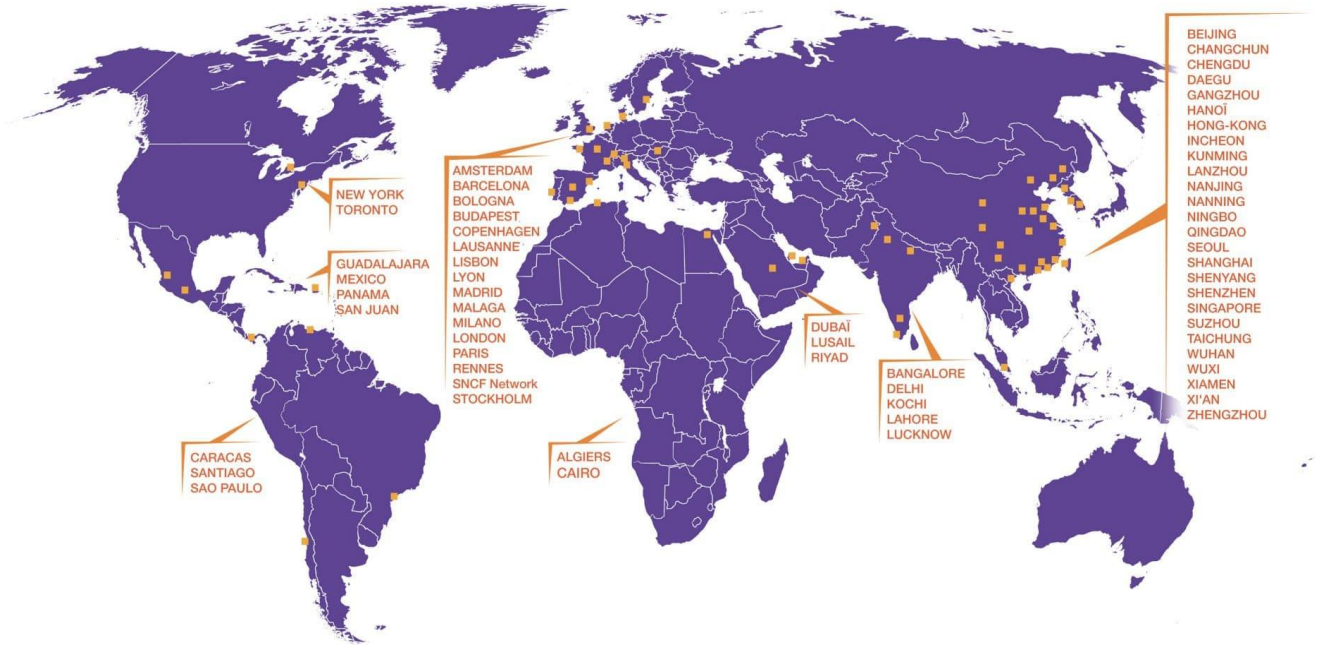
- Analyse formelle de systèmes
  - Modèle du raisonnement utilisé pour la conception
  - Très utile pour les systèmes « *legacy* »

REFERENCES

[1] *B-specification of Relay-based Railway Interlocking Systems Based on the Propositional Logic of the System State Evolution*
RSSR 2019, Lille
Dalay Israel de Almeida Pereira, David Deharbe, Matthieu Perin and Philippe Bon

[2] Safety Analysis of a CBTC System: A Rigorous Approach with Event-B
RSSR 2017, Pistoia
Mathieu Comptier, David Deharbe, Julien Molinero Perez, Denis Sabatier

30% des métros automatiques

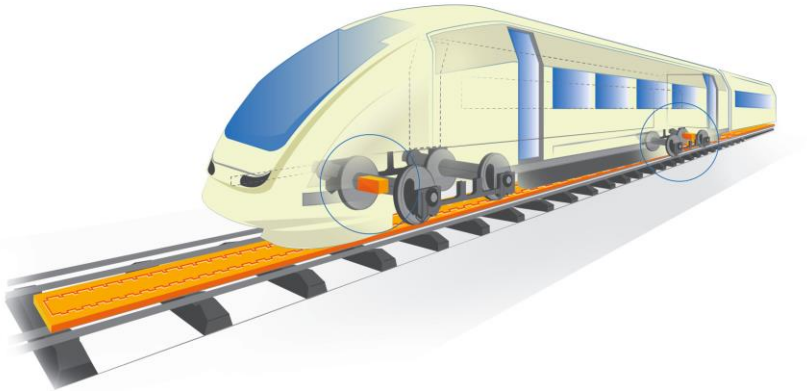


En Bref

- Utilisation raisonnée des méthodes formelles
  - Utilisation où ça sert
  - On a besoin de savoir pourquoi on fait les choses
  - Les méthodes formelles sont juste une partie de l'histoire
- Safety « by-design »
  - Un système pas conçu pour la sûreté ne sera pas sûr, même si on utilise des méthodes formelles
  - Prise de décision à partir d'une connaissance imprécise (capteurs)

Perspectives

- Besoins croissants – en relation avec la cybersécurité
  - Explosion de la population urbaine
  - Digitalisation croissante
  - Moindre/non-acceptation des accidents
- Autonomisation
  - Métros autonomes dès les années 90 (logiciel B v1.0 Paris ligne 14 depuis 1998)
  - Trains, avions, voitures, navettes, robots, etc. restent des problèmes entiers



MERCI DE VOTRE ATTENTION