



SE206: Modélisation, génération de code et vérification

Introduction générale

Ulrich Kühne, Etienne Borde
 {ulrich.kuhne,etienne.borde}@telecom-paristech.fr

Objectifs de l'UE

- Comprendre les objectifs et techniques d'analyse d'architecture des systèmes temps-réel embarqués.
- Découvrir les standards de spécification et modélisation les plus connus dans ce domaine (UPPAAL, AADL).
- Manipuler des outils d'analyse et de génération de code qui exploitent ces langages.

Objectifs de l'introduction

- Définir la notion de modèles
- Motiver l'utilisation de modèles pour la conception des systèmes temps-réel embarqués
- Expliquer l'importance du temps dans ces systèmes
- Introduire les objectifs d'analyse s'appuyant sur les modèles



Plan de l'introduction

1. Généralités sur la modélisation: abstraction, langage, modèles/méta-modèles, architecture
2. Travail en groupe sur la modélisation
3. Modélisation et génie logiciel: processus de développement, rôle des modèles
4. Modélisation pour les systèmes embarqués



Généralités sur la modélisation



Modèle = abstraction

- Un modèle est une abstraction de la réalité, abstraction à partir de laquelle il devient possible de raisonner
 - Calculer des **dimensions caractéristiques** (consommation énergétique, temps de réponse, température, etc.).
 - **Vérifier la conformité** des dimensions caractéristiques vis-à-vis d'exigences (poids d'un avion, nombre de passager, besoin en carburant par km...).
 - Fournir des **points de vue** différents en fonction des rôles, préoccupations et/ou expertises de chacun (constructeur ou utilisateur d'un train, d'un avion, d'une voiture).
- C'est une définition plutôt vague, et c'est le but: pour modéliser, il faut de la rigueur (objectif = calcul, vérification) et de l'ouverture d'esprit (moyen = abstraction, point de vue)...
- Prenons quelques exemples pour rendre tout cela plus concret...

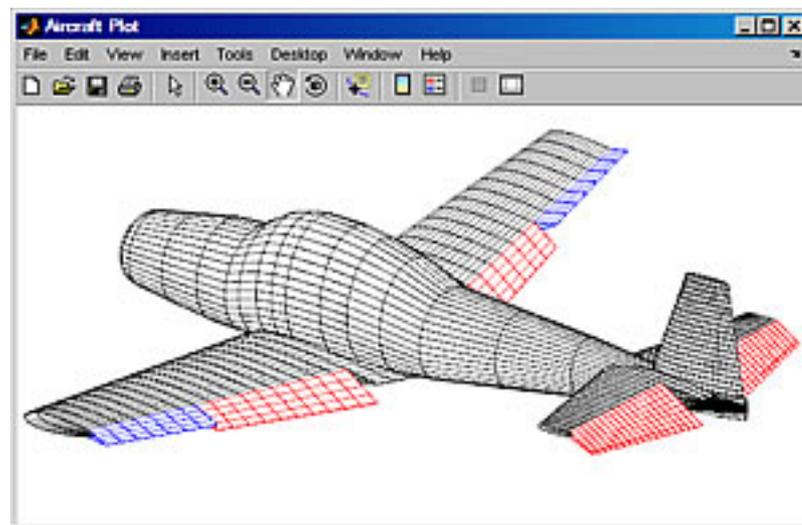
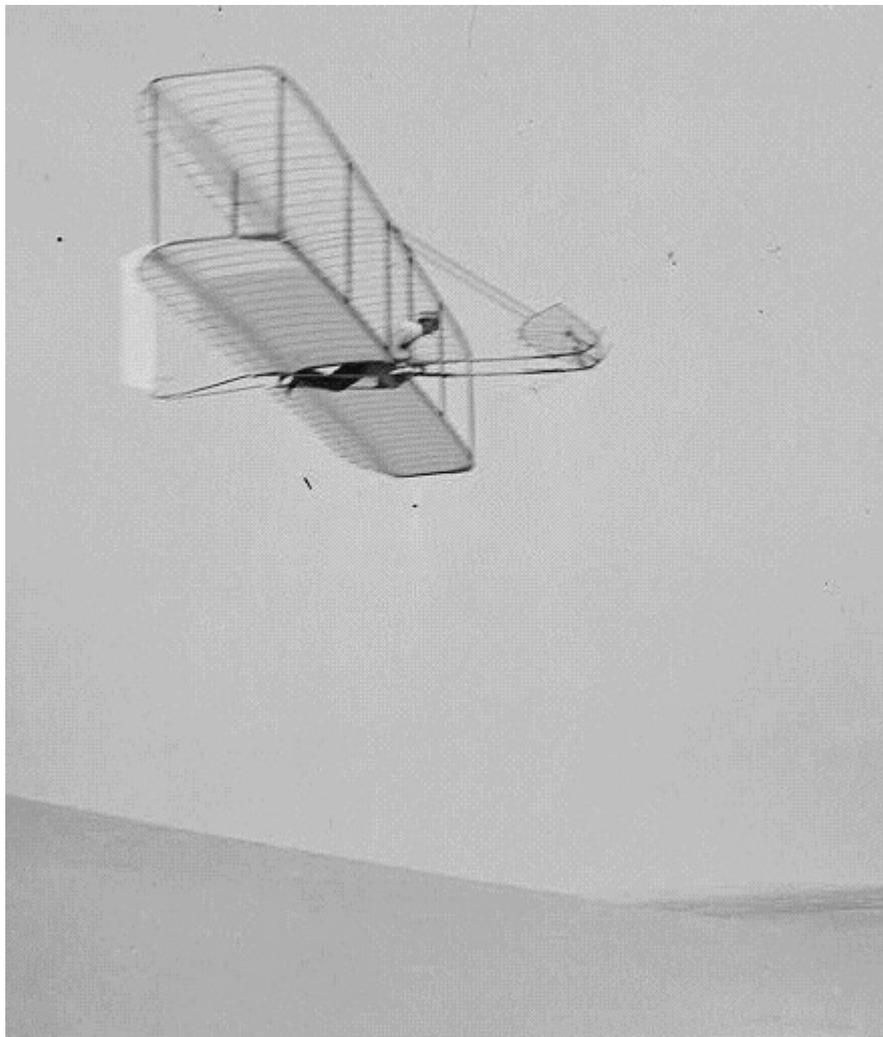
Définition de base

*Modeling, in the broadest sense, is the **cost-effective** use of something in place of something else for some **cognitive purpose**. It allows us to **use something that is simpler, safer or cheaper than reality instead of reality for some purpose**. A model represents reality for **the given purpose**; the model is an **abstraction of reality in the sense that it cannot represent all aspects of reality**. This allows us to deal with the world in a simplified manner, avoiding the complexity, danger and irreversibility of reality.*

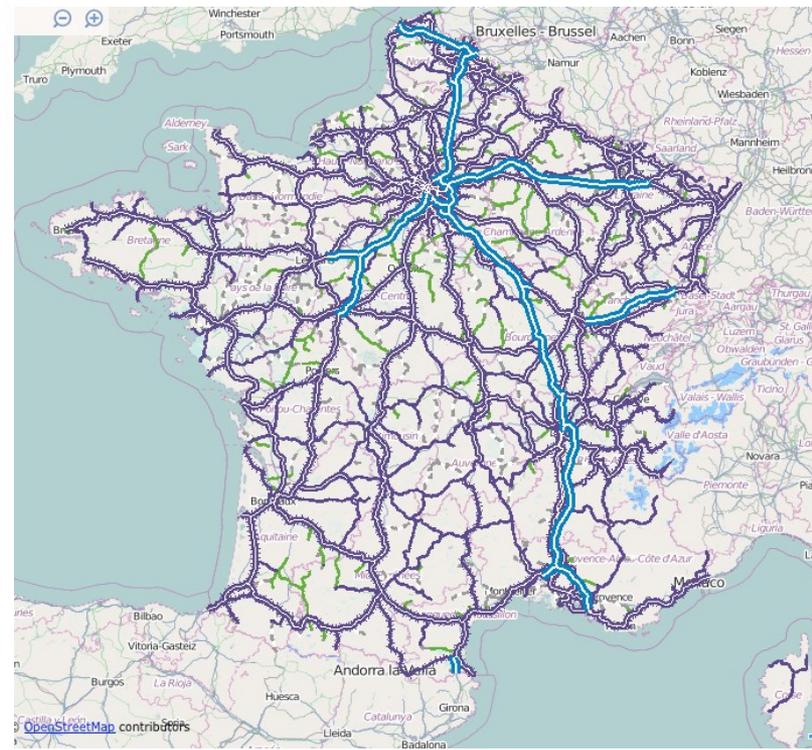
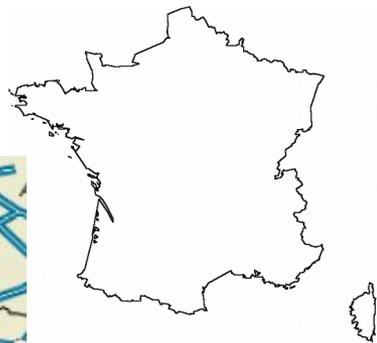
*“The Nature of Modeling” Jeff Rothenberg in Artificial Intelligence, Simulation, and Modeling, L.E. William, K.A. Loparo, N.R. Nelson, eds. New York, John Wiley and Sons, Inc., **1989**, pp. 75-92*



Réalité vs. modèle



Modèles de la France...



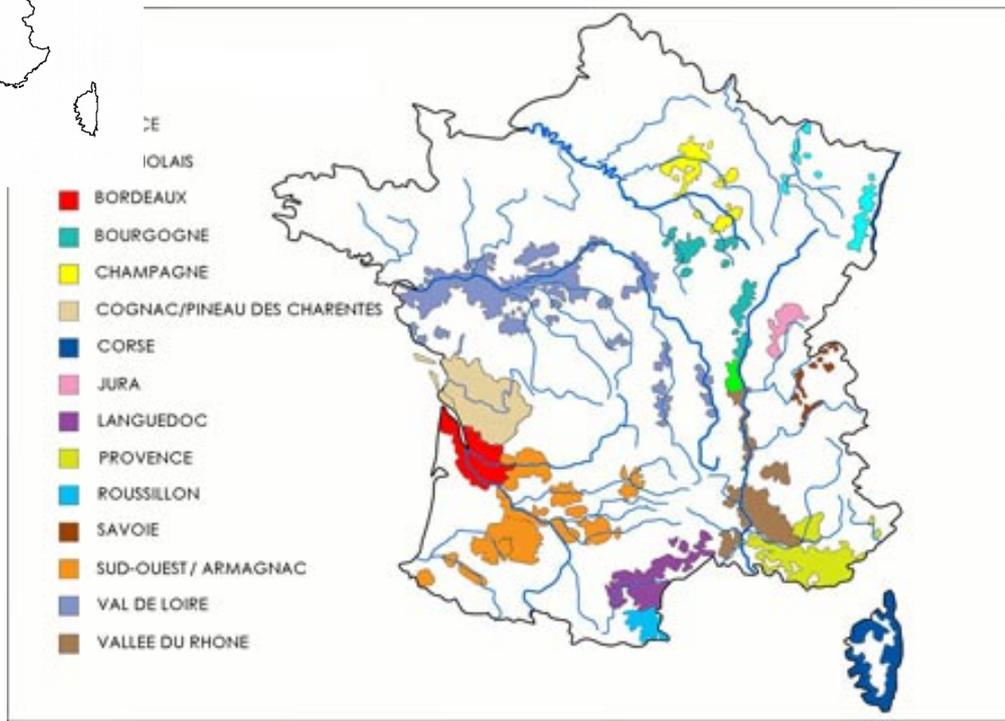
Réseau des autoroutes

Réseau des voies ferrées

Modèles de la France...



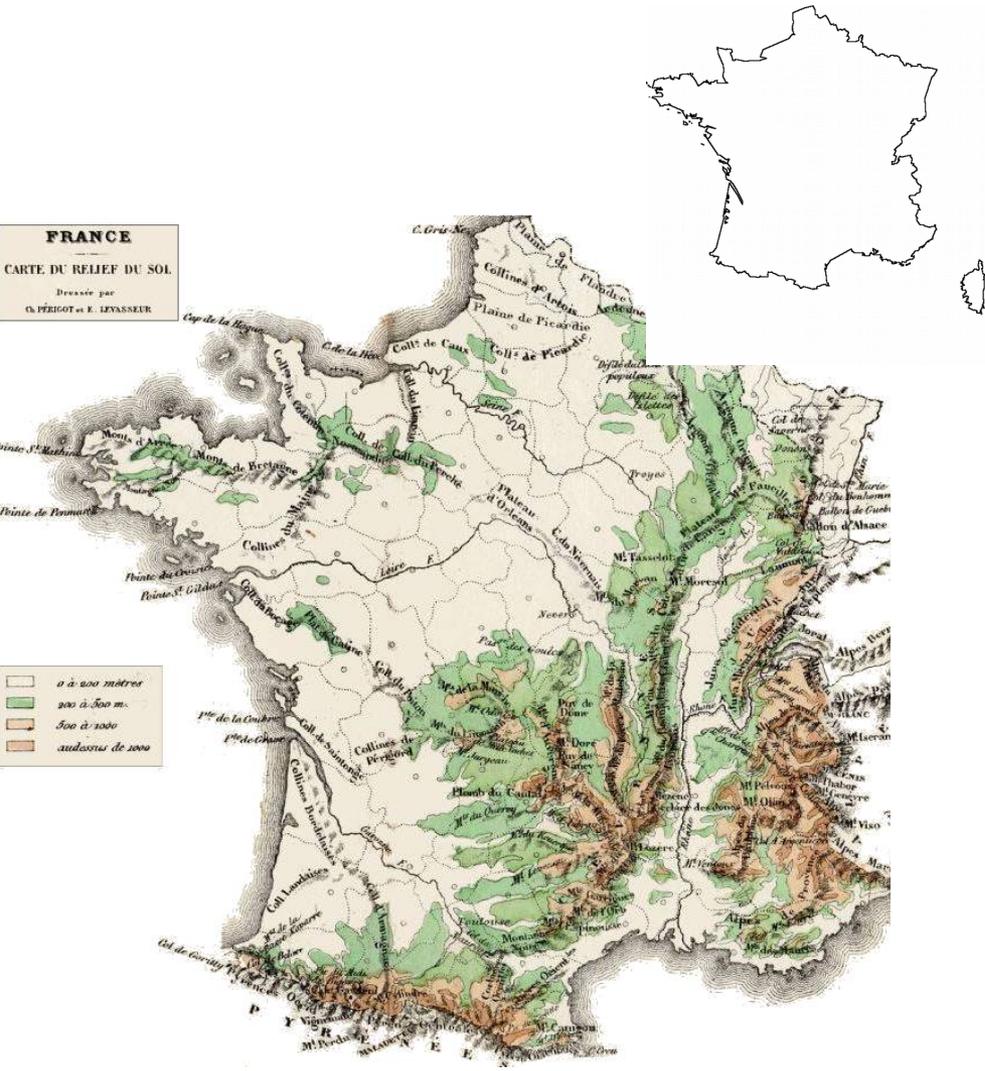
Spécialités culinaires



Vignobles



Modèles de la France...

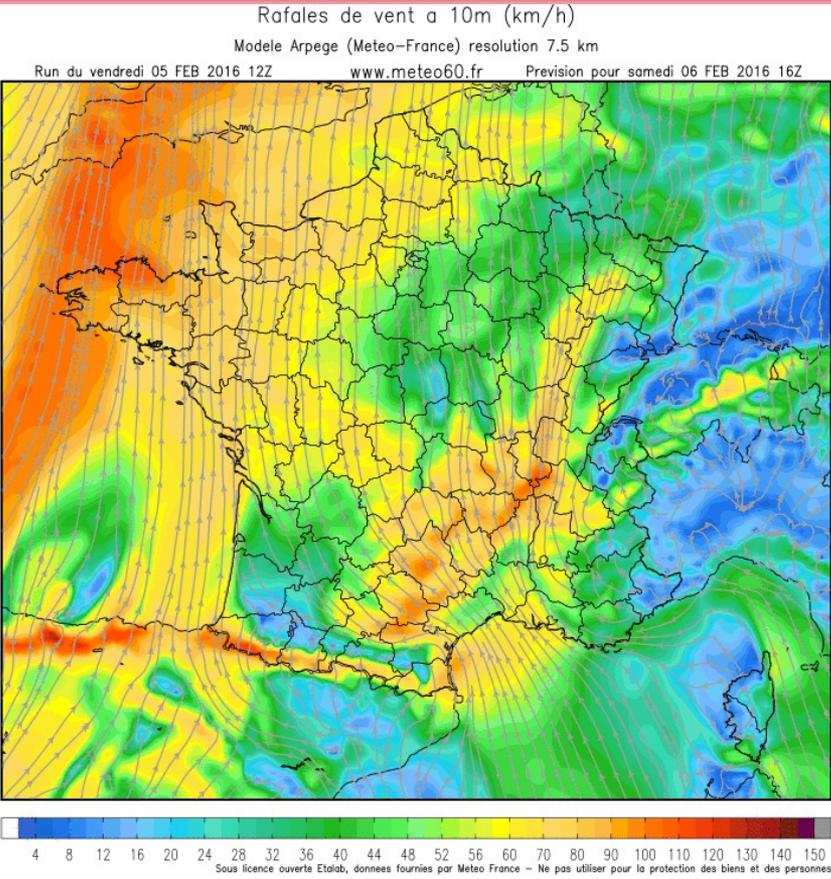


Reliefs

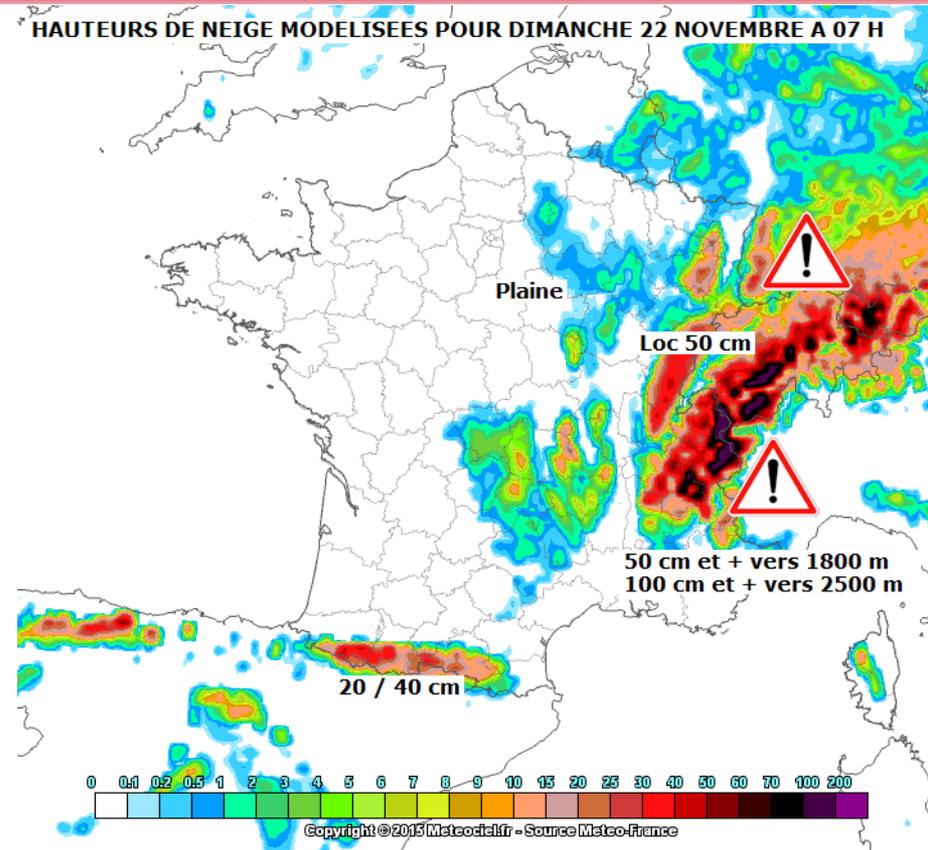


Plages

Plus sérieusement... Cartes météo



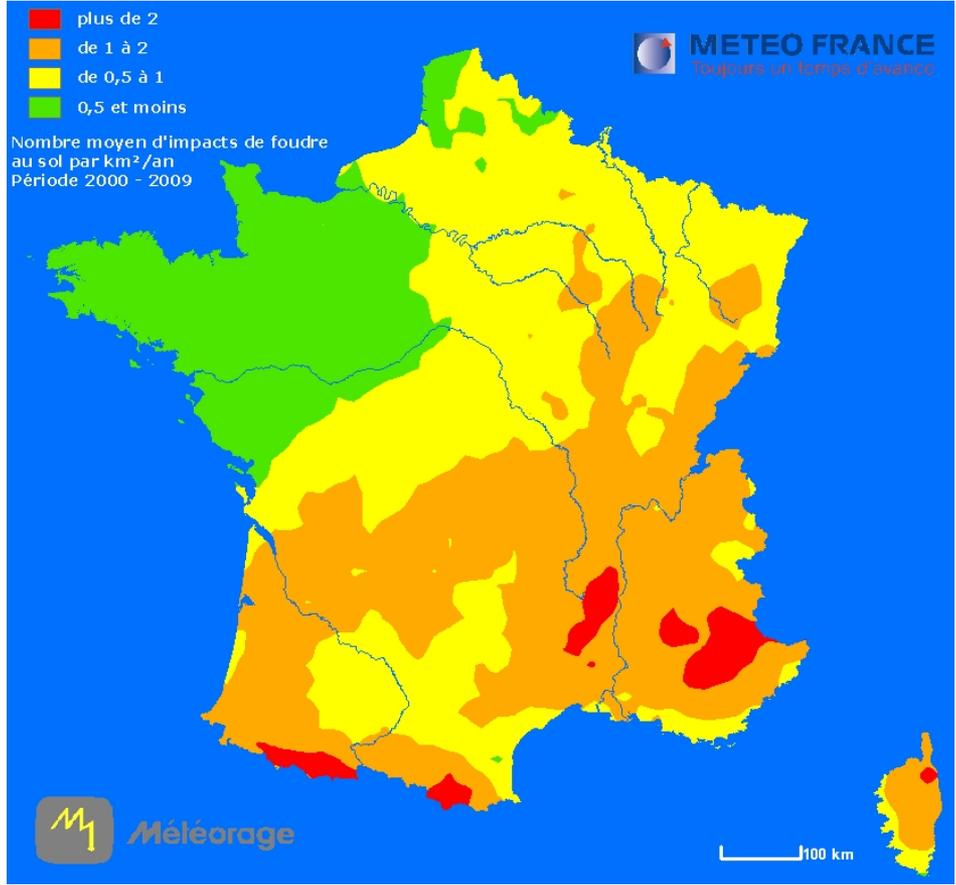
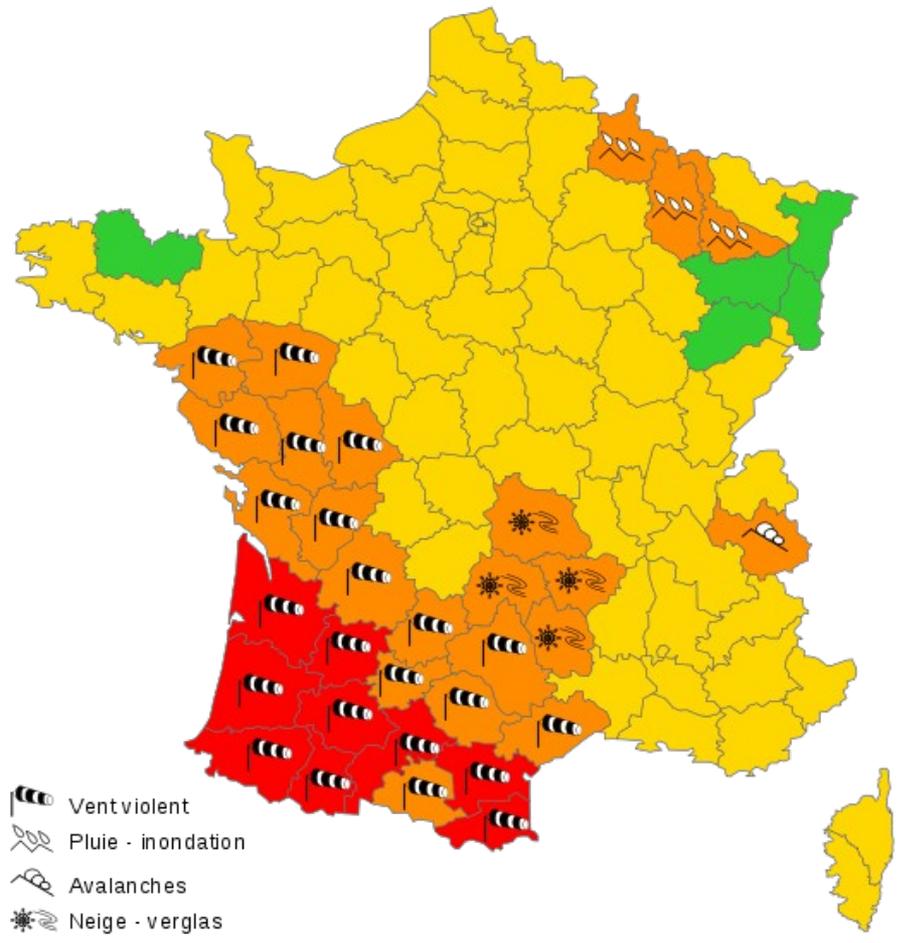
Carte des rafales de vents,
avec direction et vitesse



Carte de l'enneigement, avec
Risques d'avalanches



Cartes météo



Carte des niveaux de vigilance, vent/neige/avalanches

Nombre moyen de coup de foudre, par km², par an, entre 2000 et 2009

Carte météo = abstraction

- **Un modèle est une abstraction de la réalité**
 - Calculer des **dimensions caractéristiques**
 - Niveau de risque d'inondation/incendie/avalanche
 - Température max/min/moyenne
 - Vitesse des rafales de vent
 - **Vérifier la conformité** des dimensions caractéristiques vis-à-vis d'exigences
 - Un marin va vérifier la présence de vent, sa direction et sa vitesse...
 - Un alpiniste vérifiera la présence de neige, la présence de vent, la température...
 - Fournir des **points de vue** différents en fonction des rôles, préoccupations et/ou expertises de chacun.
 - Carte des températures, des vents, des précipitations, etc...

Modèle = syntaxe + sémantique

- Un modèle a aussi pour objectif de donner une représentation facile à comprendre, interpréter, retenir...
- Il faut pour cela:
 - Une représentation textuelle et/ou graphique.
 - Une sémantique associée, plus ou moins abstraite.
- Sur les exemples précédents, les couleurs servent souvent de représentation graphique (syntaxique) alors que la légende explique la sémantique associée.
- La sémantique associée est plus ou moins abstraite...
- Les modèles sont souvent des moyens de « prédiction »...



À vous de jouer (modéliser)

Problèmes à résoudre

- 1 Trouver un itinéraire
(de station A à station B, sans temps de parcours)
- 2 Commander la signalisation
(pour que les trains ne se rentrent pas dedans)
- 3 Analyser la capacité de transport nécessaire sur une ligne
(combien de trains avec combien de places)
- 4 Calculer le pire temp de trajet pour un itinéraire
(sans retards / accidents)



Travail en groupe

- Quel est le modèle choisi?
(syntaxe + sémantique)
- Comment répondre au problème posé?
(algorithme d'analyse)
- Quels informations/détails additionnels sont nécessaire pour résoudre le problème?
(entrées + hypothèses)
- Quels informations/détails ne sont pas nécessaire pour résoudre le problème?
(abstraction)
- 20 minutes + 5 minutes présentation



Modélisation et génie logiciel

Les modèles en informatique

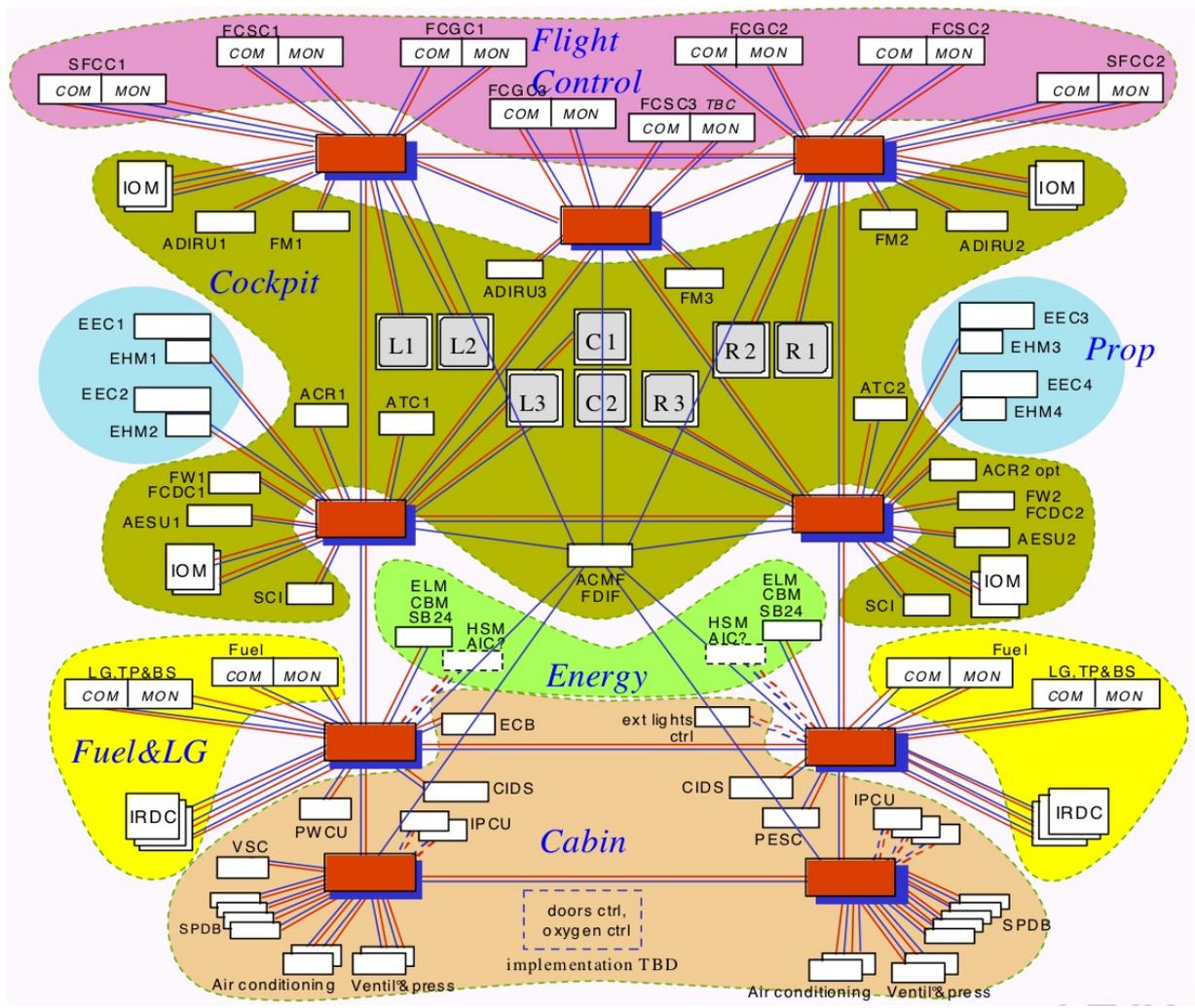
- Un programme écrit en C est-il un modèle? ...
 - *cost-effective use of something in place of something else for some cognitive purpose.*
 - Le langage C a bien été conçu pour réduire le cout de développement/portage d'un système d'exploitation
 - C'est un langage de « haut niveau », objectif cognitif.
 - *It allows us to use something that is simpler, safer or cheaper than reality instead of reality for some purpose. A model represents reality for the given purpose; the model is an abstraction of reality in the sense that it cannot represent all aspects of reality*
 - Il est plus facile d'écrire du code en C qu'en assembleur, l'objectif est le même: automatiser l'exécution d'un algorithme sur un ordinateur
 - On s'abstrait du fonctionnement de la machine (jeu d'instruction du processeur, structure de la mémoire, registres, etc...).
- Est-ce suffisant? ... Analyse/Vérification?

Rôle des modèles pour les systèmes embarqués

- S'appuyer sur des modèles lors de la conception pour « valider » au plus tôt
- Faciliter les phases de conception en s'appuyant sur une représentation abstraite (modèle):
 - Support à la communication entre ingénieurs
 - Support à la documentation des solutions choisi
 - Support à la validation au plus tôt des exigences, par raisonnement sur le modèle (calcul de propriétés caractéristiques)
 - Support à la configuration de la plate-forme (OS+routage)
 - Support à la génération de code source (C, Ada, Java...)

- En informatique, les modèles permettent de représenter une **architecture** informatique: l'**organisation** et les **relations** existantes entre les **éléments** (logiciels, matériels, et/ou sous-systèmes) d'un système informatique.
- Rappel, un modèle utilise un langage :
 - syntaxe + sémantique
- On parle de langage de description d'architecture

Exemple d'architecture avionique



Comment, et pourquoi modéliser une telle architecture?



Modélisation pour les systèmes embarqués

Particularités des systèmes embarqués

- **Les systèmes embarqués contrôlent/pilotent des systèmes physiques en interaction avec:**
 - Le monde extérieur (souvent appelé environnement du système) via des capteurs et des actionneurs.
 - Des êtres humains auxquels ils rendent service (transports, robotique, télécommunications, énergie, etc.).
- **Ils sont donc soumis à des contraintes:**
 - De temps (temps de réaction à des évènement, gigue sur le calcul de données)
 - De fiabilité (capacité à détecter/isoler/recouvrir des fautes)
 - D'empreinte mémoire (dimensionnement de la pile? Du tas?...)
 - ...



Modélisation et temps

- Le but de la modélisation d'une architecture de système temps-réel embarqué va donc être de vérifier que les données sont toujours disponibles **à la bonne date**.
- Dans un système réel (avionique, ferroviaire, automobile, spatial ...) le nombre de tâches, processeurs, bus est important.
 - Il faudra donc un modèle d'exécution, de communication, qui prenne en compte la notion de temps d'exécution, de fréquence d'activation, etc...
 - Il faudra aussi faire attention aux ressources d'exécution partagées
 - Le processeur, partagé par des tâches: l'ordonnancement jouera un rôle important dans ce type de modèle
 - Les variables, avec les mécanismes de protection type verrous
 - Les bus de communication



Organisation de l'UE

1. TH cours 2: Automates temporisés
2. TH cours 3: Vérification du matériel
3. TH cours 4: Response time analysis
4. TH cours 5 et 6: AADL
5. TH cours 7 : Génération de code

Alternance de THs de cours et de THs de TD/TP

Les TP sont notés! (pas de contrôle finale)

Positionnement des parties du cours sur le processus en cycles en V

