Part 1: Systems interoperability and related problems
What is systems interoperability

- **ISO definition**
  - The possibility to communicate, to execute programs, or to transfer data between functional units in such a way that users doesn't need to know the characteristics of the functional units

- **DOD definition**
  - The ability of systems, units, or forces to provide services and to use the services so exchanged to enable them to operate effectively together
Connecting people

- Organisational and Enterprise Reality: everywhere interoperability
- Size and global complexity is a, for not to say THE, major issue

Target organisation

N federated systems to assist people in their missions

Information exchange mechanism to coordinate actions

Development organisation for each S1, S2, etc.

N federated systems to form a global coherent system of systems
Sharing knowledge between actors
The challenge of semantic

- How to define constructively what is the semantic of actions

The semantic of the exchange must be validated to be sure that the actions are coordinated.
The problem of system safety in an interconnected world

Stimulus  Response

Reliability of individual elements is \((1 - \varepsilon)\)

How safe is the response? How reliable are the individual components?

\[
\text{Reliability}(\text{chain}) = \left[\text{Reliability}(\text{element})\right]^{\text{Length of the chain}}
\]
Sharing data, services, and events

The complexity of an interoperability project

Middleware to couple (loosely and/or tightly) the federated systems

Messages Sent and Received

System S1

System S2

System Sn

Metadata shared between the federated systems

Federated services

Federated data

History and traces of S1 interactions

History and traces of S2 interactions

History and traces of Sn interactions

Global history and traces for the federated systems as a whole
Sharing platforms and resources
Some critical interfaces

- Abstract Machine for S1 business computing
  - Local S1 communications
  - High level protocol to connect S1 to the global communication infrastructures
  - Others AM for S2, S3, ...

- Communication Abstract Machine for the federation (network computing)

- Data and services to manage the global communication infrastructure
  - (MIB, Directory access, permanent data, RAS services, etc.)

- User’s interaction and control
Reliability and safety are the fundamental issue of interoperability

- When a fault occurs in one of the systems of the federation, where is or are located the defects at the origin of the fault?
  - Is it one system alone? Or is it a conjunction of unexpected events between several systems?

- How to reconstruct the history of the transformations and events leading to the fault
  - To be sure of the reconstruction the global system must be deterministic, at least at a given level of abstraction (coarse grain)
The challenge of safety

- Integration validation verification and test of C4ISR systems as a whole is a major issue
- Black box testing is definitively unworkable in a so large context
- Formal methods are out of scope (complexity)
- Systematic construction of maturity, using all aspects of validation such as simulation, must be performed → Integration is the fundamental process + Project management office
- Intelligent redundancy, using constraints, must be used to generate pertinent test suites
- Modelling and constructive approach, step by step, is the solution to master complexity
Part 2:  
Making the global system deterministic 

Application of the transaction processing techniques to **High Level Abstract State Machine - HLASM**
Evidences from IT industry (1/2)

- **From the Hardware-Software interface viewpoint**, the execution of machine instructions of a highly parallel processor such as Intel Itanium, is deterministic
  - We know exactly how to validate such an interface since a long time ago
- **In a transactional environment, each individual transaction is deterministic**, from the user point of view (ACID properties, 2PC)
  - Classical OLTP is a well mastered techniques
In an information integration perspective the concept of atomic transaction must be extended within a larger concept of a **transaction flow based programming model** adapted to the long transactions of the real world

- The execution of such a flow is done through a **flow engine**, i.e. an **ASM** (and the associated language)
- Mechanisms to **compensate** globally such long transactions must be provided – Compensation must be based on business process rules (i.e. semantic, from the user point of view)
Questions – Problems

- **How and where to observe the system**
  - For the least, the behaviour of the system must be observable, nominal states as well as wrong states, through the **interfaces**
  - **System and software configuration is a key point**

- **How and where to control the system**
  - For the best, wrong states, once detected, **must be compensated** (*UNDO* the transaction) in order to keep the system in a coherent state
  - Compensations are ad hoc programs which are to be provided by the architecture of the system (cannot be generated automatically as in classical OLTP)
Solution

- **Architecture of the system using the concept of well defined abstract machines hierarchically organized is a key point:**
  - To master the configuration and the static complexity of the system using semantically well identified operations and services → **Allows continuous integration**
  - To make the abstract level visible through the **various logical interfaces** offered by the machines
  - To **master the dynamic complexity**, at execution time, by disentangling the various histories of the interactions within the global system
  - To adapt the mandatory **redundancy** to the right level required by the mission of the system
Modèle d’exécution générique - MVC

Serveur d'interactions Homme-Machine
- IHM
  - Gestion des interactions avec l’usager Navigation
  - Flux d'événements IHM
  - Mémoire IHM
  - Symbologie et icônes spécifiques à l’IHM

Serveur d’applications
- Programmes de contrôle mis en œuvre par le moniteur
- Moniteur d’enchaînement des transactions
- Flux de requêtes et de réponses au requêtes
- Mémoire de travail des transactions

Serveur de données
- Moniteur d’enchaînement des requêtes
- Mémoire de travail des requêtes
- BD-1
- BD-2
- BD-m

Décider
Transformer
Nommer et restituer

Administrer et mettre en œuvre les serveurs

J.Printz / CNAM - CMSL / Interopérabilité des systèmes d'information / Journée INRETS du 19 Janvier 2006
Machine MVC généralisée – Actions locales et distantes

Serveurs d'applications

- IHM
- Moniteur local
- Mémoire LD
- Transaction-1
  - Requêtes distantes
- Façade Transaction-2
  - Appel distant

Serveurs de données

- Réseau
- SD1
  - Moniteurs distants
  - Mémoire D
  - Mémoire L
  - BD-1
  - BD-2
- SD2
  - MI pour le système S2
  - BD-3

SA1

SA2

J.Printz / CNAM - CMSL / Interopérabilité des systèmes d'information / Journée INRETS du 19 Janvier 2006
Notion de MACHINE ABSTRAITE (HLASM) – Les organes fonctionnels

- Structure de contrôle
  - Vecteur d'état
    - Mémoire partagée qui permet aux actions d'interagir avec le contrôle (événements programmés, statuts des opérations effectuées)
    - • Nomenclature des stimuli et événements modifiant le contrôle.
    - • Mémoire propre à la structure de contrôle ; contient le ou les programmes de contrôle.

- Structure actions-opérations
  - Mémoire propre à la structure action (files d'attentes, caches, etc.).
  - • Nomenclature des actions-opérations.

- Mémoire de la machine abstraite
- Structure Données (MCD)
- Types de données représentant la réalité

- Structure de surveillance et de service de tous les organes de la machine abstraite – Vérification des invariants structurels – Metadata

J.Printz / CNAM - CMSL / Interopérabilité des systèmes d'information / Journée INRETS du 19 Janvier 2006
Le niveau service

Dispatching sur les différentes opérations
(avec adaptation éventuelle des formats d’appel)

Contraintes sémantiques associées au service
Structure du vecteur d’état d’une opération

Vecteur d’état des opérations de niveau N1 ou N0

Attributs généraux

Attributs spécifiques O1

Attributs spécifiques O2

... 

Attributs spécifiques O2

Ad hoc actions can be executed on demand
Le niveau opérations divisibles et opérations indivisibles

Niveau 1

Niveau 0

Automate de workflow (Une structure interprétative sur le niveau N0 est possible pour cet organe)

Propriétés ACID (logiquement indivisible)

Paramétrage du niveau N1 sur N0
Part 3: How to use syntactic and semantic constraints to generate integration tests and improve system predictability and system safety
**Input, Output, Context of an operation/service**

- **Input state** properties at time $T_0$, beginning of the operation
- **Output state** properties at time $T_1$, end of the operation

---

**Operation Service**

- **Input domain + constraints**
- **Resources necessary to the execution of the operation (allocation and restitution)**
- **State vector**
- **Allocate**
- **Free**
- **Output domain (Nominal functional Result)**

- **Context**

- **Non functional characteristics**
- History of the transformation and events
Input state and output state properties

- To be valid, input data and context of the operation must follow a given set of rules.
- According to the input values, to be valid, output data and context of the operation must respect a given set of rules.
- Rules may be expressed using a constraint language:
  - Declarative as OCL, or logical expressions, or grammar expressions.
  - Imperative (for example an automaton to be executed by a given ASM).
Structures of the input and output domain

- Input states domain where data are valid
- Borders of the valid input domain
- Wrong states
- All possible combination of the input data (input set extension)
- Other unexpected input data wrongly directed to the service

- Well known logical law as Excluded Middle is no more working
Validation to be performed when a message is received – INPUT – by the service

A message of type T is received by the service

Syntactic analysis of the message

Syntactic rules (i.e. grammar)

Conversion rules (type conversion)

Semantic analysis of the message

Semantic rules

Translation of the message into the internal language of the operation

Update the context is necessary

To next step of the transformation

Errors

On line help and diagnostic

Errors

On line help and diagnostic

J. Printz / CNAM - CMSL / Interopérabilité des systèmes d'information / Journée INRETS du 19 Janvier 2006
Validation to be performed when a message is sent – OUTPUT – by the service

1. Message of type T is to be sent
2. Collect the data to construct the message
3. Semantic analysis of the collected data
4. Building of the message
5. Translation of the message into the internal language of the receiver
6. Sending the message to the receiver(s)

Errors

On line help and diagnostic

Semantic rules

Syntactic rules (i.e. grammar)

Conversion rules (type conversion)
Input and output domain languages are obviously not independent, but they provide useful redundancy to improve system reliability and safety.

It’s a way to practice N-version programming by using semantic properties of the exchanges which must be specify in any case.
Constraints provide an explicit criteria to add useful redundant information to

Processing of the syntactic and semantic constraints, when based on ASM, provide a way to master complexity (i.e. size of the associated language), to generate automatically assertions to check input and output at execution time, to product pertinent data for V&V test at integration time

The method is based on the best mastered techniques of computer sciences, i.e. compiling techniques and transaction processing (ACID and 2PC) to specify deterministic interfaces
Méthode GAP (J.Printz – JF.Peyre)

- Constraint expression may be used to generate input or output data exactly on the border of the domain, or data which slightly violate the semantic rules
  - Our test experience shows that errors are more frequent on the neighbourhood of the border than anywhere else
  - Border must be carefully investigated

- Study done under contract N° : 99-34-027 of DGA/STTC
  - i.e. : Génération Automatique de tests Pertinents