

Politiques et interactions de politiques dans les services télécom

Luigi Logrippo

Université du Québec en Outaouais et

Université d'Ottawa

Canada

Remerciements à
plusieurs collègues et
collaborateurs

Bilingual presentation

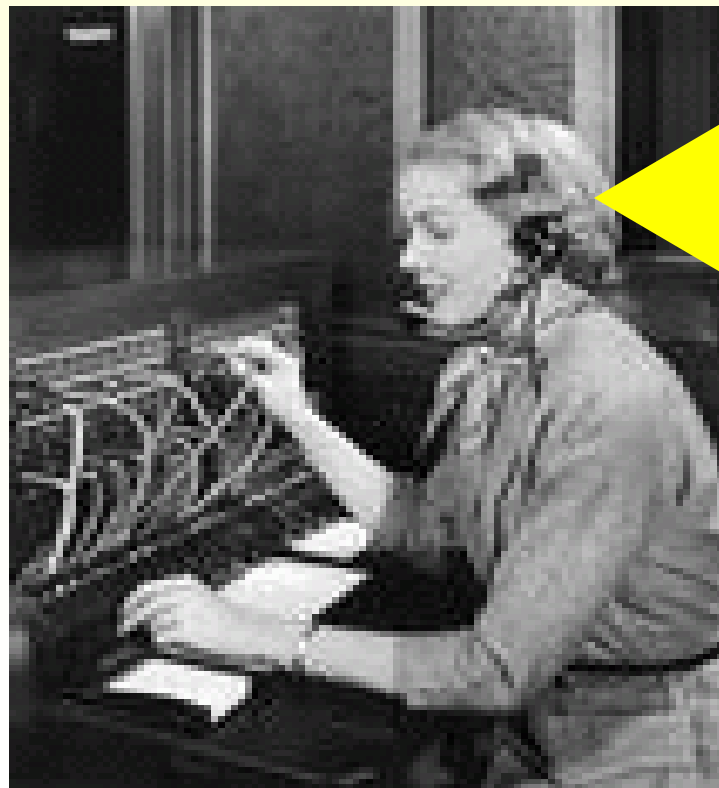
Résumé



- Nous pouvons désormais définir des politiques complexes pour déterminer le comportement de nos services télécom
- Cependant ces politiques peuvent interagir avec d'autres politiques du même ou d'autres usagers
 - ◆ Résultats surprenants ou décevants pour l'utilisateur
- Des méthodes basées sur la logique peuvent être utilisées pour détecter certaines interactions
- Nombreuses applications:
 - ◆ sécurité, orchestration, chorégraphie, contrats électroniques, etc.
- Osmose et interaction entre les politiques du monde télécom et celles du monde réel
- Entre les politiques télécom et le monde de la loi

This is where we started...

These gentle ladies knew a lot about telecom services...



Natural Intelligence

The old good time...

- Please Operator, put me in touch with a heart doctor... may be Dr. Shepp?
- Oh, no, she is out of town these days, Dr. Toby replaces her...
- Yes, put me in touch with Dr. Toby.
- Hhhmm... let's see... Thursday afternoon he is usually at his office... but at that time he does not want to take calls. Is this urgent?
- Yes!
- We'll try the office anyway, if not we'll try the hospital...



FI and resolution!

Automation

Switches were later automated
and we are still trying to recover from that

Feature Interactions

- Unfortunately, switches get mixed up with complex features, hence the research on Feature Interaction

Well-known interaction OCS/CF

A has C in OCS list



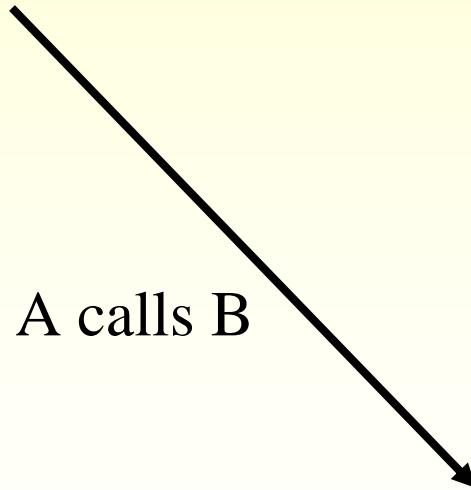
A

3. A gets connected to C



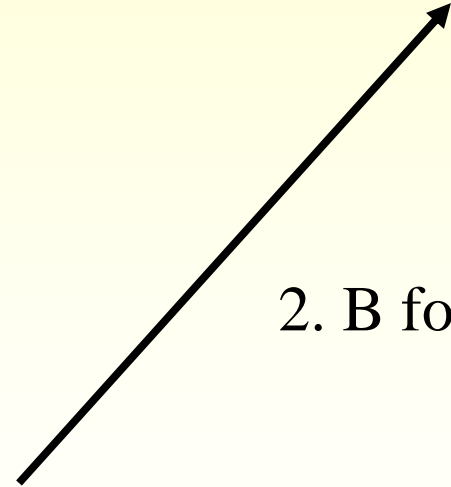
C

1. A calls B



B

2. B forwards to C



B has CFA to C

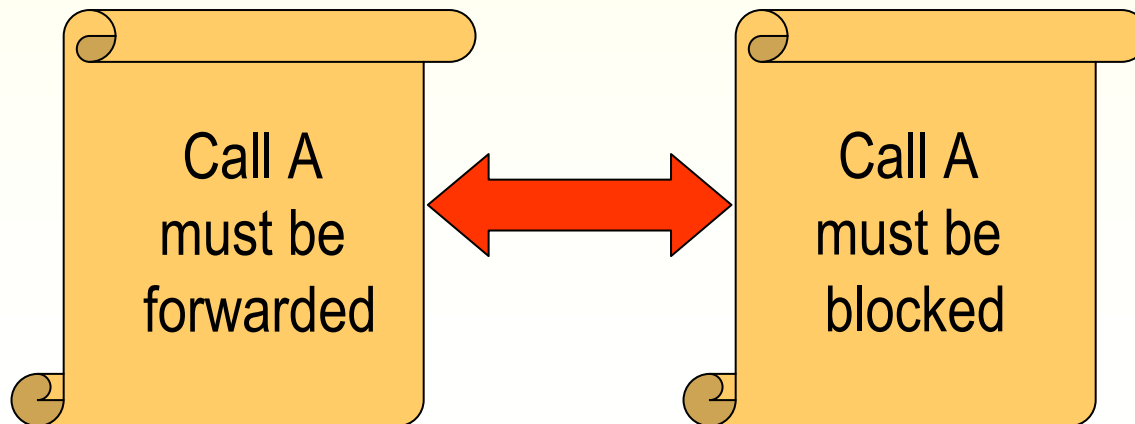
OCS invariant is violated.

Contradiction

- Nous avons observé ici une contradiction entre
 - ◆ l'intention de l'abonné à une fonctionnalité et
 - ◆ le résultat d'une autre fonctionnalité

Que c'est essentiellement une IF

- Un grand nombre de définitions de IF ont été proposées
- Une définition possible:
 - ◆ une IF est une *contradiction* entre des ensembles d'*intentions* et politiques coexistants



Law of non-contradiction

The most indisputable of all beliefs is that *contradictory statements are not at the same time true*

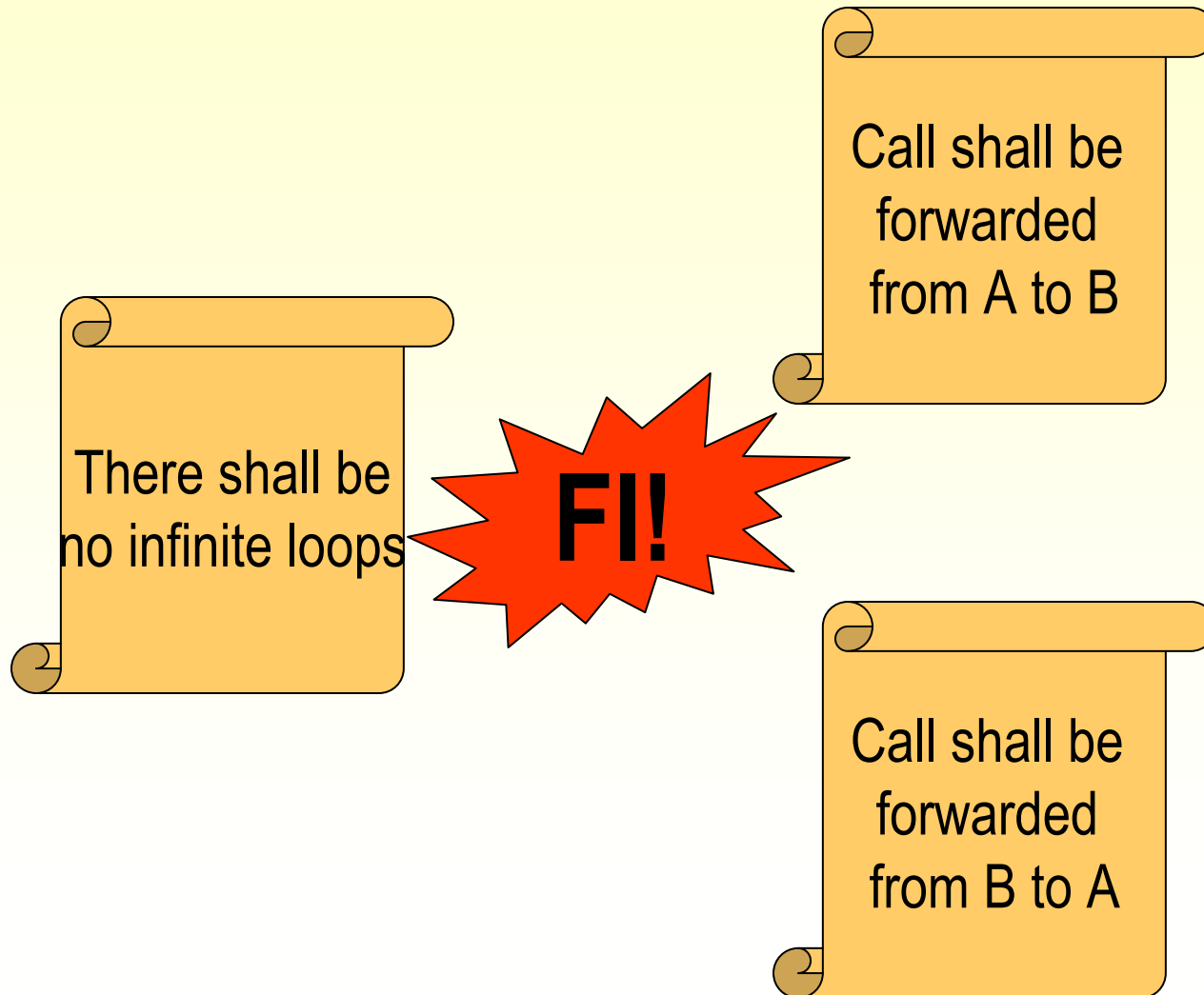
Aristotle, Metaphysics, IV, 6 (384 BC - 322 BC) (paraphrase)



Fundamental types

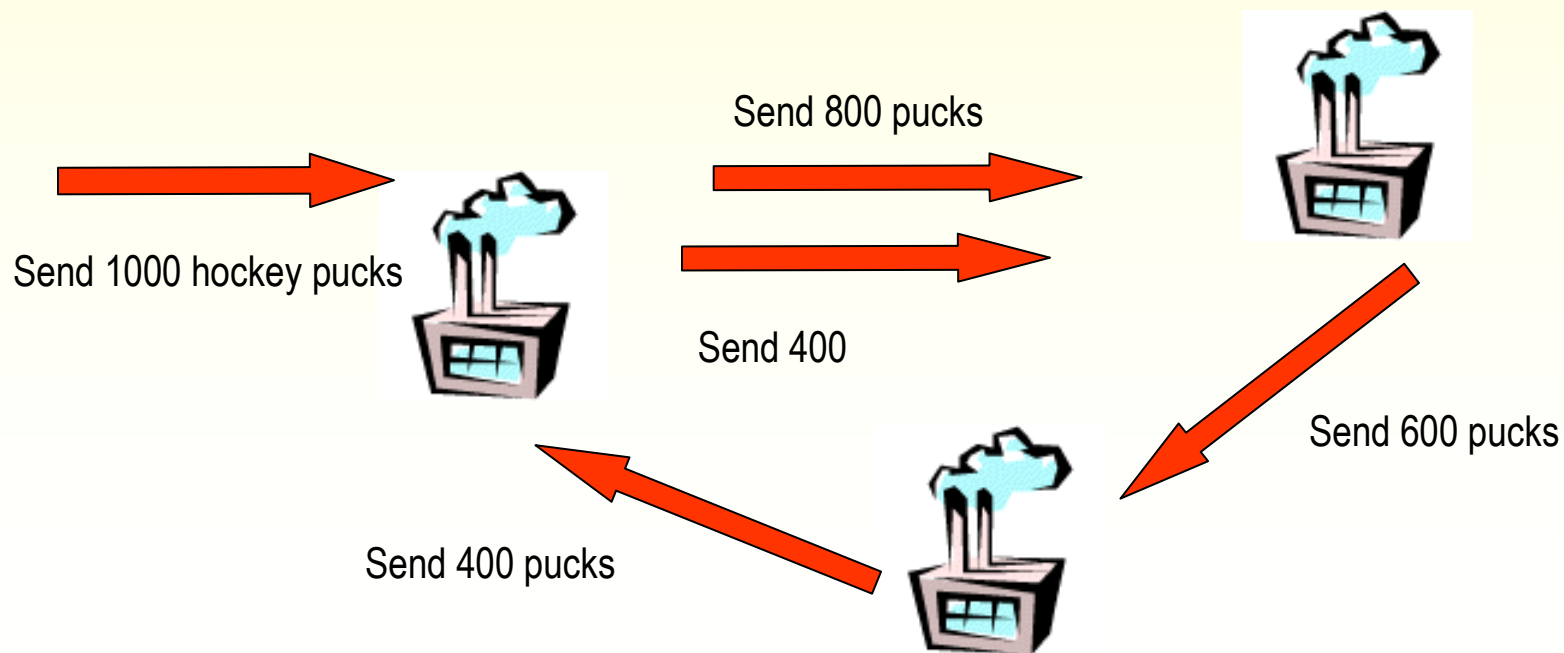
- Contradictions or inconsistency between feature of the same user, or of different users
- Contradictions or inconsistency between features when simultaneously activated
- Contradiction or inconsistency between features when sequentially activated
- Conflicts with systems axioms
 - ◆ E.g. there should be no unending loops

Infinite loops as FI



Infinite loops FIs

- Companies A, B and C have policies where each of them uses the next in a loop as suppliers of parts in excess of inventory
- This can start a chain reaction with potentially disastrous effects!



A QoS-related interaction: Automatic Call Distribution Systems

- Systems that are instructed to divert calls to others if there is overload
 - ◆ Similar mechanism!

Histoire du sujet de FI

- Le problème fut identifié vers le début des années '90 comme résultat des recherches liées à la conception de services IN
 - ◆ Surtout chez Bellcore (maintenant Telcordia)
- Huit Ateliers internationaux lui ont été dédiés, et des centaines d'articles
- Il y a eu aussi deux compétitions internationales, où les gagnants devaient trouver le plus grand nombre possible d'interactions dans un ensemble donné de fonctionnalités

From features to policies

- In Internet Telephony telecom devices are programmable
- They can be made to execute arbitrarily complex user *policies*
- The concept of policy generalizes the concept of feature
- Policy interactions generalize Feature interactions

Policies and Intentions

- Policies reflect user intentions
- However there are intentions that remain implicit
- Interactions between policies may violate user intentions, whether implicit or explicit

Will there still be FIs in VoIP?

- Consider the following situations:
 - ◆ a telephone ringsback simultaneously free and busy
 - ◆ one can dial a new call when hearing busy
 - ◆ one can get connected to someone in her black list
 - ◆ anyone can dial in to an existing conversation
 - ◆ an event under the same preconditions can give sometimes a result, other times another result
- If all this and more should be *tolerated* in VoIP, then no point looking for FI
- However user intentions are probably against several of these

CPL: a language for specifying policies

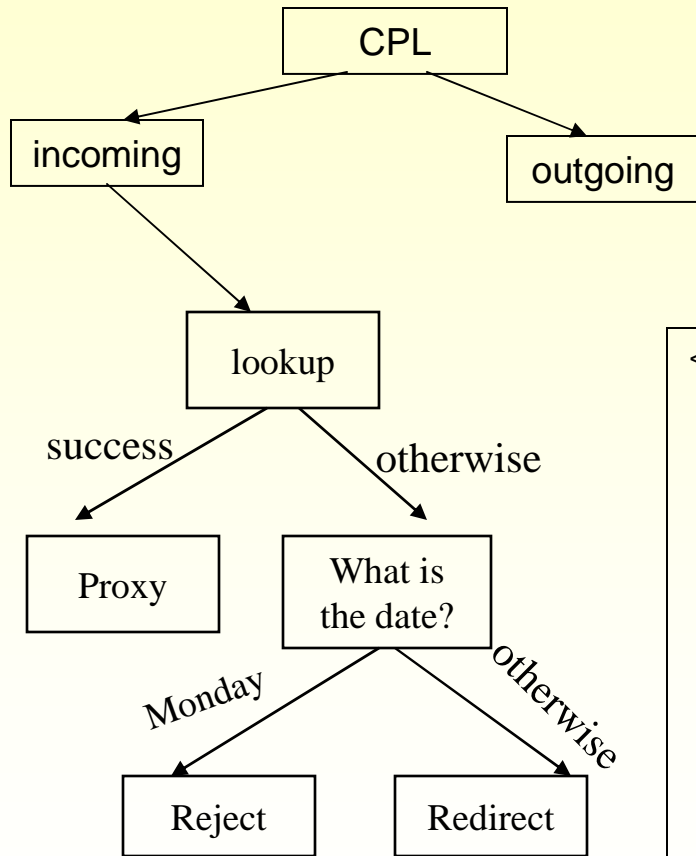
Call Processing Language

Very simple, but a taste of things to come

Thanks to Yiqun Xu
and Dongmei Jiang

an IETF RFP

CPL Structure



```
<cpl>
  <incoming>
    to execute for incoming calls
  </incoming>
  <outgoing>
    to execute for outgoing calls
  </outgoing>
</cpl>
```

```
<lookup source="registration">
  <success>
    <proxy/>
  </success>
  <otherwise>
    <time-switch>
      <time dtstart =20001001T000000" duration="24H"
        freq="weekly" byday="MO">
        <reject/>
      </time>
      <otherwise>
        <redirect/>
      </otherwise>
    </time-switch>
  </otherwise>
</lookup>
```

CPL Mode of Operation

- programmed in proxy
- intercept INVITE message
 - ◆ incoming and outgoing
- follow decision tree, based on message and/or environment values
 - ◆ address/time/priority/string switches
- execute action
 - ◆ proxy/redirect/reject
- optionally handle action response
 - ◆ proxy -> busy | no-answer

Caractéristiques de CPL

- Construit de façon à limiter les possibilités de programmation
 - ◆ N'est qu'une cascade de choix
 - ◆ Pas de boucles
 - ◆ Information très limitée sur l'état du système
 - ◆ Aucune mémoire du passé (stateless)
- Trop limité pour la programmation de fonctionnalités complexes, y inclus certaines bien établies
 - ◆ Appels conférence

Interaction de fonctionnalités en CPL

- Il est évidemment possible que des fonctionnalités spécifiées en CPL se trouvent en conflit!
- P. ex. le conflit entre OCS et CF et réalisable dans CPL
- Nous avons développé une approche logique pour la détection de ces conflits
 - ◆ Détection de conflits dans un seul CPL script: OK
 - ◆ Détection de conflit entre CPL scripts d'entités communicantes
 - Comment implémenter ceci?
 - Au moment de la connexion, il faut vérifier si la combinaison de scripts peut conduire à des dégâts?

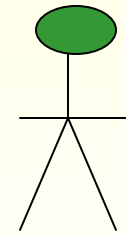
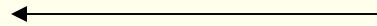
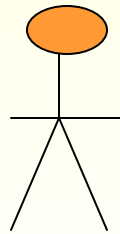
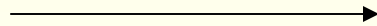
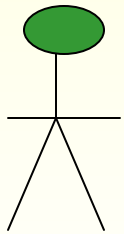
Extensions de CPL: présence

- Le système est capable de déterminer et transmettre des informations concernant la disponibilité des usagers
- Les usagers peuvent établir des politiques sur comment utiliser ces informations
- P.ex.
 - ◆ Abdel et moi voulons nous voir
 - ◆ Les agents de présence de nous deux s'échangeront des informations concernant nos mouvements pendant les heures de travail (avec notre permission!)
 - ◆ Abdel peut programmer son téléphone de façon qu'il m'appelle dès que j'arrive à son bâtiment

I want to know where Leila is all the time

I don't want my boss to know Where I am out of work hours
I allow my husband to know Where I am always

I want to know where Leila is all the time



Leila's Boss
Leila's **watcher**

Leila
Presentinity

Leila's husband
Leila's **watcher**

Quelques autres possibilités:

- Leila only accepts her boss's subscription requests from 9:00am to 5:00pm, Monday to Friday
- An automatic call to Leila is made as soon as Pierre is notified that Leila is in her office
- Leila blocks her calls to her boss when the boss is *unavailable* to take her calls (e.g. certain hours)
- Leila forwards her incoming-calls to her voice mail when she *unavailable* to communicate with others

Une minière d'interactions!

- Boss wants to know Leila's presence all the time, Leila wants boss to know it only in certain hours
 - ◆ This one can be easily solved by taking the intersection
 - ◆ But what if the intersection is empty
- What if Leila's boss is also her husband...
- What if
 - ◆ boss wants to talk to Leila as soon as she gets to office
 - ◆ but Leila has programmed her phone so that she does not receive calls within 30 mins of her arrival
- Or boss wants to send a message to all at noon but Leila has programmed phone so that she is not disturbed during lunch
- Or Leila and boss have programmed their phones to call each other as soon as they both are in the same building...

*Hell is nothing but the unforeseen
behavior of paradise...*

Luigi Logrippio, 2004

Comment traiter les IF

■ Off-line:

- ◆ Les IF sont trouvées et réglées au moment de la conception

■ On-line:

- ◆ les IF sont trouvées quand elles se produisent et sont réglées par un mécanisme dynamique

Noter la similarité avec le phénomène de l'interblocage ou impasse (deadlock)

Detecting and Handling FI at execution time

- Since each user will be able to define own features, and users can become connected arbitrarily, unpredictable FIs can occur during normal operation.
- Strategies must be developed to catch such FI before they have disastrous effects
 - ◆ Security breaches
 - ◆ Infinite loops
 - ◆ ...
- A difficult research problem

Some possible solutions

all problematic

- Feature scripts can be checked and compared at the time two users become connected
 - ◆ However this requires users to reveal their policies to the FI checker
- FI arbiters can be developed to detect FIs and intervene at the time of the interaction
- Negotiation process between parties, based on resolution policies
 - ◆ However how do we know that a FI is occurring?
 - ◆ What principles to use for arbitration and negotiation?
 - ◆ How do we insure that the process can be completed in milliseconds?

Résolution automatique de conflits

- Un problème impossible à résoudre dans le cas général, à cause de la grande variété de situations
- Peut souvent être faite en considération du contexte et de règles d'origine ergonomique et sociale, p.ex.
 - ◆ Dans un contexte d'entreprise, la règle du supérieur hiérarchique a la priorité
 - ◆ Dans le cas d'un appel entre paires, le droit de l'appelé de ne pas être dérangé doit être respecté
- Un certain nombre de règles de ce type devra être établi

Similarité avec la jurisprudence

- Au cours des siècles, un grand nombre de concepts et principes pour résoudre les conflits ont été développés par notre civilisation:
 - ◆ Concepts de famille, propriété, mariage, héritage
 - ◆ Et comment les conflits sont résolus dans chaque contexte
- Des concepts appropriés devront être développés dans notre domaine

Other places where will FI lurk

- Firewalls: contradicting clauses
- Access Control: contradicting rules on who can access which information for which purpose
 - ◆ See XACML language
- Routers: contradicting configuration rules

The world of web services

FIs galore with a vengeance!

- A phone can ring wrongly without much harm, but the purchase of an expensive item can't be cancelled as easily!
- Forwarding loops: much worse in effects and prevention!
 - ◆ E.g. loops of subcontracts can lead to disastrous economic effects
- Interactions in contracts: policies of different users clash, thus making certain contracts impossible, perhaps for futile reasons...
- Security gaps in access control

Extensions au commerce électronique

- Le commerce électronique sera un grand domaine d'applications de ces mécanismes
- Les personnes pourront déléguer la partie 'recherche' de leur magasinage à des agents automatiques qui seront fournis de politiques
- En fonction des politiques des différents agents, certaines correspondances (matches) pourront être établies ou exclues

Interaction de fonctionnalités (Waël Hassan)

Client Policy

- Buy Product P
 - Merchant can **not sell** private info
 - ◆ Credit card info
 - ◆ Name & Address
 - Merchant can **retain** customer info
 - ◆ Credit card info
 - ◆ Name & Address
- for **3 Weeks** after Purchase

Merchant Policy

- Sell Product P
 - ◆ But **subcontract Delivery to Y**
- Information required from customer sale related:
 - ◆ Credit card info
 - ◆ Name & Address
- Privacy policy, we will
 - ◆ **Not sell** customer information to thirds
 - ◆ **Retain** Information for **3 Weeks**

Company Y (DeliverProducts.com)

- A. Deliver product P
- B. **Retains** customer information for **10 Weeks**

Scenario

1. Client sends information to merchant
2. Rules of client and merchant for the sale will not contradict.
3. However merchant will proxy to Y
4. Y will retain the information for **10 Weeks** rather than **3**
5. **How to protect client's policy**

Note similarity with
OCS/CF example!

Policy interactions in contracts

- One airline's ticket change policies (taken from different texts):
 - ◆ R1: Changes permitted up to one day before departure
 - ◆ R2: Changes or cancellations must be done by calling Reservations at least two hours bef. departure
 - ◆ R3: No changes or upgrades permitted on day of departure
- Event: passenger calls Reservations on day of departure 2hrs before
- Inconsistency: change is
 - ◆ Allowed: R2
 - ◆ Disallowed: R1 and R3
- There are other possible inconsistencies in these three rules

Automatically generated contracts

- We can expect that in the future contracts will be automatically generated case by case according to patterns and situations
- Some contracts will have short lives, maybe seconds

Must be automatically tested for consistency

Possible application area:

SLA, Service-Level Agreements

Intégration du monde des politiques et du monde de la loi

- The human world of telecom and E-Commerce is regulated by laws and regulations
- Their electronic world is populated by agents that follow *policies*
- Agents engage increasingly in legal behavior, e.g.
 - they negotiate and conclude contracts
 - they can be in conflict and can be penalized
 - their penalties will affect humans
- Policies must abide the law
- The FI picture is now part of the expanded and integrated context of conflicts of agent policies and human law

Where are we heading

- In the information society real people and automatic agents will have interchangeable roles
- Laws and policies will have to be seamlessly integrated
- Their conflict resolution mechanisms will have to be seamlessly integrated
- Changes in laws should result in immediate changes in programs
- **Osmosis between machine and human world**

Technological Context

- On the law side, research is continuing in AI methods to (partially) automate logical deduction from laws to legal decisions, to solve human conflicts
- The related topic in computing is the Feature Interaction problem:
 - Agents being directed by policies to do conflicting things
 - Conflicts between agents doing different things
 - Conflicts between different levels of regulations for an agent
 - Possibly leading to malfunctions or unexpected results
 - Automatic conflict-resolution mechanisms may trigger in such situations

Executable laws

- Laws and regulations expressed as logic programs are understood by the agents and executed
- Conflicts can be detected and solved
 - at design time
 - or (more difficult) at execution time
- Conflict-resolution mechanisms will draw the consequences of laws, policies, and regulations and will resolve conflicts **in milliseconds**
 - Using automatic deduction

What is the glue?

- What can keep it all together?
- The glue is
 - very old **logic**, and
 - old **logic programming**
- Laws, regulations, policies, programs can be cast in the unifying language of logic and logic programming
 - which may include logic-based agent languages

Containing Inconsistencies

- According to classical logic,
 - ◆ a database that has two contradicting entries is all false,
 - ◆ and a game that has a couple of contradictory rules has no rules
- But in practice contradictions can be contained
- Logic systems that model this reasoning have been developed

*Do I contradict myself?
Very well, then, I contradict myself.
I am large, I contain multitudes."
-- Walt Whitman, Song of Myself*

Alloy

An Interesting language and tool

- Alloy is a software modeling language which is a subset of Z
 - ◆ First order logic
- Also similar to UML-OCL
- Policies can be automatically translated into Alloy and automatically analyzed
- Alloy verifier translates everything into a boolean formula which it tries to satisfy
- It may then come back with a counterexample
- Alloy results can be used for further decisions

Uses of Alloy in our group

- Validation of XACML access control policies
- Delegation and separation of concerns examples
- Airline contract example

Exemple – Politique XACML

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy PolicyId="OperationFichiersNotes" RuleCombiningAlgId="deny-overrides">
  <Description>
    Potilitiques concernant les accès aux fichiers des notes des étudiants
  </Description>
  <Target>
    <Subject> <AnySubject/> </Subject>
    <Resources> <AnyResource/> </Resources>
    <Actions> <AnyAction/> </Actions>
  </Target>
  <Rule RuleId="regle1" Effect="Permit">
    <Description>
      Un professeur peut lire et modifier tous les fichiers de notes de tous les cours qu'il enseigne
    </Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="function:string-equal">
            <AttributeValue DataType="string">Professeur</AttributeValue>
            <SubjectAttributeDesignator AttributeId="role-sujet" DataType="string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources> <AnyResource/> </Resources>
      <Actions> <AnyAction/> </Actions>
    </Target>
    <Condition>
      <Apply FunctionId="function:string-is-in">
        <Apply FunctionId="string-one-and-only">
          <ResourceAttributeDesignator AttributeId="cours" DataType="string"/>
        </Apply>
        <SubjectAttributeDesignator AttributeId="cours-enseignes" DataType="string"/>
      </Apply>
    </Condition>
  </Rule>
</Policy>
```

Dans un format plus... humain

Règle 1

Identificateur de la règle: regle1

Effet: Permet

Description :

Un professeur peut lire, et modifier tous les fichiers de notes de tous les cours qu'il enseigne

Cible :

- ◆ **Sujet :**
 - ◊ Si attribut **role-sujet** correspond à la valeur **Professeur**
- ◆ **Ressource :** Toutes les ressources
- ◆ **Action :** Toutes les actions

Condition :Fonction:

Paramètres:

- ◆ **Fonction:** urn:oasis:names:tc:xacml:1.0:function:string-is-in

Paramètres:

- ◊ **Fonction:** urn:oasis:names:tc:xacml:1.0:function:string-one-and-only

Paramètres:

- Ressource.cours
- ◊ Sujet.cours-enseignes

Exemple : Règles

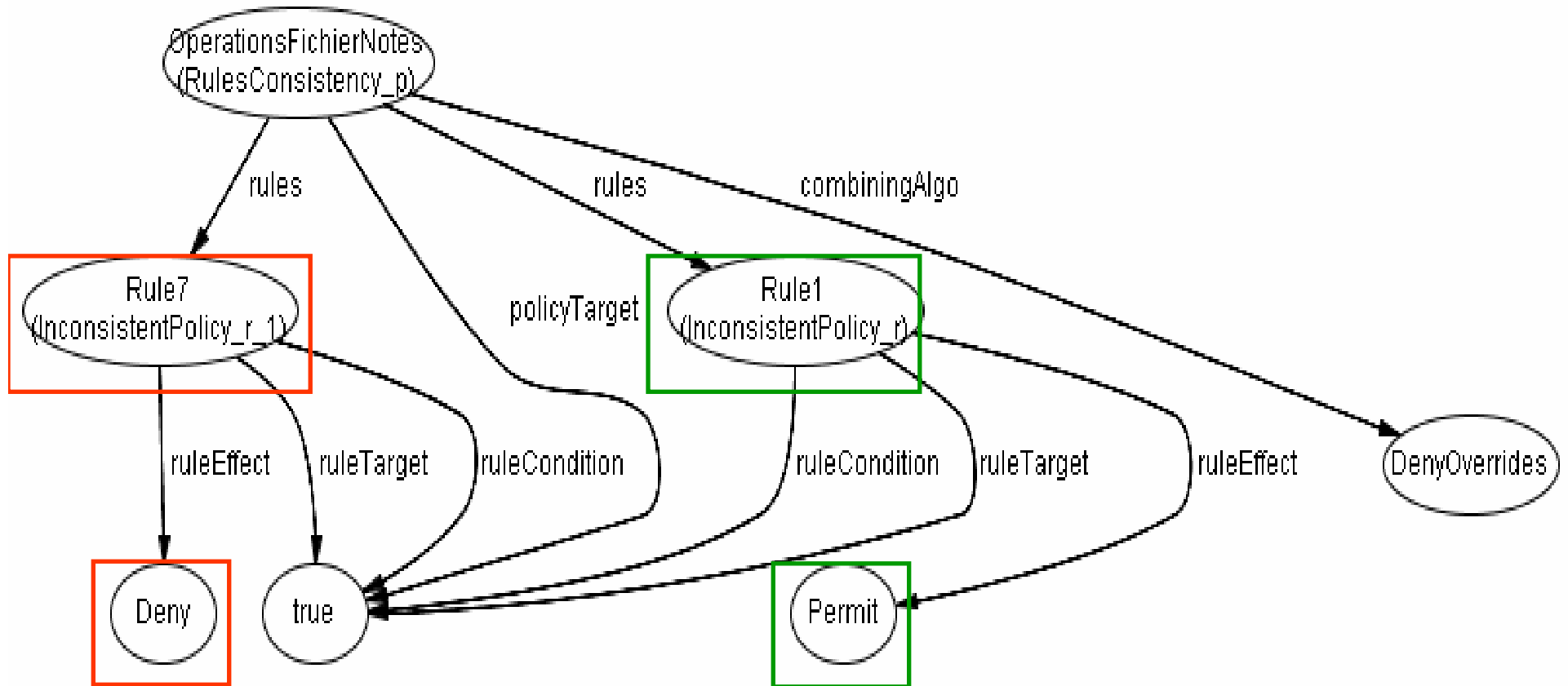
■ Permit

- ◆ (Professeur, lire ou modifier, fichier de notes) si le professeur enseigne le cours concerné
- ◆ (Étudiant, lire, fichier de notes) si l'étudiant est le propriétaire
- ◆ (Personnel, lire, fichier de notes)

■ Deny

- ◆ (Professeur, lire ou modifier, fichier de notes) si le professeur n'enseigne pas le cours concerné
- ◆ (Étudiant, lire, fichier de notes) si l'étudiant n'est pas le propriétaire
- ◆ (Étudiant ou Personnel, modifier, fichier de notes)

Alloy trouve la contradiction



Alloy découvre qu'il n'y a pas de règle qui force prof != étudiant donc un étudiant qui est aussi prof peut simultanément avoir et ne pas avoir certains droits

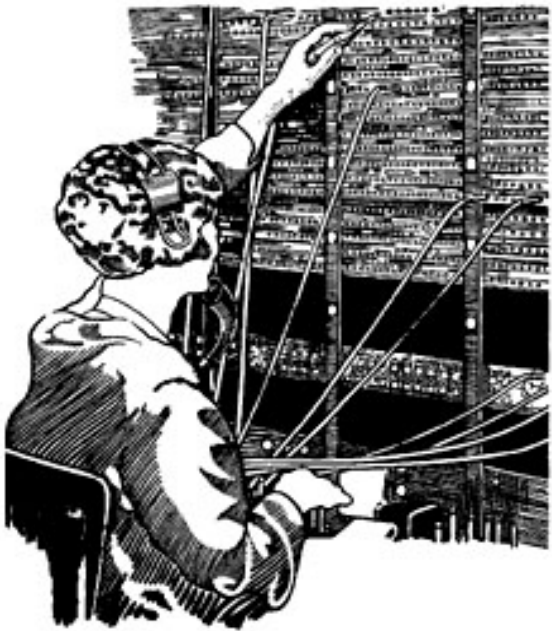
A fertile research area

- Many interesting research topics at the crossroads of
 - information society
 - human law and legal theory
 - computer programming and software engineering
- In the playfield of logic

Conclusion

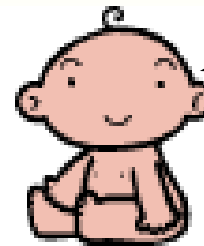
- Features and FI belong to a complex human, legal, and logical picture
- They are likely to occur in complex systems, leading to malfunctions and security breaches
- Their identification and repair is a complex research topic

*Comme dans les films,
on coupe quand on ne sait pas comment en
sortir...*



Cord switchboard

FIN



Mais non, ce
n'est que le
début...