NICTA

Law, Logic and Business Processes

Guido Governatori

RELaw 2010, Sydney, September 28, 2010

Australian Government

Department of Broadband, Communications and the Digital Economy

Australian Research Council

NICTA Members

ANU THE AUSTRALIAN NATIONAL UNIVERSITY

UNSW THE UNIVERSITY OF NEW SOUTH WALES

NSW Department of State and Regional Development

NICTA Partners

Victoria The Place To Be

THE UNIVERSITY OF MELBOURNE

The University of Sydney

Queensland Government

Griffith UNIVERSITY

QUT

The University Of Queensland AUSTRALIA

Part I

Introduction

What's compliance?

Ensuring that business operations, processes, and practices are in accordance with a given prescriptive (often legal) document

What's compliance?

NICTA

Ensuring that business operations, processes, and practices are in accordance with a given prescriptive (often legal) document

Regulatory

- Basel II
- Sarbanes-Oxley
- OFAC (USA Patriot Act)
- OSFI "blocked entity" lists
- HIPAA
- Graham-Leach-Bliley

Standards

- Best practice models
- SAP solution maps
- ISO 9000
- Medical guidelines

Contracts

- Service Agreement
- Customer Contract
- Warranty
- Insurance Policy
- Business Partnership

How to ensure compliance?

How to ensure compliance?

NICTA

Compliance is a relationship between two sets of specifications

How to ensure compliance?

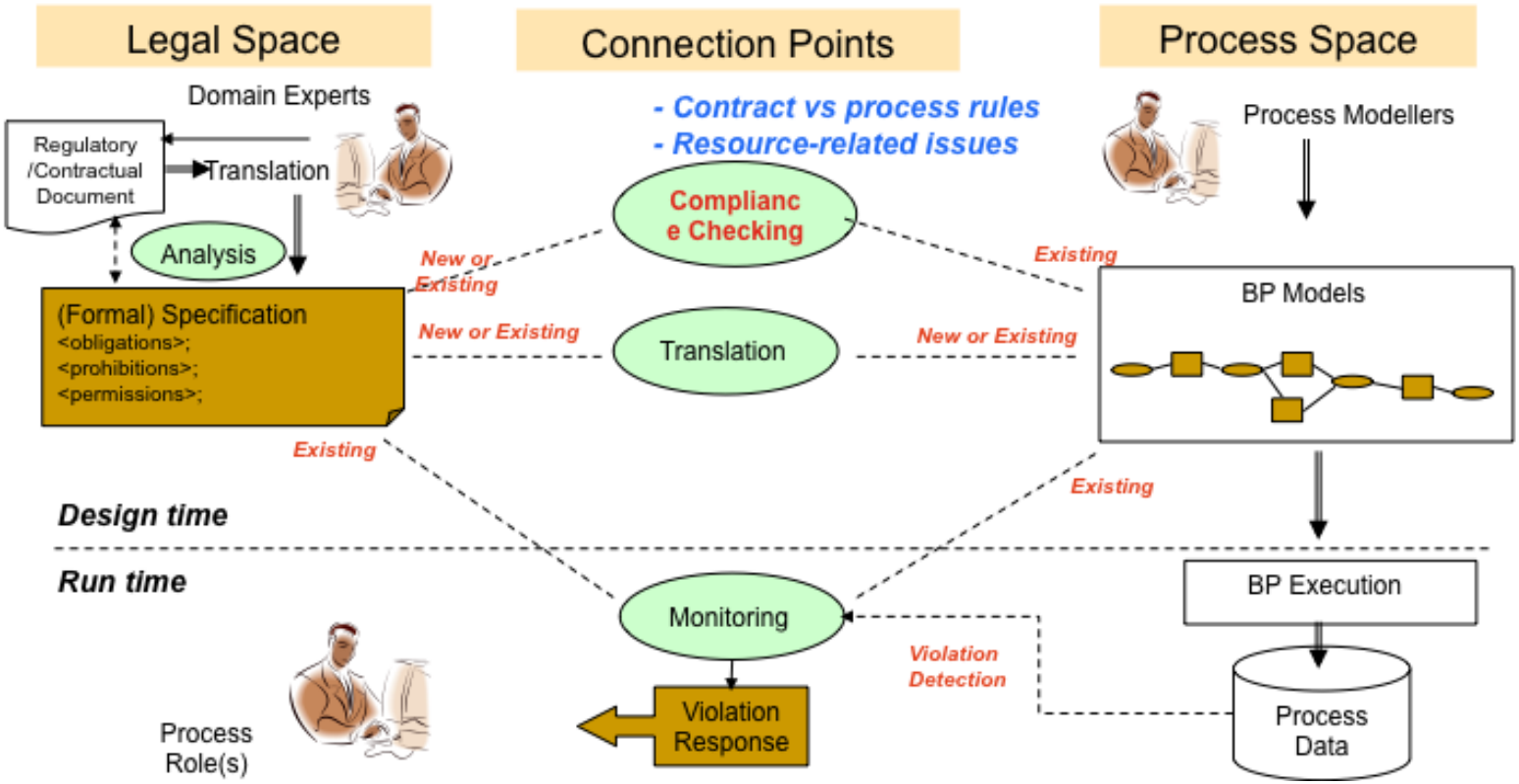
NICTA

Compliance is a relationship between two sets of specifications

Alignment of formal specifications for business processes and formal specifications for prescriptive (legal) documents.

Compliance Ecosystem

NICTA



Ensuring Compliance

- Ensuring that business processes are in accordance with given prescriptive rules
- Compliance is a relationship between two sets of specifications
- Alignment of formal specifications for business processes and formal specifications for prescriptive rules

Ensuring Compliance

- Ensuring that business processes are in accordance with given prescriptive rules
- Compliance is a relationship between two sets of specifications
- Alignment of formal specifications for business processes and formal specifications for prescriptive rules
- Ensuring that business processes are compliant requires a suitable language for expressing normative specifications in such a way as

Ensuring Compliance

- Ensuring that business processes are in accordance with given prescriptive rules
- Compliance is a relationship between two sets of specifications
- Alignment of formal specifications for business processes and formal specifications for prescriptive rules
- Ensuring that business processes are compliant requires a suitable language for expressing normative specifications in such a way as
  - we can identify formal loopholes, deadlocks and inconsistencies in normative systems, and

Ensuring Compliance

- Ensuring that business processes are in accordance with given prescriptive rules
- Compliance is a relationship between two sets of specifications
- Alignment of formal specifications for business processes and formal specifications for prescriptive rules
- Ensuring that business processes are compliant requires a suitable language for expressing normative specifications in such a way as
  - we can identify formal loopholes, deadlocks and inconsistencies in normative systems, and
  - we can make hidden conditions explicit

Ensuring Compliance

- Ensuring that business processes are in accordance with given prescriptive rules
- Compliance is a relationship between two sets of specifications
- Alignment of formal specifications for business processes and formal specifications for prescriptive rules
- Ensuring that business processes are compliant requires a suitable language for expressing normative specifications in such a way as
  - we can identify formal loopholes, deadlocks and inconsistencies in normative systems, and
  - we can make hidden conditions explicit

Without this, we do not have any guarantee that a given business process is compliant, because we do not know if all relevant norms have been considered

Part II

Norms and Logic

Artificial Intelligence and Law: Goals

NICTA

- Contribute to jurisprudence/cognitive science/AI
- Improve the training and skill of lawyers
  - More careful reading of legal materials
  - More precise drafting of legal documents
  - More rational management of risk
  - More efficient management of information
- Provide a fairer and more efficient system of justice
  - Reduce high transaction cost of legal services
  - Make it easier to treat like cases alike
  - Facilitate alternative dispute resolution
  - Advance public understanding of the law and legal system
- Avoid potential for abuse:
  - Computer programs should be tools for legal decision makers; they should not make the decisions.

Artificial Intelligence and Law: Grand Challenges

NICTA

- Representing legislation for both inference and maintenance
- Representing and reasoning with open-textured concepts
- Representing and reasoning with normative concepts
- Simulating the process of expert legal prediction/advising
- Reasoning and arguing using examples as well as rules
- Understanding and generating legal texts

Artificial Intelligence and Law: Grand Challenges

- Representing legislation for both inference and maintenance
- Representing and reasoning with open-textured concepts
- Representing and reasoning with normative concepts
- Simulating the process of expert legal prediction/advising
- Reasoning and arguing using examples as well as rules
- Understanding and generating legal texts

Formalising legislation using logic

The Basic Structure of Norms

The Basic Structure of Norms

NICTA

$$\text{if } A_1, \ldots, A_n \text{ then } B$$

where

- $A_1, \ldots, A_n$ are the applicability conditions of the norm, and
- B is the legal effect

The Basic Structure of Norms

$$\text{if } A_1, \ldots, A_n \text{ then } B$$

where

- $A_1, \ldots, A_n$ are the applicability conditions of the norm, and
- B is the legal effect

Refinements

The Basic Structure of Norms

NICTA

$$\text{if } A_1, \ldots, A_n \text{ then } B$$

where

- $A_1, \ldots, A_n$ are the applicability conditions of the norm, and
- B is the legal effect

Refinements

Rules Types

1. constitutive rules
2. technical rules
3. prescriptions

The Basic Structure of Norms

NICTA

$$\text{if } A_1, \ldots, A_n \text{ then } B$$

where

- $A_1, \ldots, A_n$ are the applicability conditions of the norm, and
- B is the legal effect

Refinements

Rules Types

1. constitutive rules
2. technical rules
3. prescriptions

Prescriptions: content

- the norm-subjects
- the action-theme
- the conditions of application
- the nature of guidance

Which Semantics?

NICTA

Contract fragment

3.1 A "Premium Customer" is a customer who has spent more that $10000 in goods.

3.2 Services marked as "special order" are subject to a 5% surcharge. Premium customers are exempt from special order surcharge.

5.2 The (Supplier) shall on receipt of a purchase order for (Services) make them available within one day.

5.3 If for any reason the conditions stated in 4.1 or 4.2 are not met the (Purchaser) is entitled to charge the (Supplier) the rate of $100 for each hour the (Service) is not delivered.

What Semantics?

NICTA

Ambiguity and Open Texture

Vehicles are not permitted in this park (H.L.A. Hart 1958)

What Semantics?

NICTA

Ambiguity and Open Texture

Vehicles are not permitted in this park (H.L.A. Hart 1958)

- Are baby carriages prohibited?
- Are tricycles prohibited?
- Are 10 speed bikes prohibited?
- Are 1000 cc Harley Davidson motorcycles prohibited?
- Is a functioning tank prohibited for ANZAC Day Parade?

Requirements for Representation of Norms (1)

NICTA

- **Isomorphism.** One-to-one correspondence between rules in the formal model and the units of natural languaguage text
- **Reification.** Rules are objects with properties (jurisdiction, authority, temporal properties)
- **Rule semantics.** Need for a rigorous semantics for correctly computing the legal effects
- **Defeasibility.**
  - **Conflicts:** exceptions, rules with different ranking status, rules enacted at different times
  - **Exclusionary rules:** rules explicitly providing conditions to make other rules inapplicable
  - **Contraposition:** Rules do not counterpose
- **Contributory reasons or factors.** "The educational value of a work needs to be taken into consideration when evaluating whether the work is covered by the copyright doctrine of fair use."

Requirements for Representation of Norms (2)

- **Rule validity.** Rules can be invalid or become invalid
- **Legal procedures.** Burden of proof; detecting violations of the law; legal compliance
- **Normative effects.** Many normative effects:
  - **Evaluative:** "Human dignity is valuable"
  - **Qualificatory:** "x is a citizen"
  - **Definitional:** "adult means a 18 year old person or older"
  - **Deontic:** "x has the obligation to do A"
  - **Potestative:** "A worker has the power to terminate his work contract"
  - **Evidentiary:** "It is presumed that dismissal was discriminatory"
  - **Existential:** "The company ceases to exist"
  - **Norm-concerning effects:** abrogation, repeal, substitution
- **Persistence of normative effects**
  - "If one causes a damage, one has to provide compensation"
  - "If one is in a public office, one is forbidden to smoke"
- **Values and goals**

Key components of Normative Systems

NICTA

A normative system is a set of clauses

- Definitional clauses (counts-as rules)
- Prescriptive clauses (norms)
  - obligations
  - permissions
  - prohibitions
  - violations

Formalising Compliance

NICTA

- Clausal forms (Sergot-Kowalski, Ghose-Koliadis, Hoffman-Weber-Governatori)

Formalising Compliance

NICTA

- Clausal forms (Sergot-Kowalski, Ghose-Koliadis, Hoffman-Weber-Governatori)
- Temporal logic (van der Aalst, Rindle-Ma)

Formalising Compliance

NICTA

- Clausal forms (Sergot-Kowalski, Ghose-Koliadis, Hoffman-Weber-Governatori)
- Temporal logic (van der Aalst, Rindle-Ma)
- Event Calculus (Goedertier, Singh, Farrell-Sergot, DEIS)

Formalising Compliance

NICTA

- Clausal forms (Sergot-Kowalski, Ghose-Koliadis, Hoffman-Weber-Governatori)
- Temporal logic (van der Aalst, Rindle-Ma)
- Event Calculus (Goedertier, Singh, Farrell-Sergot, DEIS)
- Deontic Logic (Governatori-Milosevic-Sadiq, Goedertier, Liu)

Clausal Approaches

NICTA

$$p \vee q \vee r \vee s$$

- do not distinguish different normative positions

Clausal Approaches

NICTA

$$p \vee q \vee r \vee s$$

- do not distinguish different normative positions
- directionality of norms (what are the premises and what is the conclusion)

Clausal Approaches

NICTA

$$p \vee q \vee r \vee s$$

- do not distinguish different normative positions
- directionality of norms (what are the premises and what is the conclusion)
- semantic compliance (annotations)

Temporal Logic

NICTA

$$p \rightarrow A\Box q$$

and model checking

- Temporal logic and model checking have been used to verification of software and hardware systems

Temporal Logic

NICTA

$$p \rightarrow A\Box q$$

and model checking

- Temporal logic and model checking have been used to verification of software and hardware systems
- Mature technology

Temporal Logic

NICTA

$$p \rightarrow A\Box q$$

and model checking

- Temporal logic and model checking have been used to verification of software and hardware systems
- Mature technology
- Structural Compliance only

Temporal Logic

NICTA

$$p \rightarrow A\Box q$$

and model checking

- Temporal logic and model checking have been used to verification of software and hardware systems
- Mature technology
- Structural Compliance only
- Does not distinguish normative positions

Temporal Logic

NICTA

$$p \rightarrow A\Box q$$

and model checking

- Temporal logic and model checking have been used to verification of software and hardware systems
- Mature technology
- Structural Compliance only
- Does not distinguish normative positions
- Standard Deontic Logic can be simulated in Temporal Logic

Temporal Logic

NICTA

$$p \rightarrow A\Box q$$

and model checking

- Temporal logic and model checking have been used to verification of software and hardware systems
- Mature technology
- Structural Compliance only
- Does not distinguish normative positions
- Standard Deontic Logic can be simulated in Temporal Logic
- Permissions must always be instantiated

Temporal Logic

NICTA

$$p \rightarrow A\Box q$$

and model checking

- Temporal logic and model checking have been used to verification of software and hardware systems
- Mature technology
- Structural Compliance only
- Does not distinguish normative positions
- Standard Deontic Logic can be simulated in Temporal Logic
- Permissions must always be instantiated
- Does not scale to real life examples

Deontic Logic

NICTA

Extension of logic with the operators OBL and PER.

- $\mathit{SpecialOrderPrice}(x) = \mathit{Price}(x) + 5\%$
- $\mathrm{OBL}_{\mathit{Supplier}}\, \mathit{MakeGoodsAvailble1Day}$
- $\mathrm{PER}_{\mathit{Purchaser}}\, \mathit{ChargeSupplier}$

Standard Deontic Logic

NICTA

Extension of classical logic with the modal operators OBL and PER.

- $\mathrm{OBL}\alpha \equiv \neg\mathrm{PER}\neg\alpha, \quad \mathrm{PER}\alpha \equiv \neg\mathrm{OBL}\neg\alpha$
- $\mathrm{OBL}(\alpha \to \beta) \to (\mathrm{OBL}\alpha \to \mathrm{OBL}\beta)$
- $\mathrm{OBL}\alpha \to \mathrm{PER}\alpha$ or $(\mathrm{OBL}\alpha \to \neg\mathrm{OBL}\neg\alpha)$

Standard Deontic Logic

NICTA

Extension of classical logic with the modal operators OBL and PER.

- $\mathrm{OBL}\alpha \equiv \neg\mathrm{PER}\neg\alpha, \quad \mathrm{PER}\alpha \equiv \neg\mathrm{OBL}\neg\alpha$
- $\mathrm{OBL}(\alpha \to \beta) \to (\mathrm{OBL}\alpha \to \mathrm{OBL}\beta)$
- $\mathrm{OBL}\alpha \to \mathrm{PER}\alpha$ or $(\mathrm{OBL}\alpha \to \neg\mathrm{OBL}\neg\alpha)$

Standard Deontic Logic is not able to deal with violations

Violation paradox

NICTA

Rules for RELaw Presentations

- Guido should not tell lies in his presentation
- If Guido tells a lie then he has to explain why
- It ought to be the case that if Guido does not tell a lie then he does not explain why
- Guido tells lies in his presentation

Violation paradox

NICTA

Rules for RELaw Presentations

- Guido should not tell lies in his presentation
- If Guido tells a lie then he has to explain why
- It ought to be the case that if Guido does not tell a lie then he does not explain why
- Guido tells lies in his presentation
- $\mathrm{OBL}\neg \mathit{lie}$
- $\mathit{lie} \rightarrow \mathrm{OBL}\mathit{explain}$
- $\mathrm{OBL}(\neg \mathit{lie} \rightarrow \neg \mathit{explain})$
- lie

Violation paradox

NICTA

Rules for RELaw Presentations

- Guido should not tell lies in his presentation
- If Guido tells a lie then he has to explain why
- It ought to be the case that if Guido does not tell a lie then he does not explain why
- Guido tells lies in his presentation
- $\mathrm{OBL}\neg \mathit{lie}$
- $\mathit{lie} \rightarrow \mathrm{OBL}\mathit{explain}$
- $\mathrm{OBL}(\neg \mathit{lie} \rightarrow \neg \mathit{explain})$
- lie

$\mathrm{OBL}\mathit{explain}$ and $\mathrm{OBL}\neg\mathit{explain}$

What's the problem?

What's the problem?

NICTA

$$a \Rightarrow Ob$$

$$\neg b \Rightarrow Oc$$

$$\neg c \Rightarrow Od$$

What's the problem?

NICTA

$$a \Rightarrow Ob$$
$$\neg b \Rightarrow Oc$$
$$\neg c \Rightarrow Od$$

What about a, $\neg b$ but d?

What's the problem?

NICTA

$$a \Rightarrow Ob$$
$$\neg b \Rightarrow Oc$$
$$\neg c \Rightarrow Od$$

What about a, $\neg b$ but d?

A logic of violations

Logic of Violations

NICTA

1. A (normative) prescriptive clause is represented by a rule $A_1, \ldots, A_n \vdash \mathbf{X}B$.

Logic of Violations

NICTA

1. A (normative) prescriptive clause is represented by a rule $A_1, \ldots, A_n \vdash \mathbf{X}B$.
2. A violation does not exist without an obligation it violates.

Logic of Violations

NICTA

1. A (normative) prescriptive clause is represented by a rule $A_1, \ldots, A_n \vdash \mathbf{X}B$.
2. A violation does not exist without an obligation it violates.
3. A reparation of a violation does not exist without a violation it repairs.

Logic of Violations

NICTA

1. A (normative) prescriptive clause is represented by a rule $A_1, \ldots, A_n \vdash \mathbf{X}B$.
2. A violation does not exist without an obligation it violates.
3. A reparation of a violation does not exist without a violation it repairs.
4. A reparation can be an obligation itself, and thus it can be violated.

Logic of Violations

NICTA

1. A (normative) prescriptive clause is represented by a rule $A_1, \ldots, A_n \vdash \mathbf{X}B$.
2. A violation does not exist without an obligation it violates.
3. A reparation of a violation does not exist without a violation it repairs.
4. A reparation can be an obligation itself, and thus it can be violated.
5. Permissions cannot be violated.

Logic of Violations

NICTA

1. A (normative) prescriptive clause is represented by a rule $A_1, \ldots, A_n \vdash \mathbf{X}B$.
2. A violation does not exist without an obligation it violates.
3. A reparation of a violation does not exist without a violation it repairs.
4. A reparation can be an obligation itself, and thus it can be violated.
5. Permissions cannot be violated.

- Prescriptive clauses cannot be taken in isolation.

Logic of Violations

NICTA

1. A (normative) prescriptive clause is represented by a rule $A_1, \ldots, A_n \vdash \mathbf{X}B$.
2. A violation does not exist without an obligation it violates.
3. A reparation of a violation does not exist without a violation it repairs.
4. A reparation can be an obligation itself, and thus it can be violated.
5. Permissions cannot be violated.

- Prescriptive clauses cannot be taken in isolation.
- It is possible to have chains of obligations/violations

Logic of Violations

NICTA

1. A (normative) prescriptive clause is represented by a rule $A_1, \ldots, A_n \vdash \mathbf{X}B$.
2. A violation does not exist without an obligation it violates.
3. A reparation of a violation does not exist without a violation it repairs.
4. A reparation can be an obligation itself, and thus it can be violated.
5. Permissions cannot be violated.

- Prescriptive clauses cannot be taken in isolation.
- It is possible to have chains of obligations/violations
- New prescriptive clauses can be derived from the given prescriptive clauses

Modelling Norms

NICTA

Norms are modelled as rules in FCL.

Language

- literals $p, q, \ldots$ (atomic proposition and their negation)
- deontic literals Op (Obligatory p), P (Permitted p), Fp (Forbidden p, i.e., $O\neg p$.)

Rules

- Normal rules

$$A_1, \ldots, A_n \Rightarrow OB$$

  $A_1 \ldots, A_n$ trigger the obligation of B.

- Rules for violations

$$A_1, \ldots, A_n \Rightarrow OB_1 \otimes OB_2 \otimes OB_3 \otimes \cdots \otimes OB_n$$

  $A_1 \ldots, A_n$ trigger the obligation of B_1 but if B_1 is violated then B_2 is obligatory and so on.

FCL Normal Forms

NICTA

- A ‘cleaned-up’ version of the FCL specifications
- Related clauses are merged, in particular linking original clauses and reparation clauses
- Removing redundancies, in particular clauses that are subsumed by other clauses
- Detecting and resolving conflicts
- NFCL form is used for compliance checking. NFCL forms describe behavioural and state space of contract

Part III

BPM Compliance

Business Process Model (BPM)

Business Process Model (BPM)

NICTA

- A BPM describes the tasks to be executed (and the order in which they are executed) to fulfil some objectives

Business Process Model (BPM)

NICTA

- A BPM describes the tasks to be executed (and the order in which they are executed) to fulfil some objectives
- A language for BPM usually has two elements:

Business Process Model (BPM)

NICTA

- A BPM describes the tasks to be executed (and the order in which they are executed) to fulfil some objectives
- A language for BPM usually has two elements:
  - Tasks are activities to be performed

Business Process Model (BPM)

NICTA

- A BPM describes the tasks to be executed (and the order in which they are executed) to fulfil some objectives
- A language for BPM usually has two elements:
  - Tasks are activities to be performed
  - Connectors consist of

Business Process Model (BPM)

NICTA

- A BPM describes the tasks to be executed (and the order in which they are executed) to fulfil some objectives
- A language for BPM usually has two elements:
  - Tasks are activities to be performed
  - Connectors consist of
    - sequence (a task is performed after another task),

Business Process Model (BPM)

NICTA

- A BPM describes the tasks to be executed (and the order in which they are executed) to fulfil some objectives
- A language for BPM usually has two elements:
  - Tasks are activities to be performed
  - Connectors consist of
    - sequence (a task is performed after another task),
    - parallel—and-split and and-join—(tasks are to be executed in parallel),

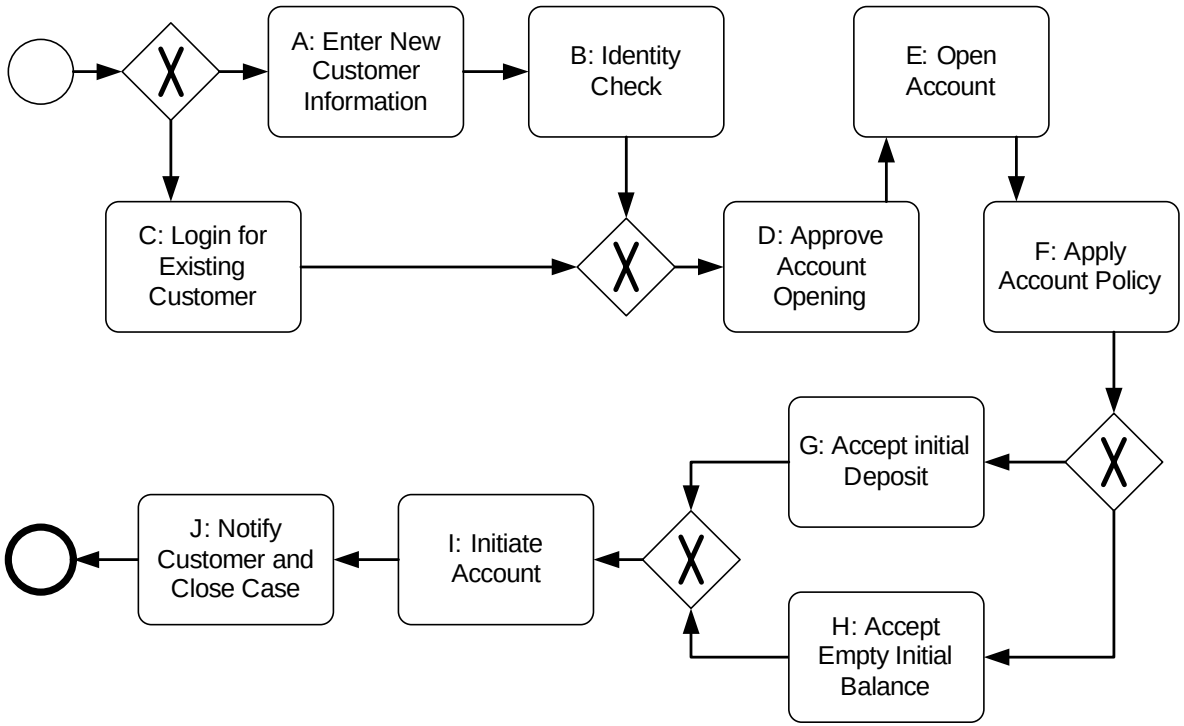
Business Process Model (BPM)

NICTA

- A BPM describes the tasks to be executed (and the order in which they are executed) to fulfil some objectives
- A language for BPM usually has two elements:
  - Tasks are activities to be performed
  - Connectors consist of
    - sequence (a task is performed after another task),
    - parallel—and-split and and-join—(tasks are to be executed in parallel),
    - choice—(x)or-split and (x)or-join—(at least (most) one task in a set of task must be executed).

Example: Account Opening Process

NICTA



Execution Traces

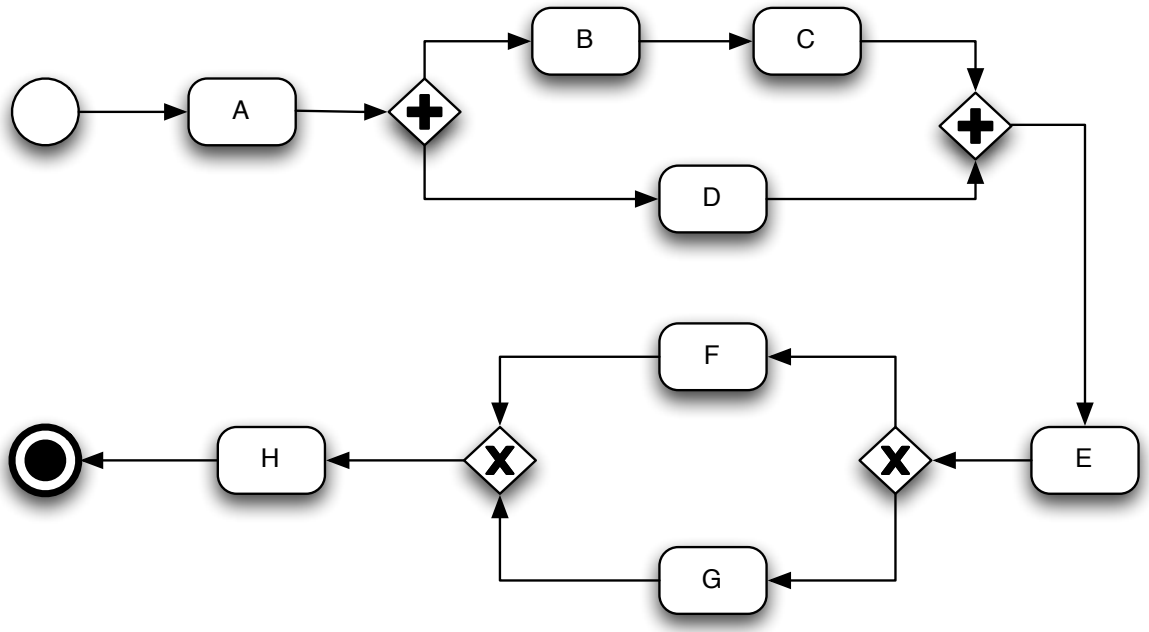


Figure: caption

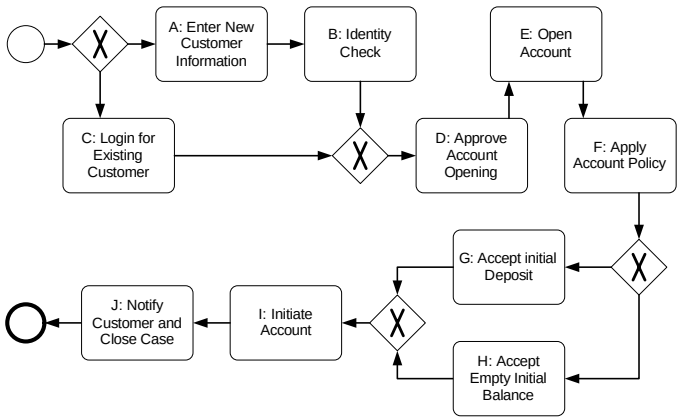
A, B, C, D, E, F, H
A, D, B, C, E, G, H
A, D, B, C, E, F, H

The Journey to Compliance

NICTA

1. Take or design a business process
2. Annotate the process
   - effects of the tasks (each task is annotated with the effects it produces)
   - rules encoding the norms relevant to the process

Adding Annotations



Task	Semantic Annotation
A	$newCustomer(x)$
B	$checkIdentity(x)$
C	$checkIdentity(x)$, $recordIdentity(x)$
E	$owner(x, y)$, $account(y)$
F	$accountType(y, type)$
G	$positiveBalance(y)$
H	$\neg positiveBalance(y)$
I	$accountActive(y)$
J	$notify(x, y)$

Rules for the Process

NICTA

- All new customers must be scanned against provided databases for identity checks.

$$r_1 : newCustomer(x) \Rightarrow O checkIdentity(x)$$

- Retain history of identity checks performed.

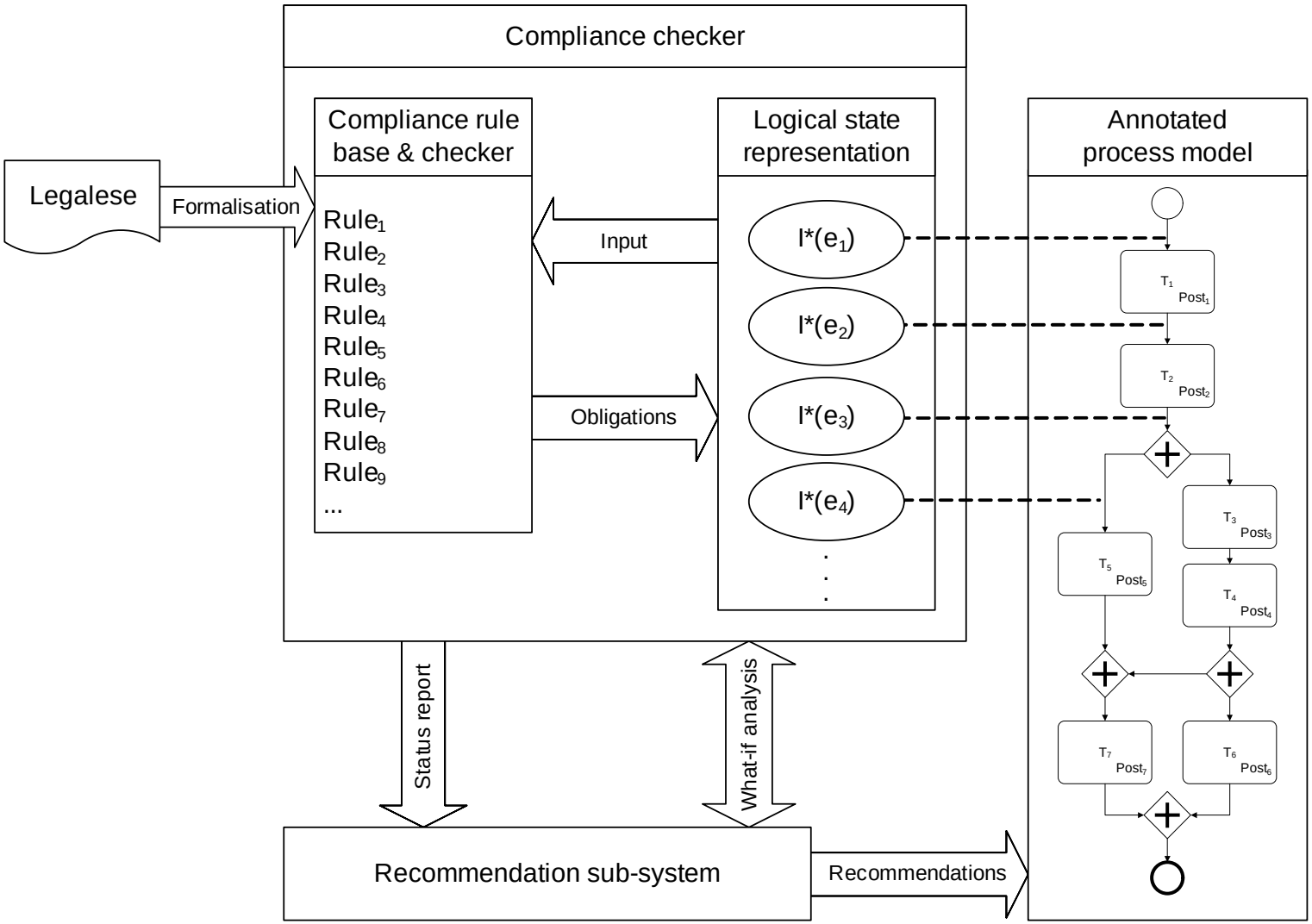
$$r_2 : checkIdentity(x) \Rightarrow O recordIdentity(x)$$

- Accounts must maintain a positive balance, unless approved by a bank manager, or for VIP customers.

$$r_3 : account(x) \Rightarrow O positiveBalance(x) \otimes O approveManager(x)$$

$$r_4 : account(x), accountType(x, VIP) \Rightarrow P\neg positiveBalance(x)$$

Business Process Compliance Architecture



Checking Compliance Recipe

NICTA

1. Determine the effects of each task and propagate them to successive tasks

Checking Compliance Recipe

NICTA

1. Determine the effects of each task and propagate them to successive tasks
2. Use the effects to trigger obligations. Run FCL with the effects of a task as input.

Checking Compliance Recipe

NICTA

1. Determine the effects of each task and propagate them to successive tasks
2. Use the effects to trigger obligations. Run FCL with the effects of a task as input.
3. Check which obligations have been fulfilled, violated

Checking Compliance Recipe

NICTA

1. Determine the effects of each task and propagate them to successive tasks
2. Use the effects to trigger obligations. Run FCL with the effects of a task as input.
3. Check which obligations have been fulfilled, violated
4. Shake well and serve!

Ideal Semantics: compliance checking

- FCL constraints determine behavioural paths (generic)
  - behavioural paths special case business processes
  - currently expressed as event sequences
- Ideal situation
  - Execution traces do not violate NFCL
- Sub-ideal situation
  - There are violations, but they are repaired/compensated
- Non-ideal (non-compliant) situation
  - There are violations, but they are NOT repaired/compensated
- Irrelevant situation
  - No rule is applicable

Handling Obligations

NICTA

An obligation chain $OA_1 \otimes \cdots \otimes OA_n$ is **active** given a set of literals S, if

- there is a rule $\Gamma \Rightarrow OA_1 \otimes \cdots \otimes OA_n$ such that $\Gamma \subseteq S$, i.e., the rule is triggered by the situation, and
- for all rule for conflicting chains, either
  - the chain is not triggered by the situation or
  - the negation of an element before the conflicting element is not in the situation.

The Algorithm

NICTA

Input: *Current* set of active obligation chains

$A_1 \otimes A_2 = C \in \mathit{Current}$

For each $C \in \mathit{Current}$

if $A_1 = OB$, then

  if $B \in S$, then

    remove($[T, R, A_1 \otimes A_2]$, *Current*),

    remove($[T, R, A_1 \otimes A_2]$, *Unfulfilled*)

    if $[T, R, B_1 \otimes B_2 \otimes A_1 \otimes A_2] \in \mathit{Violated}$ then

      add($[T, R, B_1 \otimes B_2 \otimes A_1 \otimes A_2]$, *Compensated*)

  if $\neg B \in S$, then

    add($[T, R, A_1 \otimes A_2]$, *Violated*), add($[T, R, A_2]$, *Current*)

  else

    add($[T, R, A_1 \otimes A_2]$, *Unfulfilled*).

Finally Compliant!

NICTA

Definition

- An execution trace is **compliant** iff for all $[T, R, A] \in \mathit{Current}$, $A = \mathit{OB} \otimes C$, for every $[T, R, A, B] \in \mathit{Violated}$, $[T, R, A, B] \in \mathit{Compensated}$ and $\mathit{Unfulfilled} = \emptyset$.
- An execution trace is **fully compliant** iff for all $[T, R, A] \in \mathit{Current}$, $A = \mathit{OB} \otimes C$, $\mathit{Violated} = \emptyset$ and $\mathit{Unfulfilled} = \emptyset$.
- A process is (**fully**) **compliant** iff all its execution traces are (fully) compliant.

Ontology of Norms

NICTA

- Persistent vs immediate obligations
  - An **immediate** obligation must be satisfied as soon as it occurs.
    'When banks proceed with any wire transfer, they must transmit a message, via SWIFT, to the receiving bank requesting that the payment is made according to the instructions given.'
  - A **persistent** obligation is activated and it remain in force in the future after it has been activated.
    'A service provider must not disclose personal information without the written consent of the customer'

Immediate obligations can be used to check the 'structural compliance of a BP'

Achievement vs maintenance

NICTA

- For an **achievement obligation**, a certain condition must occur at least once before the deadline
  'Customers must pay before the delivery of the good, after receiving the invoice'
- For **maintenance obligations**, a certain condition must obtain during all instants before the deadline:
  After opening a bank account, customers must keep a positive balance until bank charges are taken out.

Preemptive or Non-preemptive Obligations

- Only for achievement obligations
- **preemptive obligations**: the fulfillment of an obligation can happen before the obligation has been triggered.
- **non preemptive obligations**: the fulfillment of an obligation can happen only after the obligation has been triggered. 'Executors and administrators of a decedent's estate will be required to give notice to each beneficiary named in the Will within 60 days after the date X of an order admitting a will to probate has been signed.'

Conclusions

- Artificial Intelligence and Law and Deontic Logic have studied for over 30 years model for the representation of law and normative reasoning.

Conclusions

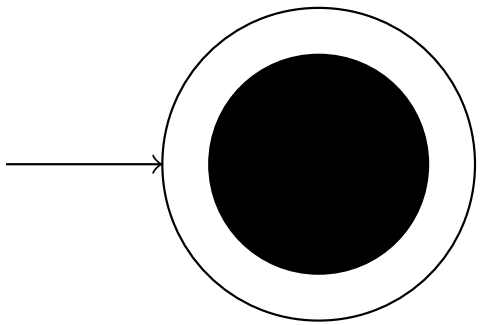
- Artificial Intelligence and Law and Deontic Logic have studied for over 30 years model for the representation of law and normative reasoning.
- Study the literature and don't invent yet another logic for reasoning about norms.

Conclusions

- Artificial Intelligence and Law and Deontic Logic have studied for over 30 years model for the representation of law and normative reasoning.
- Study the literature and don't invent yet another logic for reasoning about norms.
- If you invent a new logic, then you have to justify it: what does it do that other logics cannot do, or what does it do better.

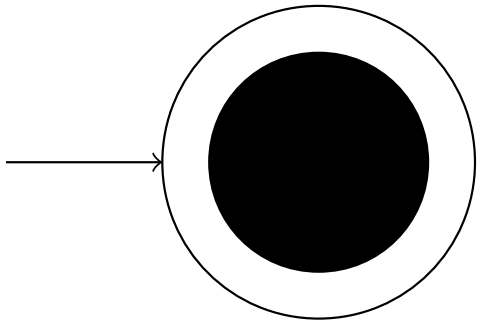
We Are Here Now

NICTA



We Are Here Now

NICTA



Dr Guido Governatori
Phone: +61 7 3300 8523
Email: guido.governatori@nicta.com.au
http://www.governatori.net

References

Thomas F. Gordon, Guido Governatori, and Antonino Rotolo.

Rules and norms: Requirements for rule interchange languages in the legal domain.
In Guido Governatori, John Hall, and Adrian Paschke, editors, *Rule Representation, Interchange and Reasoning on the Web*, LNCS 5858, pages 282–296. Springer, 2009.

Guido Governatori.

Representing business contracts in RuleML.
International Journal of Cooperative Information Systems, 14(2-3):181–216, 2005.

Guido Governatori, Zoran Milosevic, and Shazia Sadiq.

Compliance checking between business processes and business contracts.
In *EDOC 2006*, pp. 221–232. IEEE, 2006.

Guido Governatori and Antonino Rotolo.

Logic of Violations: A Gentzen System for Reasoning with Contrary-To-Duty Obligations.
Australasian Journal of Logic 4: 193-215, 2006

Guido Governatori and Antonino Rotolo.

An algorithm for business process compliance.
In Enrico Francesconi, Giovani Sartor, and Daniela Tiscornia, editors, *Jurix 2008*, pp. 186–191. IOS Press, 2008.

Guido Governatori and Shazia Sadiq.

The journey to business process compliance.
In Jorge Cardoso and Wil van der Aalst, editors, *Handbook of Research on BPM*, chapter 20, page 429–457, IGI Global, 2009.

Shazia Sadiq and Guido Governatori.

A methodological framework for aligning business processes and regulatory compliance.
In Jan van Brocke and Michael Rosemann, editors, *Handbook of Business Process Management*, Springer, 2010.

Shazia Sadiq, Guido Governatori, and Kioumars Naimiri.

Modelling of control objectives for business process compliance.
In *BPM 2007*, LNCS 4714, 2007.