InstAL 000000 Deontic Sensors

Example 00000 ODRL 00000 Epilogue 000

Representing and reasoning about policy for agent-based simulation

Julian Padget

Department of Computer Science, University of Bath

April 2, 2019

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
Contents					

- InstAL: a DSL for norm modelling
- Oeontic Sensors: normative reasoning as a service
- 4 Sample water management policy
- **5** Semantic policy representation

6 Epilogue

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
0000					
Contents					

- 2 InstAL: a DSL for norm modelling
- 3 Deontic Sensors: normative reasoning as a service
- 4 Sample water management policy
- 5 Semantic policy representation

6 Epilogue

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
0000					
Unpacking	g the title				

- normative models for intelligent agents
- agent architectures for normative reasoning
- applications in social simulation, security, games, legal reasoning, software engineering, data analytics
- norms \equiv policies \equiv regulations \equiv narratives \equiv requirements
- current work on:
 - normative reasoning as a service (Padget et al. 2018)
 - semantic representation of policy (on-going)
 - socio-cognitive technical systems (SCTS) (Noriega et al. 2017)
 - policy-making as an instance of SCTS (on-going)
 - use in social simulation (why I'm here)
- context from previous work
 - Institutional Action Language: InstAL (Padget et al. 2016b)
 - NJason: extension of Jason agent platform (Lee et al. 2014)

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
0000					
Unpacking	g the title				

- normative models for intelligent agents
- agent architectures for normative reasoning
- applications in social simulation, security, games, legal reasoning, software engineering, data analytics
- norms \equiv policies \equiv regulations \equiv narratives \equiv requirements
- current work on:
 - normative reasoning as a service (Padget et al. 2018)
 - semantic representation of policy (on-going)
 - socio-cognitive technical systems (SCTS) (Noriega et al. 2017)
 - policy-making as an instance of SCTS (on-going)
 - use in social simulation (why I'm here)
- context from previous work
 - Institutional Action Language: InstAL (Padget et al. 2016b)
 - NJason: extension of Jason agent platform (Lee et al. 2014)

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
0000					
Unpacking	g the title				

- normative models for intelligent agents
- agent architectures for normative reasoning
- applications in social simulation, security, games, legal reasoning, software engineering, data analytics
- norms \equiv policies \equiv regulations \equiv narratives \equiv requirements
- current work on:
 - normative reasoning as a service (Padget et al. 2018)
 - semantic representation of policy (on-going)
 - socio-cognitive technical systems (SCTS) (Noriega et al. 2017)
 - policy-making as an instance of SCTS (on-going)
 - use in social simulation (why I'm here)
- context from previous work
 - Institutional Action Language: InstAL (Padget et al. 2016b)
 - NJason: extension of Jason agent platform (Lee et al. 2014)

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
0000					
Unpacking	g the title				

- normative models for intelligent agents
- agent architectures for normative reasoning
- applications in social simulation, security, games, legal reasoning, software engineering, data analytics
- norms \equiv policies \equiv regulations \equiv narratives \equiv requirements
- o current work on:
 - normative reasoning as a service (Padget et al. 2018)
 - semantic representation of policy (on-going)
 - socio-cognitive technical systems (SCTS) (Noriega et al. 2017)
 - policy-making as an instance of SCTS (on-going)
 - use in social simulation (why I'm here)
- context from previous work
 - Institutional Action Language: InstAL (Padget et al. 2016b)
 - NJason: extension of Jason agent platform (Lee et al. 2014)

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
0000					
Unpacking	g the title				

- normative models for intelligent agents
- agent architectures for normative reasoning
- applications in social simulation, security, games, legal reasoning, software engineering, data analytics
- norms \equiv policies \equiv regulations \equiv narratives \equiv requirements
- o current work on:
 - normative reasoning as a service (Padget et al. 2018)
 - semantic representation of policy (on-going)
 - socio-cognitive technical systems (SCTS) (Noriega et al. 2017)
 - policy-making as an instance of SCTS (on-going)
 - use in social simulation (why I'm here)
- context from previous work
 - Institutional Action Language: InstAL (Padget et al. 2016b)
 - NJason: extension of Jason agent platform (Lee et al. 2014)

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
0000					
Unpacking	g the title				

- normative models for intelligent agents
- agent architectures for normative reasoning
- applications in social simulation, security, games, legal reasoning, software engineering, data analytics
- norms \equiv policies \equiv regulations \equiv narratives \equiv requirements
- o current work on:
 - normative reasoning as a service (Padget et al. 2018)
 - semantic representation of policy (on-going)
 - socio-cognitive technical systems (SCTS) (Noriega et al. 2017)
 - policy-making as an instance of SCTS (on-going)
 - use in social simulation (why I'm here)
- context from previous work
 - Institutional Action Language: InstAL (Padget et al. 2016b)
 - NJason: extension of Jason agent platform (Lee et al. 2014)

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
		_			
Why mod	lel policies	<u> </u>			

• evidence-based policy-making

- safer Al
- explainable Al
- human accountability + responsibility in S(C)TS
- confidence in outcomes
- nothing new here? depends on
 - the policy-modelling language
 - who/what does the reasoning

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
		_			
Why mod	lel policies	<u> </u>			

• evidence-based policy-making

- safer Al
- explainable Al
- human accountability + responsibility in S(C)TS
- confidence in outcomes
- nothing new here? depends on
 - the policy-modelling language
 - who/what does the reasoning

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
10/1		•			
VVhy mod	lel policies	<u>{</u>			

- evidence-based policy-making
- safer Al
- explainable Al
- human accountability + responsibility in S(C)TS
- confidence in outcomes
- nothing new here? depends on
 - the policy-modelling language
 - who/what does the reasoning

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
		_			
Why mod	lel policies	<u> </u>			

- evidence-based policy-making
- safer Al
- explainable Al
- human accountability + responsibility in S(C)TS
- confidence in outcomes
- nothing new here? depends on
 - the policy-modelling language
 - who/what does the reasoning

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
	lal nalisias	2			
	lel policies				

- evidence-based policy-making
- safer Al
- explainable Al
- human accountability + responsibility in S(C)TS
- confidence in outcomes
- nothing new here? depends on
 - the policy-modelling language
 - who/what does the reasoning

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
	lal nalisias	2			
	lel policies				

- evidence-based policy-making
- safer Al
- explainable Al
- human accountability + responsibility in S(C)TS
- confidence in outcomes
- nothing new here? depends on
 - the policy-modelling language
 - who/what does the reasoning

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
00000						
Why agent-based models?						

• abstraction (equational) vs. synthesis (agent-based)

- ...or top-down vs. bottom-up
- approaches emphasize different dimensions

- does a more complex model help understanding of complex systems?
- nothing new here? depends on
 - 💿 💿 ratio of reactive : deliberative : generative behaviour
 - 🕠 adaptation: hard-coded vs. data-driven behaviour
- policy is late-binding for simulation
- simulation is test environment for policy

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
Why ager	nt-based m	odels?			

- abstraction (equational) vs. synthesis (agent-based)
- ...or top-down vs. bottom-up
- approaches emphasize different dimensions

- does a more complex model help understanding of complex systems?
- nothing new here? depends on
 - 💿 💀 ratio of reactive : deliberative : generative behaviour
 - 🗉 adaptation: hard-coded vs. data-driven behaviour
- policy is late-binding for simulation
- simulation is test environment for policy

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
Why ager	nt-based m	odels?			

- abstraction (equational) vs. synthesis (agent-based)
- ...or top-down vs. bottom-up
- approaches emphasize different dimensions

- does a more complex model help understanding of complex systems?
- nothing new here? depends on
 - 🐳 ratio of reactive : deliberative : generative behaviour
 - 🗉 adaptation: hard-coded vs. data-driven behaviour
- policy is late-binding for simulation
- simulation is test environment for policy

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
Why ager	nt-based m	odels?			

- abstraction (equational) vs. synthesis (agent-based)
- ...or top-down vs. bottom-up
- approaches emphasize different dimensions

- does a more complex model help understanding of complex systems?
- nothing new here? depends on

ratio of reactive deliberative generative behaviour

- policy is late-binding for simulation
- simulation is test environment for policy

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
Why ager	nt-based m	odels?			

- abstraction (equational) vs. synthesis (agent-based)
- ...or top-down vs. bottom-up
- approaches emphasize different dimensions

- does a more complex model help understanding of complex systems?
- nothing new here? depends on
 - ratio of reactive : deliberative : generative behaviour
 - adaptation: hard-coded vs. data-driven behaviour
- policy is late-binding for simulation
- simulation is test environment for policy

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
Why ager	nt-based m	odels?			

- abstraction (equational) vs. synthesis (agent-based)
- ...or top-down vs. bottom-up
- approaches emphasize different dimensions

- does a more complex model help understanding of complex systems?
- nothing new here? depends on
 - ratio of reactive : deliberative : generative behaviour
 - adaptation: hard-coded vs. data-driven behaviour
- policy is late-binding for simulation
- simulation is test environment for policy

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
Why ager	nt-based m	odels?			

- abstraction (equational) vs. synthesis (agent-based)
- ...or top-down vs. bottom-up
- approaches emphasize different dimensions

- does a more complex model help understanding of complex systems?
- nothing new here? depends on
 - ratio of reactive : deliberative : generative behaviour
 - adaptation: hard-coded vs. data-driven behaviour
- policy is late-binding for simulation
- simulation is test environment for policy

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
Why ager	nt-based m	odels?			

- abstraction (equational) vs. synthesis (agent-based)
- ...or top-down vs. bottom-up
- approaches emphasize different dimensions

accuracy	granularity	fidelity
heterogeneity	precision	scalability

- does a more complex model help understanding of complex systems?
- nothing new here? depends on
 - ratio of reactive : deliberative : generative behaviour
 - adaptation: hard-coded vs. data-driven behaviour
- policy is late-binding for simulation
- simulation is test environment for policy

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
Why ager	nt-based m	odels?			

- abstraction (equational) vs. synthesis (agent-based)
- ...or top-down vs. bottom-up
- approaches emphasize different dimensions

accuracy	granularity	fidelity
heterogeneity	precision	scalability

- does a more complex model help understanding of complex systems?
- nothing new here? depends on
 - ratio of reactive : deliberative : generative behaviour
 - adaptation: hard-coded vs. data-driven behaviour
- policy is late-binding for simulation
- simulation is test environment for policy

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
Tesleveles	••				
Technolog	gy jigsaw				

- Agent-based simulation constraints:
 - sample size = memory
 - serialization = time
 - parameter range + dimensions = time to sweep
 - simple (individual) models = fidelity?
- Map to HPC? Overheads of many small tasks
- HPC opportunity: fidelity++ \Rightarrow better fit + all above



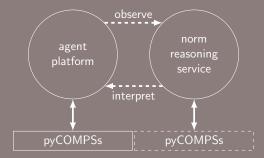
Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
Technolog	ny jigsaw				
	sy ngsaw				

- Agent-based simulation constraints:
 - sample size = memory
 - serialization = time
 - parameter range + dimensions = time to sweep
 - simple (individual) models = fidelity?
- Map to HPC? Overheads of many small tasks
- HPC opportunity: fidelity++ \Rightarrow better fit + all above



Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					
Tesleveles	••				
Technolog	gy jigsaw				

- Agent-based simulation constraints:
 - sample size = memory
 - ${\sf serialization} \hspace{.1in} = \hspace{.1in} {\sf time}$
 - parameter range + dimensions = time to sweep
 - simple (individual) models = fidelity?
- Map to HPC? Overheads of many small tasks
- HPC opportunity: fidelity++ \Rightarrow better fit + all above



Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
	00000					
Contents						

InstAL: a DSL for norm modelling

- 3 Deontic Sensors: normative reasoning as a service
- 4 Sample water management policy
- 5 Semantic policy representation

6 Epilogue

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
	00000				
Context					
CONTEXT					

- what ought (not) to be true
- what permissions (P) or prohibitions (F) hold
- what obligations (O) hold
- deontic logic (Wright 1951) of F, P, O
- ~~ knowledge representation as norms.
- ullet \rightsquigarrow governance of agents in multiagent systems
- \circ \rightsquigarrow governance of actors in socio-cognitive technical systems.

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
	00000				
Context					
CONTEXT					

- what ought (not) to be true
- what permissions (P) or prohibitions (F) hold
- what obligations (O) hold
- deontic logic (Wright 1951) of F, P, O
- $\circ \rightsquigarrow$ knowledge representation as norms
- solution with a second state of agents in multiagent systems
- ullet ightarrow governance of actors in socio-cognitive technical systems

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
	00000				
Context					
Context					

- what ought (not) to be true
- what permissions (P) or prohibitions (F) hold
- what obligations (O) hold
- deontic logic (Wright 1951) of F, P, O
- $\bullet \rightsquigarrow$ knowledge representation as norms
- $\bullet \rightsquigarrow$ governance of agents in multiagent systems
- \circ \rightsquigarrow governance of actors in socio-cognitive technical systems.

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
	00000				
Context					
Context					

- what ought (not) to be true
- what permissions (P) or prohibitions (F) hold
- what obligations (O) hold
- deontic logic (Wright 1951) of F, P, O
- $\bullet \rightsquigarrow$ knowledge representation as norms
- $\bullet \rightsquigarrow$ governance of agents in multiagent systems
- ullet \rightsquigarrow governance of actors in socio-cognitive technical systems

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	○○●○○○	00000000000	00000	00000	000
Conceptua	l overview				

- norm = constraint on action in a context
- norm = part of policy or regulation or requirement.
- institution = set of norms
- institution = policy or regulations or requirements
- associates action with (institutional) consequences
- constitutive norms (Searle 1995):
- counts-as (Jones et al. 1996)

real-world event ~> institutional event



Introduction 00000	InstAL 000000	Deontic Sensors	Example 00000	ODRL 00000	Epilogue 000		
Conceptu	al overview	1					
• ir	nspiration: 〈	economics (N social science social policy	es (Harré et				
 norm = constraint on action in a context 							

Introduction 00000	InstAL ○0●000	Deontic Sensors	Example 00000	ODRL 00000	Epilogue 000	
Conceptua	al overview					
• in	spiration: 〈	economics (N social science social policy	es (Harré et			
 norm = constraint on action in a context 						
 norm = part of policy or regulation or requirement 						

Introduction 00000	InstAL ○0●000	Deontic Sensors	Example 00000	0DRL 00000	Epilogue 000	
Conceptu	al overview					
o ir	nspiration: <	economics (N social science				
• "		social policy				
		constraint on act				
 norm = part of policy or regulation or requirement institution = set of norms 						

Introduction 00000	InstAL 00000	Deontic Sensors 00000000000	Example 00000	ODRL 00000	Epilogue 000
Conceptu	al overview				
• ir	nspiration: {	economics (f social science social policy	es (Harré et	: al. 1972)	
	 norm = p institution institution 	onstraint on act part of policy or n = set of norms n = policy or reg	regulation or gulations or 1	requiremen requirements	

- constitutive norms (Searle 1995): brute facts →→ social facts
- counts-as (Jones et al. 1996)

real-world event ~> institutional event



Introduction 00000	InstAL ○○●○○○	Deontic Sensors 0000000000	Example 00000	ODRL 00000	Epilogue 000
Conceptual	overview				
	ſ	economics (No			
• ins	piration: {	social sciences social policy ((

- ${\scriptstyle \bullet}$ norm = constraint on action in a context
- norm = part of policy or regulation or requirement
- institution = set of norms
- institution = policy or regulations or requirements
- associates action with (institutional) consequences
- constitutive norms (Searle 1995): brute facts → social facts
- counts-as (Jones et al. 1996)

real-world event ~> institutional event



Introduction 00000	InstAL 00000	Deontic Sensors 00000000000	Example 00000	ODRL 00000	Epilogue 000
Conceptua	l overview				
• in:	spiration: $\left\{ \right.$	economics (N social science social policy	és (Harré et	al. 1972)	

- ${\scriptstyle \bullet}$ norm = constraint on action in a context
- norm = part of policy or regulation or requirement
- institution = set of norms
- institution = policy or regulations or requirements
- associates action with (institutional) consequences
- constitutive norms (Searle 1995): brute facts \rightsquigarrow social facts
- counts-as (Jones et al. 1996).

real-world event ~> institutional event



Introduction 00000	InstAL 00000	Deontic Sensors	Example 00000	ODRL 00000	Epilogue 000
Conceptua	al overview				
• in	spiration: {	economics (N social science social policy	és (Harré et	al. 1972)	
		onstraint on act			
		art of policy or I		requirement	ţ
		$\mathfrak{n} = set of norms$			
		 policy or reg action with (inst 			
0 00		orms (Searle 19			

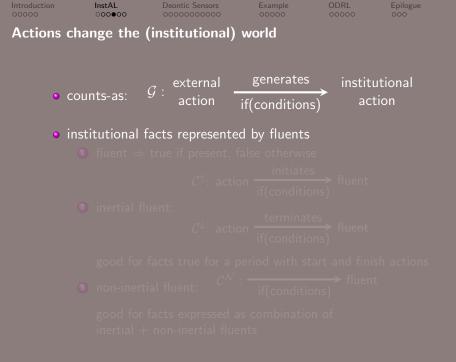
brute facts \rightsquigarrow social facts

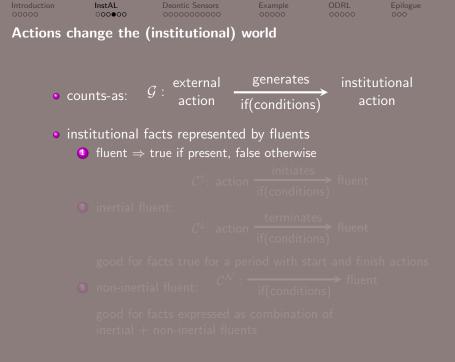
• counts-as (Jones et al. 1996):

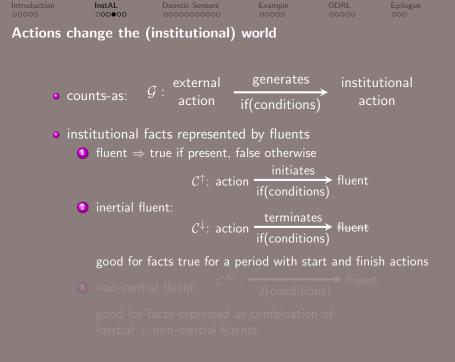
real-world event \rightsquigarrow institutional event

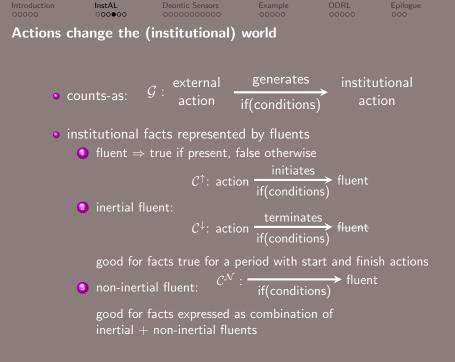












Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	○○○○●○		00000	00000	000
Making it	work				

• mathematical model: sets + relations $(\mathcal{G}, \mathcal{C}) \rightsquigarrow$ labelled transition system

$$\Delta \stackrel{e_1}{\longrightarrow} S_1 \stackrel{e_2}{\longrightarrow} S_2 \stackrel{e_3}{\longrightarrow} \dots$$

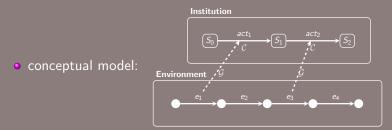


generalizes to multiple, connected institutions

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	○000●0		00000	00000	000
Making it	: work				

• mathematical model: sets + relations (\mathcal{G}, \mathcal{C}) \rightsquigarrow labelled transition system

$$\Delta \stackrel{e_1}{\longrightarrow} S_1 \stackrel{e_2}{\longrightarrow} S_2 \stackrel{e_3}{\longrightarrow} \dots$$

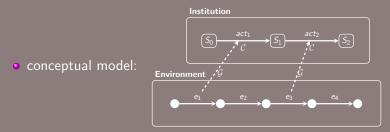


generalizes to multiple, connected institutions

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	○000●0		00000	00000	000
Making it	: work				

mathematical model: sets + relations (G,C) → labelled transition system

$$\Delta \stackrel{e_1}{\longrightarrow} S_1 \stackrel{e_2}{\longrightarrow} S_2 \stackrel{e_3}{\longrightarrow} \dots$$



• generalizes to multiple, connected institutions

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	○0000●		00000	00000	000
Implemen	tation				

- o python front-end
- compiler: InstAL to Answer Set Programming
- python API to clingo (answer set solver, C++)
- answer sets delivered in JSON
- visualization tools generate images from traces

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	○○○○○●		00000	00000	000
Implemen	itation				

- python front-end
- compiler: InstAL to Answer Set Programming
- python API to clingo (answer set solver, C++)
- answer sets delivered in JSON
- visualization tools generate images from traces

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	○○○○○●		00000	00000	000
Implemen	itation				

- python front-end
- compiler: InstAL to Answer Set Programming
- python API to clingo (answer set solver, C++)
- answer sets delivered in JSON
- visualization tools generate images from traces

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	○○○○○●		00000	00000	000
Implemen	itation				

- python front-end
- compiler: InstAL to Answer Set Programming
- python API to clingo (answer set solver, C++)
- answer sets delivered in JSON
- visualization tools generate images from traces

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	○○○○○●		00000	00000	000
Implemen	itation				

- python front-end
- compiler: InstAL to Answer Set Programming
- python API to clingo (answer set solver, C++)
- answer sets delivered in JSON
- visualization tools generate images from traces

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	○0000●		00000	00000	000
Implemen	itation				

- python front-end
- compiler: InstAL to Answer Set Programming
- python API to clingo (answer set solver, C++)
- answer sets delivered in JSON
- visualization tools generate images from traces

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			
Contents					

1 Introduction

2 InstAL: a DSL for norm modelling

3 Deontic Sensors: normative reasoning as a service

4 Sample water management policy

5 Semantic policy representation

6 Epilogue

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	000000		00000	00000	000
Al softwa	re siloes				

- agent platform package =

 agent behaviour + environment + [institution(s)]
 see Aldewereld et al. (2016) for survey
- institutions absent or optional extra
- environment interface standard Behrens et al. (2011)
- buy the package: can't build platform from components

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	000000		00000	00000	000
AI softwa	re silnes				

- agent platform package =

 agent behaviour + environment + [institution(s)]
 see Aldewereld et al. (2016) for survey
- institutions absent or optional extra
- environment interface standard Behrens et al. (2011)
- buy the package: can't build platform from components

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	000000		00000	00000	000
AI softwa	re siloes				

- agent platform package =

 agent behaviour + environment + [institution(s)]
 see Aldewereld et al. (2016) for survey
- institutions absent or optional extra
- environment interface standard Behrens et al. (2011)
- buy the package: can't build platform from components

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	000000		00000	00000	000
AI softwa	re siloes				

- agent platform package =

 agent behaviour + environment + [institution(s)]
 see Aldewereld et al. (2016) for survey
- institutions absent or optional extra
- environment interface standard Behrens et al. (2011)
- buy the package: can't build platform from components

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		00000000000			
Software e	engineering	g approach			

• refactoring: make norm reasoning a separate component

- design pattern: blueprint for deontic sensors
- resource-oriented architecture: deploy as RESTful services
- ~> decoupling
- ~> institution re-use
- ~ institution certification
- access normative reasoning across + outside AI

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
		0000000000				
Software	engineering	g approach				

- refactoring: make norm reasoning a separate component
- design pattern: blueprint for deontic sensors
- resource-oriented architecture: deploy as RESTful services
- ~> decoupling
- ~> institution re-use
- ~ institution certification
- access normative reasoning across + outside Al

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			
Software e	engineering	g approach			

- refactoring: make norm reasoning a separate component
- design pattern: blueprint for deontic sensors
- resource-oriented architecture: deploy as RESTful services
- ~> decoupling
- \rightsquigarrow institution re-use
- \rightsquigarrow institution certification
- access normative reasoning across + outside Al

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		00000000000			
•					
Software	engineering	g approach			

- refactoring: make norm reasoning a separate component
- design pattern: blueprint for deontic sensors
- resource-oriented architecture: deploy as RESTful services
- ~ decoupling
- ~> institution re-use
- ~ institution certification
- access normative reasoning across + outside Al

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		00000000000			
•					
Software	engineering	g approach			

- refactoring: make norm reasoning a separate component
- design pattern: blueprint for deontic sensors
- resource-oriented architecture: deploy as RESTful services
- o ~> decoupling
- ~> institution re-use
- ~> institution certification
- access normative reasoning across + outside AI

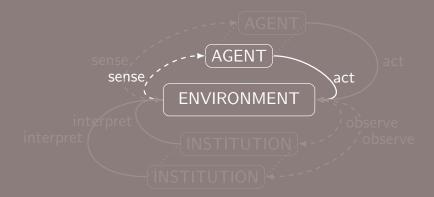
Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		00000000000			
•					
Software	engineering	g approach			

- refactoring: make norm reasoning a separate component
- design pattern: blueprint for deontic sensors
- resource-oriented architecture: deploy as RESTful services
- ~ decoupling
- \rightsquigarrow institution re-use
- $\circ \rightsquigarrow$ institution certification
- access normative reasoning across + outside AI

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue		
		00000000000					
Software engineering approach							

- refactoring: make norm reasoning a separate component
- design pattern: blueprint for deontic sensors
- resource-oriented architecture: deploy as RESTful services
- o ~> decoupling
- \rightsquigarrow institution re-use
- $\circ \rightsquigarrow$ institution certification
- access normative reasoning across + outside AI

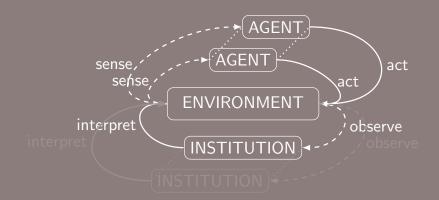




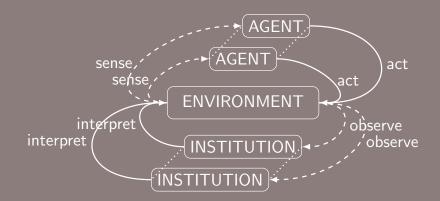






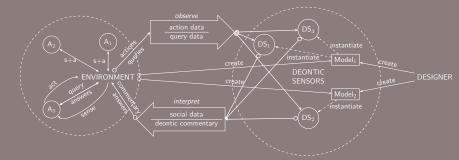






Introduction 00000	InstAL 000000	Deontic Sensors	Example 00000	ODRL 00000	Epilogue 000	
_						

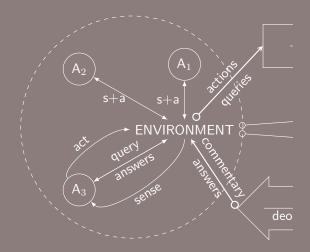
Deontic Sensors Architecture



resource-oriented architecture (ROA) pattern for normative reasoning services (Padget et al. 2018)

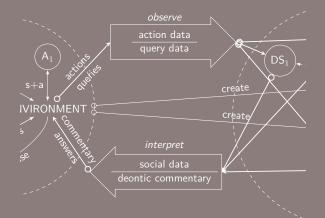
Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			

Architecture: environment and agents



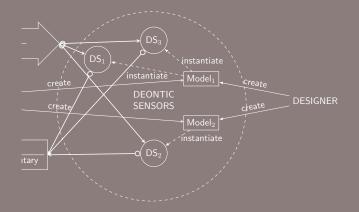
Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			

Architecture: observe-interpret



Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		000000000000			

Architecture: deontic sensors



Introduction 00000	InstAL 000000	Deontic Sensors	Example 00000	ODRL 00000	Epilogue 000	
ROA end	ooints					
0 -	→POST /n	nodel/				
			Create m	odel from	specification	
	-/model/	'X				

- \rightarrow POST /model/X/instance/Y/query/ Create a query of instance Y with POST data \leftarrow /model/X/instantiate/Y/query/Z
- $\mathbb{O} \to \operatorname{GET} / \operatorname{model} / X / \operatorname{instance} / Y / \operatorname{query} / Z / \operatorname{output} Read result of query}$

Introduction 00000	InstAL 000000	Deontic Sensors	Example 00000	ODRL 00000	Epilogue 000	
ROA end	points					
0 -	→POST /n	nodel/	~			

Create model from specification

 $\leftarrow / \texttt{model} / X$

- ② → POST /model/X/instance/ Create instance of model X with POST data ← /model/X/instance/Y
- → POST /model/X/instance/Y/query/ Create a query of instance Y with POST data ← /model/X/instantiate/Y/query/Z
- \rightarrow GET /model/X/instance/Y/query/Z/output Read result of query

Introduction 00000	InstAL 000000	Deontic Sensors	Example 00000	0DRL 00000	Epilogue 000	
ROA endp	ooints					

 $\bigcirc \rightarrow \text{POST} / \text{model} /$

Create model from specification

 \leftarrow /model/X

② → POST /model/X/instance/ Create instance of model X with POST data ← /model/X/instance/Y

- → POST /model/X/instance/Y/query/ Create a query of instance Y with POST data ← /model/X/instantiate/Y/query/Z
- \rightarrow GET /model/X/instance/Y/query/Z/output Read result of query

Introduction 00000	InstAL 000000	Deontic Sensors	Example 00000	ODRL 00000	Epilogue 000	
ROA end	points					

 $\bigcirc \rightarrow \text{POST} / \text{model} /$

Create model from specification

- $\leftarrow / \texttt{model} / X$
- ② → POST /model/X/instance/ Create instance of model X with POST data ← /model/X/instance/Y
- → POST /model/X/instance/Y/query/ Create a query of instance Y with POST data ← /model/X/instantiate/Y/query/Z
- \rightarrow GET /model/X/instance/Y/query/Z/output Read result of guery

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			

- agent platform: Jason (Bordini et al. 2007)
 - Belief-Desire-Intention (BDI) agent architecture
 - means-end reasoning
 - open-minded commitment
- normative reasoning: InstAL (Padget et al. 2016a)
 - InstAL: Institutional Action Language
 - builds model in Answer Set Prolog
 - symbolic model checking
 - single event \rightarrow new model state: +/- FPO +/- domain facts
 - multiple events \rightarrow alternative model states
- InstAL as a service:
 - 🔹 python Flask for RESTful APL
 - celery + rabbitMQ for server task creation/communications
 - python InstAL compiler
 - clingo for grounding + solving.
 - python client: JSON payload both ways

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			

- agent platform: Jason (Bordini et al. 2007)
 - Belief-Desire-Intention (BDI) agent architecture
 - means-end reasoning
 - open-minded commitment
- normative reasoning: InstAL (Padget et al. 2016a)
 - InstAL: Institutional Action Language
 - builds model in Answer Set Prolog
 - symbolic model checking
 - single event \rightarrow new model state: +/- FPO +/- domain facts
 - multiple events \rightarrow alternative model states
- InstAL as a service:
 - python Flask for RESTful APL
 - celery + rabbitMQ for server task creation/communications
 - python InstAL compiler.
 - clingo for grounding + solving.
 - python client: JSON payload both ways

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			

- agent platform: Jason (Bordini <u>et al. 2007)</u>
 - Belief-Desire-Intention (BDI) agent architecture
 - means-end reasoning
 - open-minded commitment
- normative reasoning: InstAL (Padget et al. 2016a)
 - InstAL: Institutional Action Language
 - builds model in Answer Set Prolog
 - symbolic model checking
 - single event \rightarrow new model state: +/- FPO +/- domain facts
 - $\bullet \ \ \mathsf{multiple} \ \ \mathsf{events} \rightarrow \mathsf{alternative} \ \ \mathsf{model} \ \mathsf{states}$
- InstAL as a service:
 - python Flask for RESTful API
 - celery + rabbitMQ for server task creation/communications
 - python InstAL compiler
 - clingo for grounding + solving
 - python client: JSON payload both ways

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			

- agent platform: Jason (Bordini <u>et al. 2007)</u>
 - Belief-Desire-Intention (BDI) agent architecture
 - means-end reasoning
 - open-minded commitment
- normative reasoning: InstAL (Padget et al. 2016a)
 - InstAL: Institutional Action Language
 - builds model in Answer Set Prolog
 - symbolic model checking
 - single event \rightarrow new model state: +/- FPO +/- domain facts
 - $\bullet \ \ \text{multiple events} \rightarrow \text{alternative } \underline{\text{model states}}$
- InstAL as a service:
 - python Flask for RESTful API
 - celery + rabbitMQ for server task creation/communications
 - python InstAL compiler
 - clingo for grounding + solving
 - python client: JSON payload both ways

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			

- agent platform: Jason (Bordini <u>et al. 2007)</u>
 - Belief-Desire-Intention (BDI) agent architecture
 - means-end reasoning
 - open-minded commitment
- normative reasoning: InstAL (Padget et al. 2016a)
 - InstAL: Institutional Action Language
 - builds model in Answer Set Prolog
 - symbolic model checking
 - single event \rightarrow new model state: +/- FPO +/- domain facts
 - $\bullet \ \ \text{multiple events} \rightarrow \text{alternative } \underline{\text{model states}}$
- InstAL as a service:
 - python Flask for RESTful API
 - celery + rabbitMQ for server task creation/communications
 - python InstAL compiler
 - clingo for grounding + solving
 - python client: JSON payload both ways

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			

- agent platform: Jason (Bordini <u>et al. 2007)</u>
 - Belief-Desire-Intention (BDI) agent architecture
 - means-end reasoning
 - open-minded commitment
- normative reasoning: InstAL (Padget et al. 2016a)
 - InstAL: Institutional Action Language
 - builds model in Answer Set Prolog
 - symbolic model checking
 - single event \rightarrow new model state: +/- FPO +/- domain facts
 - $\bullet \ \ \text{multiple events} \rightarrow \text{alternative } \underline{\text{model states}}$
- InstAL as a service:
 - python Flask for RESTful API
 - celery + rabbitMQ for server task creation/communications
 - python InstAL compiler
 - clingo for grounding + solving
 - python client: JSON payload both ways

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			

- agent platform: Jason (Bordini <u>et al. 2007)</u>
 - Belief-Desire-Intention (BDI) agent architecture
 - means-end reasoning
 - open-minded commitment
- normative reasoning: InstAL (Padget et al. 2016a)
 - InstAL: Institutional Action Language
 - builds model in Answer Set Prolog
 - symbolic model checking
 - single event \rightarrow new model state: +/- FPO +/- domain facts
 - $\bullet \ \ \mathsf{multiple} \ \ \mathsf{events} \rightarrow \mathsf{alternative} \ \ \mathsf{model} \ \mathsf{states}$
- InstAL as a service:
 - python Flask for RESTful API
 - celery + rabbitMQ for server task creation/communications
 - python InstAL compiler
 - clingo for grounding + solving
 - python client: JSON payload both ways

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			

- agent platform: Jason (Bordini <u>et al. 2007)</u>
 - Belief-Desire-Intention (BDI) agent architecture
 - means-end reasoning
 - open-minded commitment
- normative reasoning: InstAL (Padget et al. 2016a)
 - InstAL: Institutional Action Language
 - builds model in Answer Set Prolog
 - symbolic model checking
 - single event \rightarrow new model state: +/- FPO +/- domain facts
 - \circ multiple events \rightarrow alternative model states
- InstAL as a service:
 - python Flask for RESTful API
 - celery + rabbitMQ for server task creation/communications
 - python InstAL compiler
 - clingo for grounding + solving
 - python client: JSON payload both ways

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		000000000			
• • • • •		~			

Instantiating for HPC

replace Celery with pyCOMPSs

2 replace Flask (RESTful) API with conventional API

extend agent platform Controller to

- send actions to InstAL.
- receive interpretations from InstAL
- publish interpretation for agents to perceive.
- extend agent reasoning to account for normative percepts

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		000000000			
		~			
Instantiat	ing for HP	C			

eplace Flask (RESTful) API with conventional API

extend agent platform Controller to

- send actions to InstAL
- \circ receive interpretations from InstALL
- ightarrow publish interpretation for agents to perceive.
- extend agent reasoning to account for normative percepts

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			
		-			
Instantiat	ing for HP	C			

replace Flask (RESTful) API with conventional API

3 extend agent platform Controller to

- send actions to InstAL
- receive interpretations from InstAL
- publish interpretation for agents to perceive

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			
		-			
Instantiat	ing for HP	C			

replace Flask (RESTful) API with conventional API

extend agent platform Controller to

- send actions to InstAL
- receive interpretations from InstAL
- publish interpretation for agents to perceive

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			
		-			
Instantiat	ing for HP	C			

replace Flask (RESTful) API with conventional API

extend agent platform Controller to

- send actions to InstAL
- receive interpretations from InstAL
- publish interpretation for agents to perceive

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			
		-			
Instantiat	ing for HP	C			

replace Flask (RESTful) API with conventional API

3 extend agent platform Controller to

- send actions to InstAL
- receive interpretations from InstAL
- publish interpretation for agents to perceive

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
		0000000000			
		-			
Instantiat	ing for HP	C			

replace Flask (RESTful) API with conventional API

extend agent platform Controller to

- send actions to InstAL
- receive interpretations from InstAL
- publish interpretation for agents to perceive

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
			00000			
Contents						

1 Introduction

- 2 InstAL: a DSL for norm modelling
- 3 Deontic Sensors: normative reasoning as a service
- 4 Sample water management policy
- 5 Semantic policy representation

6 Epilogue

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
			0000			
Instal de	clarations					

Different types of declarations:

1	type Industry;		
2	<pre>exogenous event discharge(WWTP,Mass);</pre>		
	violation event illegalDischarge(WWTP, Mas	ss)	
4	<pre>inst event iDischarge(WWTP,Mass);</pre>		
5	<pre>fluent highMercury(Mass);</pre>		
6	obligation fluent obl(
7	<pre>iInform(Industry,WWTP,Mass),</pre>		
8	<pre>iRelease(Industry,WWTP,Mass),</pre>		deadlin
9	<pre>failureToInform(Industry,WWTP,Mass));</pre>		

Introdu		InstAL 000000	Deontic Sensors	Example 00000	ODRL 00000	Epilogue 000
Inst	AL rule	es				
	Generat	tes, initiate	es and terminate	es rules:		
	discha	rge(WWTP,M	lass) generates	iDischarge	(WWTP,Mass)	
2	if	treated(WW	TP,Mass,Treatme			
			lass) generates		charge(WWTP	,Mass)
4			d(WWTP,Mass,Tre			
			lass) generates	illegalDisc	charge(WWTP	,Mass)
		highMercur			1.5	
7			(WWTP,Mass) ini	tiates ille	egalBecause	(untreated,
8		TP,Mass)	d (INTE Maga The	(atmost).		
0 9			d(WWTP,Mass,Tre (WWTP,Mass) ini			
			(WWIF,Mass) 111 WWTP,Mass)	.craces III	eyaibecause	
10		highMercur				
11			Mass) terminate	s treated(W	WTP.Mass.T	(reatment)
12			TP,Mass,Treatme			
13		rm(WWTP,Ma eatment)	ss,Treatment) i	nitiates th	reated(WWTP	
14	if	treating(W	WTP,Mass);			
15	initia	-				
16		rcury(m2),				
17		Contract(w				
18		nform(i1,w tp1,M))	wtp1,M),iReleas	se(i1,wwtp1,	.M),failure	ToInform(i1,

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
			00000			
~ •						
Sample ri	JN					

• grounding specification:

- 1 Industry: i1 i2
- 2 Mass: m1 m2
- 3 Reason: untreated high_mercury
- 4 Treatment: tk
- 5 WWTP: wwtp1 wwtp2

input trace:

- 1 observed(inform(i1,wwtp1,m2))
- 2 observed(release(i1,wwtp1,m2))
- 3 observed(receive(wwtp1,i1,m2))
- 4 observed(perform(wwtpl,m2,tk))
- 5 observed(discharge(wwtp1,m2))
- 6 observed(release(i2,wwtp2,m1))
- 7 observed(receive(wwtp2,i2,m1))
- 8 observed(discharge(wwtp2,m1))

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
			00000			
Sample ru	in					
Sample I						

• grounding specification:

- 1 Industry: i1 i2
- 2 Mass: m1 m2
- 3 Reason: untreated high_mercury
- 4 Treatment: tk
- 5 WWTP: wwtp1 wwtp2

• input trace:

- 1 observed(inform(i1,wwtp1,m2))
- 2 observed(release(i1,wwtp1,m2))
- 3 observed(receive(wwtp1,i1,m2))
- 4 observed(perform(wwtp1,m2,tk))
- 5 observed(discharge(wwtp1,m2))
- 6 observed(release(i2,wwtp2,m1))
- 7 observed(receive(wwtp2,i2,m1))
- 8 observed(discharge(wwtp2,m1))

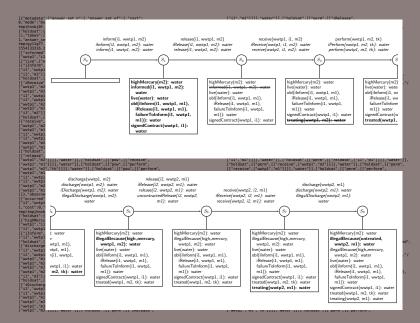
Introduction	InstAL 000000	Deontic Sensors	Example ○000●	ODRL 00000	Epilogue
Trace visi	ualization				
tmnihnvhi@f.ln	<pre>["answer_set_n":1, "answer_set_of" :i_shot", "pid":7887, "source_file: o"], "timestamp":1554133316.23172</pre>	19. "version": "1.0.1"}. "state":		<pre>'holdsat",[["perm",[["iReleas 'pows":[["holdsat",[["pow",[" harge",["wwtp1","m1"]]]],"wat</pre>	
{"holdsat":{"f [],"tpows":[]} 1."answer set	<pre>fluents":[],"gpows":[],"ipows":[] ,"observed":[],"occurred":[]}}, of":1."cost":0."mode":"multi she</pre>],"obls":[],"perms":[],"pows": ("metadata":("answer_set_n": ot","pid":7887,"source_files":["/tmp/ tamp":	<pre>[["discharge",["wwtpl","n2 ["i1","wwtpl","n1"]]]],"wa ["i1","wwtp1","n2"]]]],"wa</pre>	"]]]] "water"]],["holdsat",[[" ter"]],["holdsat",[["pow",[[" ter"]],["holdsat",[["pow",[[" ter"]],["holdsat",[["pow",[["	"pow",[["inform", inform", perform".
1554133316.231	17219, "version":"1.0.1"}, "state":	:{"holdsat":{"fluents":[["holdsat",			
["wwtp1","i1"] [["live",["wat	<pre>ii, water]], ["holdsat", [["highMe er"]], "water"]]], "gpows": [], "ip</pre>	<pre>"holdsat",[["signedContract", ercury",["m2"]],"water"]],["holdsat", ows":[],"obls":[["holdsat",[["obl",</pre>	["wwtp1","m2"]]]],"water"] ["i1","m1"]]]],"water"]],[],["holdsat",[["pow",[["reces],["holdsat",[["pow",[["release" "holdsat",[["pow",[["release"	ise",
[["iInform",[" ["i1","wwtp1", ["i1","m2"]]]]	<pre>'11", "wwtp1", "m1"]],["1Release", "m1"]]]], "water"]]], "perms":[["f , "water"]], ["holdsat", [["perm",</pre>	<pre>wsf:[],"obls':[["holdsst",[["obl", ["il","nl"]],["falureToInform", holdsst",[["perm",[["iRelease", ["iRelease",["il","nl"]]]],"wster"]], "]]]],"wster"]],["holdsst",[["perm",</pre>	["11", "m2"]]]], "water"]]], ["11", "m2"]]]]], "occurred" ["occurred", [["iRelease", []	<pre>holdsat',[["pow',[["release" tpows":[]),"observed":[["obs :[["occurred",[["release",["a i1","m2"]],"water"]],["occur "]]]}, ("metadata":{"answer</pre>	erved",[["release", .1","m2"]],"water"]], red".[["failureToInform".
[["wwtpl","m2"	"tk"]]]] "water"]] ["holdsat" []	losat",[["perm",[["Perform",	<pre>["i1", "wwtp1", "m2"]], "wate 1, "cost":0, "mode": "multi_si tmp/tmpihovbi0f_lp"l. "time</pre>	r"]]]}},{"metadata":{"answer_ hot","pid":7887,"source_files stamp":1554133316.2317219,"ve	<pre>set_n":1,"answer_set_of": ":["/tmp/tmprgyllg77.lp","/ rsion":1.8.l"}."state":</pre>
["wwtp1","m1", ["i1","wwtp1",	"tk"]]]], "water"]], ["holdsat",[] "m2"]]]], "water"]], ["holdsat",[] "m1"]]]], "water"]], ["holdsat",[]	["perm",[["iInform", ["perm",[["iInform",	{"holdsat":{"fluents":[["holdsat",[["informed",["	oldsat",[["treating",["wwtp1" i1","wwtp1","m2"]],"water"]], ',"i1"]],"water"]],["holdsat"	,"m2"]],"water"]], ["holdsat",
["wvtp1", "m2"]	<pre>[]]], "water"]], ["holdsat", [["perr ll]] "water"]], ["holdsat", [["perr</pre>	n",[["iDischarge", "[["releare"	["m2"]], "water"]], ["holdsa:	t",[["live",["water"]],"water	"]]], "gpows":[], "ipows": "m1"]] ["ipolesse"
["i1", "m2"]]]] ["holdsat", [["	<pre>, "water"]], ["holdsat", [["perm",] "perm", [["receive", ["wwtpl", "m2"] wwtpl", "m1"]]]]]</pre>	<pre>[["release",["il","nl"]]],"water"]],]]]],"water"]],["boldsat",[["perm", dsat",[["perm",[["perform", ""perm",[["inform", ""perm",[["inform","]]]</pre>	["il", "m1"]], ["failureToIn [["holdsat", [["perm", [["iR [["iRelease", ["il", "m1"]]]]	form",["il","wwtpl","ml"]]], lease",["il","m2"]]],"water ,"water"]],["holdsat",[["perm",[["iRed ,["holdsat",[["perm",[["iPerm","],["iPerm",""iPerm",["iPerm",["iPerm",["iPerm",""iPerm",["iPerm",""iPerm",["iPerm",""iPerm",["iPerm",""iPerm",""iPerm",""iPerm",""iPerm",""iPerm",""iPerm",""iPerm",""iPerm",""iPerm","["iPerm","";";";";";";";";";";";";";";";";";";	"water"]]],"perms": "]],["holdsat",[["perm", "".[["iBeceive".
["wwtp1","m2", ["wwtp1","m1",	"tk"]]]],"water"]],["holdsat",[] "tk"]]]],"water"]],["holdsat",[]	["perm",[["perform", ["perm",[["inform",	["wwtp1","m2"]]]],"water"] ["wwtp1","m1"]]],"water"]	<pre>[,["holdsat",[["perm",[["iRec],["holdsat",[["perm",[["iPerm"],["iPerm"</pre>	eive", form", "Deferm"
["i1", "wrtp1", ["wrtp1", "m2"]	"m2"]]]], "water"]], ["holdsat", [] "m1"]]]], "water"]], ["holdsat", []]]], "water"]], ["holdsat", [["perm	['perm',[["discharge", m",[["discharge",	["wvtp1","m1","tk"]]]],"wa ["i1","wvtp1","m2"]]],"wa	ter"]],["holdsat",[["perm",[] ter"]],["holdsat",[["perm",[] ter"]],["holdsat",[["perm",[]	"iInform", "iInform",
["wwtp1","m1"] [["holdsat",[[[["release",["	<pre>]]], "water"]], ["holdsat", [["perr]]], "water"]], ["holdsat", [["perr "pow", [["release", ["il", "m2"]]]] 'il", "m1"]]], "water"]], ["holdsat]], "water"]], ["holdsat", [["pow"</pre>	",["null"]],'water"]]],"pows":],"water"]],["holdsat",[["pow", t",[["pow",[["receive",	["11", "wwtp1", "m1"]]]], "wa ["wwtp1", "m2"]]]], "water"] ["wwtp1", "m1"]]]], "water"]	ter"]],["holdsat",[["perm",[] ter"]],["holdsat",[["perm",[],["holdsat",[["perm",[["iDis],["holdsat",[["perm",[["rele "holdsat",[["perm",[["release	"1D1scharge", charge", ase",
["wtpl", "ml"]	[]]] "water"]] ["ho]dsat" [["now"	".[["nerform".	["i1", "m2"]]]], "water"]],[["holdsat", [["perm", [["rece [["receive", ["wwtp1", "m1"]]	"holdsat",[["perm",[["release sive",["wwtp1","m2"]]]],"wate]]]."water"]].["bo]dsat",[["r	",["il","ml"]]]],"water"]], r"]],["holdsat",[["perm", erm".[["perform".
["wvtp1","m1", ["i1","wvtp1",	"tk"]]]],"water"]],["holdsat",["tk"]]]],"water"]],["holdsat",["m2"]]]],"water"]],["holdsat",["m1"]]]],"water"]],["holdsat",["pow",[["inform", ["pow",[["inform",	["wwtp1","m2","tk"]]]],"wa ["wwtp1","m1","tk"]]],"wa	<pre>ive",['wwtpl',"mit"]]], 'wate]]], 'water"]],['holdsat',[['prm",[ter']],['holdsat',[['perm",[ter']],['holdsat',[['perm",[</pre>	"perform", "inform",
[]},"observed" [["occurred",[["il","wwtpl",	':[["observed",[["inform",["il",' [["inform",["il","wwtpl","m2"]], "m2"]],"water"]]]}},{"metadata":	"[["null']], "water"]]], "tpows": "wtp1", "n2"]]]], "occurred": "water"]], ["occurred": "("answer_set_n":1, "answer_set_of":	["wwtpl","ml"]]]],"water"] [["holdsat",[["pow",[["rele [["release",["il","ml"]]]]],["holdsat",[["perm",[["disc],["holdsat",[["perm",["null" ease",["i1","m2"]]]],"water"] "water"]],["holdsat",[["pow"]],"water"]]],"pows":],["holdsat",[["pow", ',[["receive",
1, cost 10, mo tmp/tmpihnybi6		317219. "version":"1.0.1"}."state":	["wwtp1", "m2"]]]], "water"] ["wwtp1", "m1"]]]], "water"]],["holdsat",[["pow",[["reces],["holdsat",[["pow",[["perfo	ve', irm",
[["highMercury	<pre>",["m2"]],"water"]],["holdsat", "water"]] ["boldrat" [["inform</pre>	[["signedContract",	["wwtpl", "ml", "tk"]]]], "wa ["il", "wwtpl", "m2"]]]], "wa	<pre>ter"]],</pre>	inform", inform",
["11","wrtp1", [["iInform",[" ["i1","wrtp1",	"m2"]],"water"]]],"gpows":[],"1; 'il","wwtpl","m1"]],["iRelease", ."m1"]]]],"water"]]],"perms":[["	<pre>http://www.sectore.com/sector sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore.com/sectore</pre>	["11", "wwtp1", "m1"]]]], "wa ["wwtp1", "m2"]]]], "water"] ["wwtp1", "m1"]]]], "water"]	ter"]],["holdsat",[["pow",[["],["holdsat",[["pow",[["disch],["holdsat",[["pow",["null"]	discharge", arge",],"water"]]],"tpows":
["notosat",[["	["wwtpl" "m2"]]]] "water"]] ["bu	l']]]], "Water"]],["noldsat",[["perm", n]dsat",[["perm",[["inform",	<pre>[]}, "observed":[["observed [["receive",["wwtpl","m2"]] ["wwtpl","m2"]], "water"]]]]</pre>	',[["receive",["wwtp1","m2"]]],"water"]],["occurred",[["iF }],{"metadata":{"answer_set_r	<pre>]]], "occurred":[["occurred", leceive", ":1_"answer_set_of":</pre>
["il", "wrtpl", ["wrtpl", "ml",	"nl"]]],"water"]],["holdsat",[] "n2"]]]],"water"]],["holdsat",["tk"]]]],"water"]],["holdsat",[['perm',[['perform", ['perm',[['perform",	1, "cost":0, "mode": "multi_si tmp/tmpjhnvbj0f.lp"], "time:	not", "pid":7887, "source_files stamp":1554133316.2317219, "ve	<pre>1, water ' ; tpows :], "occurred":[["occurred", teceive", ":1,"answer_set_of": ":["/tmp/tmpryulg77.lp","/ ursion1:"1.0.1"}, "state":</pre>
	"tk"]]]], "water"]], ["holdsat", []], "water"]], ["holdsat", [["perm]], "water"]], ["holdsat", [["perm		[["highNercury", ["g2"]], "w	ater"]] ["bo]dsat" [["signed(water j,["noldsat",
["i1", "m1"]]]] ["holdsat", [["	<pre>, "water"]], ["holdsat", [["perm", perm", [["iDischarge", ["wwtpl", "n" ["wwtpl", "m2"]]]] "water"]] [""</pre>	<pre>[["release",["i1","n2"]]]],"water"]], al"]]]],"water"]], ["holdsat",[["perm", ["iInform", ["perm",[["iInform",</pre>	["holdsat", [["treated", ["w [], "obls": [["holdsat", [["ol ["il", "m]"]], ["failwraTolo	["holdsat",[["informed",["i1" vtpl","n2","tk"]],"water"]]], bl",[["iInform",["i1","wwtpl" form",["i1","wwtpl","m1"]]]]	"gpows":[],"ipows": ,"m1"]],["iRelease", "water"]]] "perms":
["il", "wrtpl", ["il", "wrtpl",	"m1"]]], water"]], ["holdsat",["m2"]]], water"]], ["holdsat",["tk"]]], "water"]], ["holdsat",[['perm',[['iInform', ['perm',[['iPerform',	[["holdsat",[["perm",["nul ["wwtp1","m1"]]]],"water"]	form",["il","wwtpl","ml"]]]], L"]],"water"]],["holdsat",[["],["holdsat",[["perm",[["disc],["holdsat",[["perm",[["info	perm",[["discharge", harge",
	"tk"]]]],"water"]],["holdsat",[] "tk"]]]],"water"]],["holdsat",[]]]],"water"]],["holdsat",[["perm		["il", "wvtpl", "m2"]]]], "water"] ["il", "wvtpl", "m1"]]]], "wa ["il", "wvtpl", "m2"]]]], "wa],["holdsat",[["perm",[["info ter"]],["holdsat",[["perm",[[ter"]],["holdsat",[["perm",[]	"inform", "perform".

"wwtpl" "m2"]]]] "water"]] ["holdsat" [["perm" [["iRe]ea

["wwtpl","ml","tk"]]]],"water"]],["holdsat",[["perm",[["perform

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
			00000		

Trace visualization



Introduction 00000	InstAL 000000	Deontic Sensors	Example ○○○○●	ODRL 00000	Epilogue 000
Trace visu	ualization				
<pre>[{"metadata":("answer set n":1,"answer set of":1,"cost":</pre>		:1,"cost":	["il","ml"]]]],"water"]],["holdsat",[["perm",[["iRelease	e*,

0, "mode": "multi_shot" tmpihnybi0f_lo"l_"tim	<pre>set i, asset set or i, cost or i, cost or i, cost or i, i i i i i i i i i i i i i i i i i i</pre>	e
<pre>{"holdsat":{"fluents" [],"tpows":[]},"obser</pre>	:(], "opows":(], "ipows":(], "obls":(], "perms":(], "pows": [["discharge", ["wwtpl", "n2"]]], "wster"]), ["holdsat", [["pow", [["inform", """"]]]], "wster"]), ["holdsat", [["pow", [["inform", """]]]], "wster"]), ["holdsat", [["pow", [["inform", """]]]], "wster"]), ["holdsat", [["pow", [["inform", """]]]], "wster"]), ["holdsat", [["pow", [["inform", ""]]]], "wster"]), ["holdsat", [["inform", ""]]], ["holdsat", [["inform", ""]]]], "wster"]), ["holdsat", [["inform", ""]]], ["holdsat", [["inform", ""]]], ["holdsat", [["inform", ""]]]], ["holdsat", [["inform", ""]]], ["holdsat", [["inform", "]]], ["holdsat", ["inform", "]]], ["holdsat", "]]], ["holdsat", "]]], ["holdsat", "]]], ["holdsat", "]	
1, "answer_set_of":1," tmprgyllg77.lp", "/tmp	inform(i1,wwtp1,m2): water 🔶	
1554133316.2317219, "v [["informed",["i1", "w	ilnform(il wwth1 m2); water	
["wwtpl","i1"]],"wate [["live",["water"]],"	release(i1,wwtp1,m2): water	
[["iInform",["i1","ww ["i1","wwtp1","n1"]]]	iRelease(i1 wwth1 m2); water	
["i1","m2"]]]],"water ["holdsat",[["perm",[receive/united i i m2); unster	F" :
[["iReceive",["wwtp1" ["wwtp1","m2","tk"]]]	iReceive(wwtp1,i1,m2): water	
["wwtpl","ml","tk"]]] ["il","wwtpl","m2"]]]	perform(wwtp1,n2,tk): water	
["il", "wwtp1", "n1"]]] ["wwtp1", "m2"]]]], "wa	Devform(united 1 m2 th) under	
["wwtp1","m1"]]]],"wa ["i1","m2"]]]],"water	discharge(wwtp1,m2): water	
["holdsat",[["perm",[[["receive",["wwtpl",	discharge(witp1,m2): Water	
["wwtpl", "m2", "tk"]]] ["wwtpl", "m1", "tk"]]]	iDischarge(wwtp1,m2): water	
["il", "wwtpl", "m2"]]] ["il", "wwtpl", "m1"]]]	release(i2,wvtp2,m1): water	
["wwtpl", "m2"]]]], "wa ["wwtpl", "m1"]]]], "wa	iRelease(i2.wwtp2,m1): water	
<pre>[["holdsat",[["pow",[[["release",["il","ml</pre>	(D)	
["wwtpl","m2"]]]],"wa ["wwtpl","m1"]]]],"wa	uncontractedRelease(12, wwtp2,m1): Water [, "water [, "water [, "water [, "water [, "l"] perm	
["wwtp1","m2","tk"]]] ["wwtp1","m1","tk"]]]	iReceive(wwtp2,i2,m1): water	
["i1","wwtp1","m2"]]] ["i1","wwtp1","m1"]]]		
["wwtp1","m2"]]]],"wa ["wwtp1","m1"]]]],"wa	discharge(wwtp2,m1): water ♦ illegalDischarge(wwtp2,m1): water ♦ pows*1	
<pre>[]}, "observed":[["obs [["occurred",[["infon</pre>	inget in the second s	
["i1","wwtp1","m2"]], 1,"cost":0,"mode":"mu	highMercury(m2): water	
<pre>tmp/tmpjhnvbj0f.lp"], {"holdsat":{"fluents"</pre>	illegalBecause(untreated,wwtp2,m1): water	
<pre>[["highMercury",["m2" ["wwtp1","i1"]],"wate</pre>		
["il","wwtp1","m2"]], [["iInform",["il","ww	informed(i1,wwtp1,m2): water	
["il","wwtpl","nl"]]] ["holdsat",[["perm",[live(water): water	rred",
<pre>[["discharge",["wwtpl ["i1","wwtp1","m1"]]]</pre>	obi(inform(i1,wwtp1,m1),iRelease(i1,wwtp1,m1),failure foinform(i1,wwtp1,m1)): water	
["i1","wwtp1","m2"]]] ["wwtp1","m1","tk"]]]	signedContract(wwtp1,i1): water	
["wwtp1","m2","tk"]]] ["wwtp1","m1"]]]],"wa	treated(wwtp1,m2,tk): water	
["wwtp1","m2"]]]],"wa ["i1","m1"]]]],"water ["holdsat",[["perm",[treating(wwtp1,m2): water	11.
["noldsat",[["perm",[[["iDischarge",["wwtp ["i1","wwtp1","m1"]]]	treating(wwtp2,m1): water	
["il"."wwtp1"."m2"]]]	rge:,],"water"]],["holdsat",[["perm",[["iPerform", ["wtpl", "m2"]]]],"water"]],["holdsat",[["perm",[["inform",	
["wwtp1","m2","tk"]]]], water]], [holdsat ,[[pem ,[[riecom,]]], [water]], [holdsat ,[[pem ,[[rinom,]]], [holdsat ,[[pem ,[[rinom,]]]]], water]], [holdsat ,[[pem ,[[rinom,]]]]], [holdsat ,[[pem ,[[rinom,]]]]], [holdsat ,[[pem ,[[rinom,]]]]], [holdsat ,[[pem ,[rinom,]]]], [holdsat ,[[pem ,[rinom,]]]], [holdsat ,[rinom,]]], [holdsat ,[rinom,]]], [holdsat ,[rinom,]]], [holdsat ,[rinom,]], [holdsat ,[rinom,]]], [holdsat ,[rinom,]]], [holdsat ,[rinom,]], [holdsat ,[rinom,]]], [holdsat ,[rinom,]]]], [holdsat ,[rinom,]]], [holds	

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
				0000		
Contents						

1 Introduction

- 2 InstAL: a DSL for norm modelling
- 3 Deontic Sensors: normative reasoning as a service
- 4 Sample water management policy
- **5** Semantic policy representation

6 Epilogue

00000				0000	
Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue

Where's the semantics?

• unfortunately, each model is "just another program"

- where "illegalDischarge" is just a string
- and the implementation is the programmer's interpretation
- need to connect real world to model (automatically)
- ullet natural language o model?
- ullet semantic representation of policy o model?
- W3C's Open Digital Rights Language (ODRL)
- Originally conceived for asset rights management: early 2000s
- ODRL 2.2 (2018) generalizes to policy... to some degree

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
				0000		
NA/I I .		• •				
Where's t	he semant:	ICS ?				

- unfortunately, each model is "just another program"
- where "illegalDischarge" is just a string
- and the implementation is the programmer's interpretation
- need to connect real world to model (automatically)
- ullet natural language o model?
- ullet semantic representation of policy o model?
- W3C's Open Digital Rights Language (ODRL)
- Originally conceived for asset rights management: early 2000s
- ODRL 2.2 (2018) generalizes to policy... to some degree

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
				0000		
Where's t	he semant:	CICS (

- unfortunately, each model is "just another program"
- where "illegalDischarge" is just a string
- and the implementation is the programmer's interpretation
- need to connect real world to model (automatically)
- ullet natural language o model?
- semantic representation of policy ightarrow model?
- W3C's Open Digital Rights Language (ODRL)
- Originally conceived for asset rights management: early 2000s
- ODRL 2.2 (2018) generalizes to policy... to some degree

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
				0000		
Where's t	he semant:	CICS (

- unfortunately, each model is "just another program"
- where "illegalDischarge" is just a string
- and the implementation is the programmer's interpretation
- need to connect real world to model (automatically)
- ullet natural language o model?
- semantic representation of policy ightarrow model?
- W3C's Open Digital Rights Language (ODRL)
- Originally conceived for asset rights management: early 2000s
- ODRL 2.2 (2018) generalizes to policy... to some degree

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue		
				0000			
		• •					
Where's the semantics?							

- unfortunately, each model is "just another program"
- where "illegalDischarge" is just a string
- and the implementation is the programmer's interpretation
- need to connect real world to model (automatically)
- natural language \rightarrow model?
- semantic representation of policy ightarrow model?
- W3C's Open Digital Rights Language (ODRL)
- Originally conceived for asset rights management: early 2000s
- ODRL 2.2 (2018) generalizes to policy... to some degree

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue			
				0000				
10/1		·						
Where's the semantics?								

- unfortunately, each model is "just another program"
- where "illegalDischarge" is just a string
- and the implementation is the programmer's interpretation
- need to connect real world to model (automatically)
- natural language ightarrow model?
- semantic representation of policy ightarrow model?
- W3C's Open Digital Rights Language (ODRL)
- Originally conceived for asset rights management: early 2000s
- ODRL 2.2 (2018) generalizes to policy... to some degree

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
00000	000000	0000000000		0000		
VVhere's t	he semant:	LICS (

- unfortunately, each model is "just another program"
- where "illegalDischarge" is just a string
- and the implementation is the programmer's interpretation
- need to connect real world to model (automatically)
- natural language ightarrow model?
- semantic representation of policy ightarrow model?
- W3C's Open Digital Rights Language (ODRL)
- Originally conceived for asset rights management: early 2000s
- ODRL 2.2 (2018) generalizes to policy... to some degree

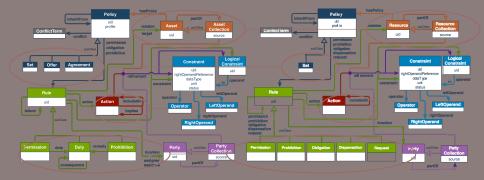
Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
				00000		
Where's t	he semant:	ics?				

- unfortunately, each model is "just another program"
- where "illegalDischarge" is just a string
- and the implementation is the programmer's interpretation
- need to connect real world to model (automatically)
- natural language ightarrow model?
- ullet semantic representation of policy o model?
- W3C's Open Digital Rights Language (ODRL)
- Originally conceived for asset rights management: early 2000s
- ODRL 2.2 (2018) generalizes to policy... to some degree

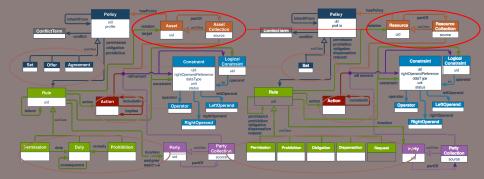
Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
				00000		
Where's t	he semant:	ics?				

- unfortunately, each model is "just another program"
- where "illegalDischarge" is just a string
- and the implementation is the programmer's interpretation
- need to connect real world to model (automatically)
- natural language ightarrow model?
- semantic representation of policy \rightarrow model?
- W3C's Open Digital Rights Language (ODRL)
- Originally conceived for asset rights management: early 2000s
- ODRL 2.2 (2018) generalizes to policy... to some degree

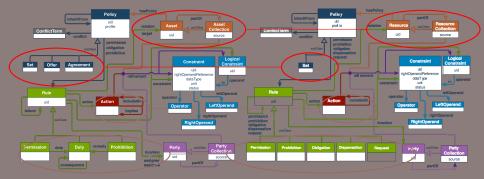
Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
				00000	



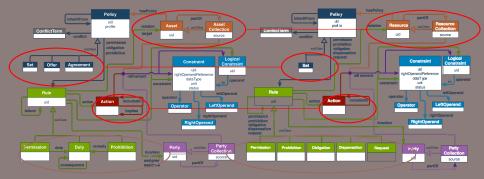
Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
				00000	



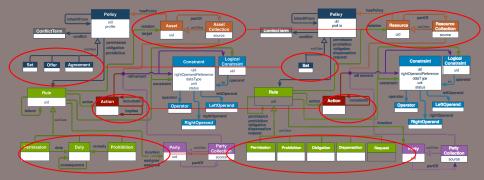
Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
				00000	



Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
				00000	



Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
				00000	



Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
				00000		
Example O		, complianco	roquost			

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
				00000		
.						
Data prot	ection					

- use-case: fragments of articles of GDPR
- check business process compliance with GDPR
- H2020 SPECIAL project
- develop ODRL ightarrow InstAL translator
- aim to synthesize ODRL from natural language policies

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
				00000		
D						
Data nrot	tection					

- use-case: fragments of articles of GDPR
- check business process compliance with GDPR
- H2020 SPECIAL project
- ullet develop ODRL o InstAL translator
- aim to synthesize ODRL from natural language policies

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
				00000		
Data protection						

- use-case: fragments of articles of GDPR
- check business process compliance with GDPR
- H2020 SPECIAL project
- ullet develop ODRL o InstAL translator
- aim to synthesize ODRL from natural language policies

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
				00000		
Data prot	ection					

- use-case: fragments of articles of GDPR
- check business process compliance with GDPR
- H2020 SPECIAL project
- ${\color{black}\bullet}$ develop ODRL ${\color{black}\to}$ InstAL translator
- aim to synthesize ODRL from natural language policies

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue	
				00000		
Data prot	ection					

- use-case: fragments of articles of GDPR
- check business process compliance with GDPR
- H2020 SPECIAL project
- ${\color{black}\bullet}$ develop ODRL ${\color{black}\to}$ InstAL translator
- aim to synthesize ODRL from natural language policies

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	000000		00000	00000	●○○
Contents					

1 Introduction

- 2 InstAL: a DSL for norm modelling
- 3 Deontic Sensors: normative reasoning as a service
- 4 Sample water management policy
- 5 Semantic policy representation

6 Epilogue

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	000000		00000	00000	○●○
Summary					

• formal model of institutions: policies, ...

- (formal model of (directed) bridges: connect institutions)
- computational model
- achieves:
 - refactoring and decoupling of normative reasoning
 - publication of normative reasoning as a service
- enables/facilitates:
 - testing of normative models.
 - sharing/re-use of normative models.
 - certification of normative models (of regulation)
 - regulatory compliance-checking as a service

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	000000	00000000000	00000	00000	○●○
Summary					

- formal model of institutions: policies, ...
- (formal model of (directed) bridges: connect institutions)
- computational model
- achieves:
 - refactoring and decoupling of normative reasoning
 - publication of normative reasoning as a service
- enables/facilitates:
 - testing of normative models.
 - sharing/re-use of normative models.
 - certification of normative models (of regulation)
 - regulatory compliance-checking as a service

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
					000
Summary					

- formal model of institutions: policies, ...
- (formal model of (directed) bridges: connect institutions)
- computational model
- achieves:
 - refactoring and decoupling of normative reasoning
 - publication of normative reasoning as a service
- enables/facilitates:
 - testing of normative models.
 - sharing/re-use of normative models.
 - certification of normative models (of regulation)
 - regulatory compliance-checking as a service

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	000000	00000000000	00000	00000	○●○
Summary					

- formal model of institutions: policies, ...
- (formal model of (directed) bridges: connect institutions)
- computational model
- achieves:
 - refactoring and decoupling of normative reasoning
 - publication of normative reasoning as a service
- enables/facilitates:
 - testing of normative models.
 - sharing/re-use of normative models.
 - certification of normative models (of regulation)
 - regulatory compliance-checking as a service

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					000
Summary					

- formal model of institutions: policies, ...
- (formal model of (directed) bridges: connect institutions)
- computational model
- achieves:
 - refactoring and decoupling of normative reasoning
 - publication of normative reasoning as a service
- enables/facilitates:
 - testing of normative models
 - sharing/re-use of normative models.
 - certification of normative models (of regulation)
 - regulatory compliance-checking as a service

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					000
Summary					

- formal model of institutions: policies, ...
- (formal model of (directed) bridges: connect institutions)
- computational model
- achieves:
 - refactoring and decoupling of normative reasoning
 - publication of normative reasoning as a service
- enables/facilitates:
 - testing of normative models
 - sharing/re-use of normative models
 - certification of normative models (of regulation)
 - regulatory compliance-checking as a service

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					000
Summary					

- formal model of institutions: policies, ...
- (formal model of (directed) bridges: connect institutions)
- computational model
- achieves:
 - refactoring and decoupling of normative reasoning
 - publication of normative reasoning as a service
- enables/facilitates:
 - testing of normative models
 - sharing/re-use of normative models
 - certification of normative models (of regulation)
 - regulatory compliance-checking as a service

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
					000
Summary					

- formal model of institutions: policies, ...
- (formal model of (directed) bridges: connect institutions)
- computational model
- achieves:
 - refactoring and decoupling of normative reasoning
 - publication of normative reasoning as a service
- enables/facilitates:
 - testing of normative models
 - sharing/re-use of normative models
 - certification of normative models (of regulation)
 - regulatory compliance-checking as a service.

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000					000
Summary					

- formal model of institutions: policies, ...
- (formal model of (directed) bridges: connect institutions)
- computational model
- achieves:
 - refactoring and decoupling of normative reasoning
 - publication of normative reasoning as a service
- enables/facilitates:
 - testing of normative models
 - sharing/re-use of normative models
 - certification of normative models (of regulation)
 - regulatory compliance-checking as a service

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
					000
Discussio	n				
Discussion					

- how to share and test policy models?
- how to certify policy models?
- how to discover policy models for re-use?
- how to record policy states for audit?
- how to capture written policy formally?
- how to capture policy heterarchies?
- how to revise a policy: which is the master copy?

QUESTIONS WELCOME!

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	000000	00000000000	00000	00000	○O●
Discussion					

- how to share and test policy models?
- how to certify policy models?
- how to discover policy models for re-use?
- how to record policy states for audit?
- how to capture written policy formally?
- how to capture policy heterarchies?
- how to revise a policy: which is the master copy?

QUESTIONS WELCOME!

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
					000
Discussio	n				

- how to share and test policy models?
- how to certify policy models?
- how to discover policy models for re-use?
- how to record policy states for audit?
- how to capture written policy formally?
- how to capture policy heterarchies?
- how to revise a policy: which is the master copy?

QUESTIONS WELCOME!

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
					000
Discussio	n				

- how to share and test policy models?
- how to certify policy models?
- how to discover policy models for re-use?
- how to record policy states for audit?
- how to capture written policy formally?
- how to capture policy heterarchies?
- how to revise a policy: which is the master copy?

QUESTIONS WELCOME!

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
					000
Discussio	n				

- how to share and test policy models?
- how to certify policy models?
- how to discover policy models for re-use?
- how to record policy states for audit?
- how to capture written policy formally?
- how to capture policy heterarchies?
- how to revise a policy: which is the master copy?

QUESTIONS WELCOME!

	Example	ODRL	Epilogue
			000
Discussion			

- how to share and test policy models?
- how to certify policy models?
- how to discover policy models for re-use?
- how to record policy states for audit?
- how to capture written policy formally?
- how to capture policy heterarchies?
- how to revise a policy: which is the master copy?

QUESTIONS WELCOME!

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
					000
Discussion	า				

- how to share and test policy models?
- how to certify policy models?
- how to discover policy models for re-use?
- how to record policy states for audit?
- how to capture written policy formally?
- how to capture policy heterarchies?
- how to revise a policy: which is the master copy?

QUESTIONS WELCOME!

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
					000
Discussion	า				

- how to share and test policy models?
- how to certify policy models?
- how to discover policy models for re-use?
- how to record policy states for audit?
- how to capture written policy formally?
- how to capture policy heterarchies?
- how to revise a policy: which is the master copy?

QUESTIONS WELCOME!

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
					000
Discussion	า				

- how to share and test policy models?
- how to certify policy models?
- how to discover policy models for re-use?
- how to record policy states for audit?
- how to capture written policy formally?
- how to capture policy heterarchies?
- how to revise a policy: which is the master copy?

QUESTIONS WELCOME!

Introduction	InstAL	Deontic Sensors	Example	ODRL	Epilogue
00000	000000		00000	00000	000
Bibliograp	hy I				

Aldewereld, Huib, Olivier Boissier, Virginia Dignum, Pablo Noriega, and Julian Padget, eds. (2016). Social Coordination Frameworks for Social Technical Systems. Vol. 30. Law, Governance and Technology. Springer. DOI: 10.1007/978-3-319-33570-4_.

- Behrens, Tristan, Koen Hindriks, and Jürgen Dix (2011). "Towards an environment interface standard for agent platforms". In: Ann. Math. Artif. Intell. 61.4, pp. 261–295. DOI: 10.1007/s10472-010-9215-9.
 - Bordini, Rafael, Jomi Hübner, and Michael Wooldridge (2007). *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons, Ltd. DOI: 10.1002/9780470061848.ch4.
 - Harré, Rom and Paul F Secord (1972). "The explanation of social behaviour.". In:
 - Jones, A.J.I. and M. Sergot (1996). "A formal characterisation of institutionalised power". In: *Logic Journal of IGPL* 4.3, pp. 427–443.

Introduction 00000	InstAL 000000	Deontic Sensors	Example 00000	0DRL 00000	Epilogue 000	
Bibliograp	ohy II					

Lee, Jeehang, Julian Padget, Brian Logan, Daniela Dybalova, and Natasha Alechina (2014). "N-Jason: Run-Time Norm Compliance in AgentSpeak(L)". In: Engineering Multi-Agent Systems - Second International Workshop, EMAS 2014, Paris, France, May 5-6, 2014, Revised Selected Papers. Ed. by Fabiano Dalpiaz, Jürgen Dix, and M. Birna van Riemsdijk. Vol. 8758. Lecture Notes in Computer Science. Springer, pp. 367–387. DOI: 10.1007/978-3-319-14484-9_19.

Noriega, Pablo, Jordi Sabater-Mir, Harko Verhagen, Julian Padget, and Mark d'Inverno (2017). "Identifying Affordances for Modelling Second-Order Emergent Phenomena with the WIT Framework". In: Autonomous Agents and Multiagent Systems - AAMAS 2017 Workshops, Visionary Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers. Ed. by Gita Sukthankar and Juan A. Rodriguez-Aguilar. Vol. 10643. Lecture Notes in Computer Science. Springer, pp. 208–227. DOI: 10.1007/978-3-319-71679-4_14.

North, Douglas (1991). Institutions, Institutional Change and Economic Performance. CUP.

Ostrom, Elinor (1990). Governing the Commons. The Evolution of Institutions for Collective Action. CUP.

Introduction	InstAL	Deontic Sensors	Example	0DRL	Epilogue
00000	000000		00000	00000	000
Bibliograp	ohy III				

Padget, Julian, Emad Elakehal, Tingting Li, and Marina De Vos (2016a). "InstAL: An Institutional Action Language". In: Social Coordination Frameworks for Social Technical Systems. Springer, pp. 101–124.

- Padget, Julian, Emad ElDeen Elakehal, Tingting Li, and Marina De Vos (2016b). "InstAL: An Institutional Action Language". In: Social Coordination Frameworks for Social Technical Systems. Springer, pp. 101–124. DOI: 10.1007/978-3-319-33570-4_6.
- Padget, Julian, Marina De Vos, and Charlie Ann Page (July 2018). "Deontic Sensors". In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. International Joint Conferences on Artificial Intelligence Organization, pp. 475–481. DOI: 10.24963/ijcai.2018/66.
 - Searle, John (1995). *The Construction of Social Reality*. Allen Lane, The Penguin Press.

Wright, Georg von (1951). "Deontic Logic". In: Mind 60.237, pp. 1–15.