From Funny Logics via Funny Programming to Funny Applications

John-Jules Ch. Meyer
Intelligent Systems Group
Universiteit Utrecht

From Agent Logics via Agent Programming to Agent Applications

John-Jules Ch. Meyer
Intelligent Systems Group
Universiteit Utrecht

From Agent Logics via Agent Programming to Funny Applications

John-Jules Ch. Meyer
Intelligent Systems Group
Universiteit Utrecht

Introduction: Intelligent Agents

(Cognitive) Agents

- Software / hardware entities that display a certain degree of *autonomy* / taking initiative, are proactive/goal-directed
- Mostly described in terms of having 'mental states' ('*strong*' notion of agency) *'cognitive' agents*
- Display informational and motivational attitudes

Agent metaphor

- From an *engineering perspective*, the agent's *metaphor* (i.e. using agent concepts metaphorically) helps to *design* and *construct* complicated (distributed) systems!!
- Example: (multi) robotic systems
  - *Cognitive robotics*

Ideas behind agents

- Stemming from philosophy
  - Practical reasoning, reasoning about actions
  - Characterization of rational decision-making
  - Balancing *desires* and *beliefs*
  - Interplay between *beliefs*, *desires*, *intentions* (Bratman)
  - Intentional stance (Dennett)

Practical Reasoning

- (Bona fide) practical syllogism

Exercise would be good for me.
Jogging is exercise.

Therefore, jogging would be good for me.

- 'Just' deductive reasoning

Practical Reasoning

- More interesting practical syllogism

Would that I exercise.
Jogging is exercise.

Therefore, I shall go jogging

- No deduction, rather specification of selection of action / decision of the agent

Dennett's intentional stance

- The *intentional stance* is the strategy of interpreting the behaviour of an entity by treating it *as if it were* a *rational agent* that governed its *choice* of *action* by a *consideration* of its *beliefs* and *desires*
  - Anthropomorphic instance of the *design (functionality) stance*, contra the *physical stance*
  - Instrumental / operational use of beliefs and desires of *human beings*: no causally active inner states of people, just calculational devices

Bratman : the role of intentions

- *Rational* behavior needs, besides *beliefs* and *desires*, also *intentions*
- Two justifications for this:
  - (Resource-bounded)agents need to settle on some desire(s) and commit themselves
  - Co-ordination of future actions after commitment(s)

Bratman

- Intentions, *unlike mere desires*, play the following functional roles:
  - Intentions normally pose *problems* for the agent; the agent needs to determine a way to achieve them □ *focus on solving concrete problems*
  - Intentions provide a "screen of admissibility" for *adopting other intentions*
  - Agents "track" the success of their attempts to achieve their intentions -- may give rise to *replanning*

Agent Logics

Agent logics

- Logics for specifying intelligent/rational agents inspired by Bratman's philosophy:
  - BDI logic
  - Cohen & Levesque
  - KARO logic
- BDI model/architecture (Rao & Georgeff)

Agent logics

- philosophical logic
- a formal treatment of intensional notions
- various 'flavours':
  - epistemic / doxastic
  - temporal / dynamic (action logic)
  - deontic
  - combinations (BDI, KARO)

Cohen & Levesque

- Achievement goals
  A-GOAL i □ = GOAL i (LATER □) □ BEL i ¬ □
- No deferral forever assumption
  ⊨ □¬(GOAL i (LATER □))
  - Agents eventually drop all achievement goals!

Cohen & Levesque

- Persistent goals

P-GOAL i □ =
GOAL i (LATER □) □
BEL i ¬□ □
[BEFORE(BEL i □ BEL i □¬□)
¬GOAL i (LATER □)]

Cohen & Levesque

- Intention ('intend-to-do')

- INTEND$_1$ i □ = P-GOAL i [DONE i
  (BEL i (HAPPENS □))?; □]

KARO logic

Van Linder, Van der Hoek, Meyer

- motivational attitudes
  - PossIntend(□, □) □ <commit□>□Com(□)
  - PossIntend(□, □) □ ¬Auncommit□
  - Com(□) □ <uncommit□> ¬Com(□)
  - Com(□) □ BCom(□)
  - Com($□_1$; $□_2$) □ Com($□_1$) □ B[$□_1$]Com($□_2$)
  - Com(□) □ ¬Can(□,true) □
    Can(uncommit□, ¬Com(□))

Rao & Georgeff : BDI theory

- *"Rational agent* possesses mental attitudes of *beliefs, desires and intentions,* representing the *information, motivational, and deliberative states of an agent*, respectively"
- "These mental attitudes determine the system's behaviour and are critical for achieving *adequate* or *optimal* performance when deliberation is subject to *resource bounds*" --- *computational* perspective!

Rao & Georgeff's BDI Logic

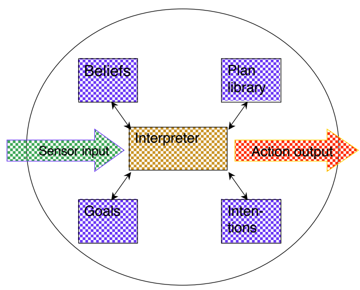
- Commitment strategies in BDI logic
  - INTEND(□) inevitable□(¬INTEND(□))
    "no infinite deferral"
  - INTEND(inevitable□□)
    inevitable(INTEND(inevitable□□) U BEL(□))
    "blindly committed agent"

Rao & Georgeff's BDI Logic

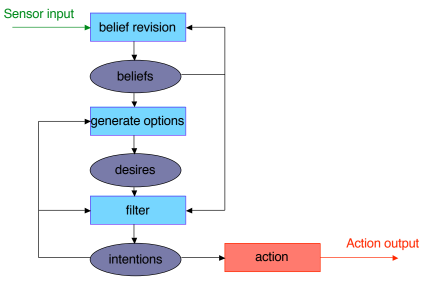
- INTEND(inevitable□□)
  inevitable(INTEND(inevitable□□) U
  (BEL(□) ¬BEL(optional□□)))
  "single-minded committed agent"
- INTEND(inevitable□□)
  inevitable(INTEND(inevitable□□) U
  (BEL(□) ¬GOAL(optional□□)))
  "open minded committed agent"

Agent design & programming

BDI Architecture



BDI architecture: 'deliberation cycle'



Agent-oriented programming (Y. Shoham)

- AOP = programming *mental states*
  - meaning of an AOP program is a *'mental state transformer'*
  - BDI agent programming languages
    - Agent-0 Shoham 1993
    - AgentSpeak(L) Rao 1996, Bordini et al.
    - 3APL Hindriks et al. 1999, Dastani et al.
    - AF-APL Collier et al. 2004
- Representation of mental attitudes
  - which mental attitudes? how represented? semantics?
    - AgentSpeak beliefs, intentions, events
    - 3APL '99 beliefs, plans

The language 3APL

Hindriks et al.

- Attempt to get a 'true' agent language using 'mental' (BDI-like) concepts
  - So agent concepts used in *implementation*
- Supplied with *formal semantics*
- Mixture of *imperative* and *logic programming* aspects

Mental attitudes in 3APL

BDI theory	3APL
Beliefs	Beliefs
Desires	
Intentions	Plans (procedural)

Mental attitudes in 3APL

BDI theory	3APL (new version)
Beliefs	Beliefs
Desires	Goals (declarative)
Intentions	Plans (procedural)

3APL agent (original version)

- a complex mental state incorporating
  - beliefs about the agent's environment
  - plans describing actions to achieve the goals
- set of mechanisms working on mental state
  - to execute plans (controlling the environment)
  - for decision-making or practical reasoning

    (*plan revision, goal planning*)
- a set of capabilities, i.e. basic actions

3APL program

- a set of capabilities: basic actions:
  - e.g. `gripper_up, pickup, move_left, move_right, sense`
- an initial belief base: simple propositions:
  - e.g. `block_on_table`
- a set of initial plans: imperative programs:
  - e.g. `gripper_up ; pickup`

3APL program (ctd)

- a set of plan revision rules: guarded clauses of the form □□ □|□', where
  - □ is a plan,
  - □ is a guard and
  - □' is a (revised) plan
- e.g. `gripper_up;pickup` □ `no_block | find_block;gripper_up;pickup`
- If the guard is implied by the agent's belief base the rule becomes applicable and *may* be applied.

3APL control architecture

- the control architecture implements the deliberation or (Sense)-Update-Act cycle:
  - Rule application phase (plan generation / updating);
  - Execution Phase (belief updating by plan execution)

3APL control loop ('deliberation cycle')

- 1. Find rules matching plans (= commitments)
- 2. Select rules from (1) matching the beliefs.
- 3. Select rule from (2) and fire it on plan base

- 4. Select plans that can be executed
- 5. Select one plan from (4) and execute it

3APL: extensions

Dastani, Van Riemsdijk et al.

- Extend language, e.g.
  - Declarative goals □
  - Plan generation and goal revision rules
    - □□ □|□ (plan generation)
    - □□ □|□' (goal revision)
  - Communication primitives
  - *Programmable* control (deliberation) loop
  - Nested modalities, e.g. B_iG_j□

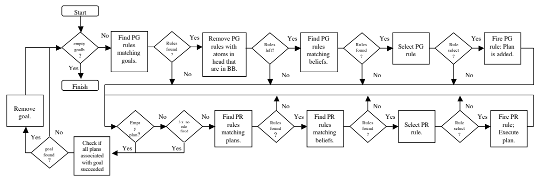
Extended 3APL agent

- a complex mental state incorporating
  - beliefs about the agent's environment
  - plans, describing actions to achieve the goals
  - goals, representing the states of affairs to be achieved
- set of mechanisms working on mental state
  - to execute plans (controlling the environment)
  - for decision-making or practical reasoning (plan revision, plan generation)
- a set of capabilities, i.e. basic actions

Extended Deliberation Cycle

1. Find Plan Generation Rules that Match Goals
2. Remove Plan Generation Rules with atoms in head that exist in Belief Base
3. Find Plan Generation (PG) Rules that Match Beliefs
4. Select a Plan Generation (PG) Rule to Apply
5. Apply the Plan Generation (PG) Rule, thus adding a plan to the planbase
6. Find Plan Revision (PR) Rules that Match Plans
7. Find Plan Revision (PR) Rules that Match Beliefs
8. Select a Plan Revision (PR) Rule to Apply to a Plan
9. Apply the Plan Revision (PR) Rule to the Plan
10. Find Plans To Execute
11. Select a Plan To Execute
12. Execute the (first basic action of the) Plan

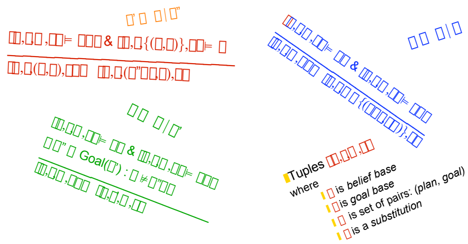
Extended Deliberation Cycle



Formal Semantics

- Designing a programming language
  - define the constructs of the language
  - define the semantics of a program in this language
  - define "what happens" if the program is executed
- Formal semantics
  - give meaning to programs formally and precisely
  - many advantages
    - *problems become clear*
    - *comparison with other languages*
    - *basis for verification*

Formal Semantics



Towards program verification

Birna van Riemsdijk

- Dynamic logic for 3APL
  - in PDL: $[\pi_1;\pi_2]\phi \leftrightarrow [\pi_1][\pi_2]\phi$
  - this is *not* a validity for 3APL plans
  - dynamic logic for restricted plans
    - sound and complete axiomatization
  - extendable to logic for arbitrary plans
    - but.... infinitary axiomatization
    - not (yet) really a practical method for program verification

Programming emotional agents

with M. Dastani & R, van der Ree

- Main ideas:
  - Inspired by emotions of humans
    - Emotional are *not opposed to rational* behaviour, rather *complementary* (Damasio)
  - Focus on *functional role* of emotions
    - Emotions help structuring agent behaviour / design
      - *"Emotional states organize ready repertoires of action"*
      - *"Emotions are heuristics"*
  - Our perspective: emotions as a 'designing tool' for agents

Emotions in KARO and 3APL

Happiness

- PossIntend(□, □) □ Com(□) □ happy(□,□) □ [*deliberate*]□(PossIntend(□, □) □ Com(□))
- I.e. *persistence* of PossIntend and Com through this deliberation step

Sadness

- PossIntend(□, □) □ Com(□) □ sad(□,□) □
  [*deliberate*]□(¬PossIntend(□, □) ¬Com(□)
  Com(if Can(□, □) then □ else *replan*(□',□); □' fi))

In 'meta'-3APL:

- if PossIntend(□, □) □ Com(□) □ sad(□,□) then: [drop_goal(□) + drop_plan(□) + (¬Can(□,□)?; replan(□',□); □') + (Can(□,□)?; □)]

Angriness

- angry(□,□,□) □ [*deliberate*](Com(*stit*(Can(□,□)))

In extended 3APL:

- angry(□,□,□) □ Can(□,□)

Multi-agent programming

- Sources from different areas:
  - Distributed AI / problem solving
  - Concurrent programming
- Special issues:
  - Balancing the co-operation aspects with the individual agent's autonomy
    - How to connect agent's BDI attitudes to the system as a whole?
    - Communication & Coordination
  - Roles: enactment / 'deactment'

Multi-Agent 3APL: communication

Van Eijk, De Vries, et al.

- *Communication* between agents, using techniques from both *constraint programming* and *communicating sequential processes (CSP)*
- Communication primitives
  - Speech acts
- Typical (FIPA/KQML-like) construct:
  - *send*(agent, performative, content)

Heterogeneous MAS: Ontology negotiation

Van Diggelen et al.

- Research on ONP's that
  - Enable *lazy* ontology alignment
  - Give rise to *minimal shared ontology*
  - Guarantee *sound and lossless* communication
  - Are based on *'Gricean'-like maxims*
    - E.g. what the sender did *not* say (when he could be more specific), he does not know either
  - Tested on case with news publishers (using different ontologies)

AO Methodology

Dastani, Hulstijn, Dignum, Meyer

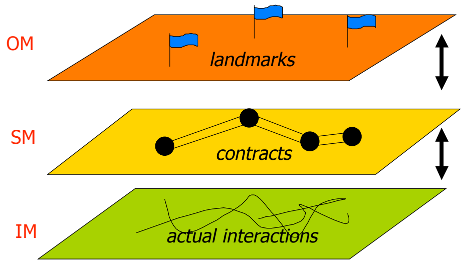
- Need for separate methods for AOP
  - AOP is not simply a variant/instance of OOP
  - Needed: relation between behavior of *complex MASs* and that of the *constituent individual* agents
    - *Norms*
    - Institutions
    - Protocols
    - …

OperA

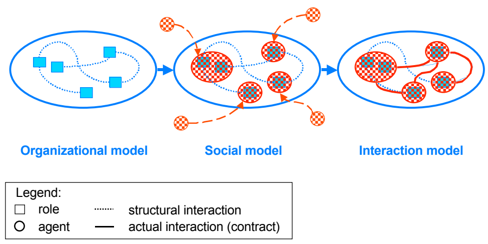
Virginia Dignum et al.

- Organizational Model
  - represents *organizational aims and requirements*
  - *roles, interaction structures, scene scripts, norms*
- Social Model
  - represents *agreements concerning participation of individual agents ('hiring agents for playing roles')*
- Interaction Model
  - represents *agreements concerning interaction between the agents themselves*

OperA: 3-tiers specification



OperA



Consequence of OperA

- Separation between the individual (agent) and collective (society) level
  - For the construction of individual agents you can use what you want, e.g. *BDI model*!!
  - Link between the individual and organisation via interaction structure, roles, norms and *contracts*!

From analysis via design to implementation

- *Roles* from analysis □ *agent types* in 3APL
- Agent type:
  - specification of *deliberation process* +
  - set of roles (characterized in terms of *beliefs*, *goals*, *plans*, *capabilities*, *messages*, *PR/PG/GR rules*)
- *Norms* may be implemented in various ways:
  - as *goals*
  - in *social or interaction structure*
    - obligations, protocols
  - in *environment*
    - norm enforcement

Extending 3APL with obligations? Towards BDI+ (eg BOID theory)

- One might also consider augmenting 3APL by rules dealing with obligations directly:

  □□□ □|□ (goal generation by obligations)

  or even

  □□□ □,□|□ (goal generation by obligations + desires)

- Are these sensible in practice?

Normative Agent Systems

Aldewereld, Grossi, Dignum

- Problem: how to control agents in an *open* multi-agent system / agent society?
  - Electronic Institutions
  - Norms vs protocols

Protocols in Normative Agent Systems

- Norms tend to be vague and ambiguous
- Protocols in (electronic) institutions:
  - Protocols can help agents follow procedure
  - Protocols can help obtain results
  - Protocols should be norm-compliant
  - Following the protocol does not violate any of the norms
- How to prove norm compliance of protocols?

Verifying Norm Compliancy of Protocols

Aldewereld et al.

- Formal methods based on program verification
- Translate protocol to program, norms to LTL formulas
  - Intermediate states are important! use temp. logic
- Need to connect the abstract level to the concrete level
- Norm compliance is a safety/invariance property of the protocol
- Liveness check for checking effectiveness

Formal Aspects of Institutions and Organisations

Grossi et al.

- Development of logical frameworks for grounding the specification of multi-agent institutions and organisations
  - Formal analysis of constitutive norms (counts-as)
  - Formal analysis of the notion of *social structure* within groups of agents
- Use of modal logics and description logic

Agent Applications

Applications: our projects

- ANITA "Administrative Normative Information Transaction Agents" (ANITA) with UM, RUG and UL
- STW project "Distributed Model-Based Diagnosis and Repair" with TUD, NLR and UM
- PhD project together with the company Emotional Brain in Almere and RUN on multi-agent expert systems
- BSIK/ICIS project “Adaptive Support Systems” with TNO
- Intelligent companions (with DECIS, RUG, Philips, Berchet)
- Ontologies for MAS (with Information Science & IBM)
- Agent programming for mobile devices (with Melbourne)
- Adaptive Support Systems (with TNO)
- AIBO Soccer (with DECIS, TUD, UvA)
- Virtual Characters in Games (with GIVE group and TNO)

(Multi) Robot Systems

- Traffic & transport
- Space robots
- Rescue robots
- Robot soccer
- Robot companions
- …

NASA explorer robots

Autonomous vehicles

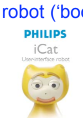
Autonomous Unmanned Aerial Vehicle - Linköping

Robot soccer

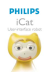
AIBO programming

Intelligent Robot Companions

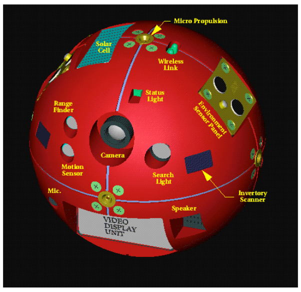
- Companions of human users
  - Personal assistants
    - PSA's for ISS (NASA)
    - Intelligent user interface (Philips)
  - Playmates / Mentors
    - Toy robot ('boon companion', Berchet)



Philips iCat



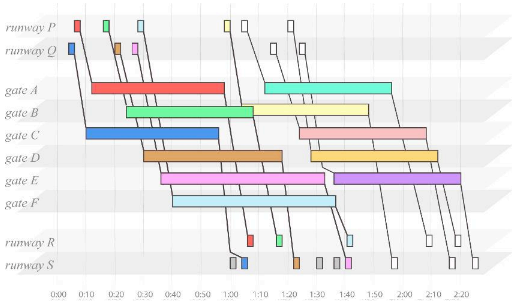
NASA's Personal Satellite Assistant (PSA)



'Boon companion' project

- Aim: devising an intelligent companion
  - toy robot companion (Berchet)
  - intelligent interface (iCat, Philips)
- UU part(s) in the project:
  - Reasoning ('deliberation') module
    - Personal reasoning (BDI, emotions, perceptions)
    - Social reasoning (roles, norms, obligations, interaction)
  - Communication / dialogue module
- Integrated use of cognitive / BDI model(s) (extended with emotions), agent programming, learning techniques, NLP

Airport Traffic Plan Repair



Mechanism Design for Airport Traffic Planning

G. Jonker

- Aim: implementing multiagent techniques in the domain of *airport traffic tactical planning*.
  - In the last stage of planning, just before execution, various disrupting events can occur that disrupt the planning at an airport.
  - Determining the best way to solve these disruptions is a typical multiagent resource allocation problem, in the sense that it has to satisfy certain criteria: efficiency, fairness, incentive-compatibility.

Multi-agent expert systems

Lebbink et al.

- Motivation:
  - Expert system for multi-disciplinary domains
    - Case: company Emotional Brain wants an expert system for group (dys)functioning diagnosis, based on diverse medical, psychological, sociological data and background knowledge
  - Idea: use a *multi-agent system* that can coordinate / negotiate the opinions of the various agents (experts) and make a decision

Multi-agent decision-making

- Formalism to prescribe the activity of making decisions by agents and activity of communication between agents.
- Sets of rules define when agents may make decisions and when to utter communicative acts; in addition, these rules define how the agent's cognitive state is updated afterwards.
  - Sets of rules make up either *decision games* or *dialogue games*.

Adaptive Support Systems

Universiteit Utrecht

Bob van der Vecht, André Meyer, Martijn Neef

Universiteit Utrecht

Research Questions

- What are the advantages of the integration of humans and agents in collaborative systems?
- What socio-technological challenges must be met to achieve support systems that are capable of altering their role or function in an organisation according to changing operational demands?

Goals

- Develop a methodology for designing collaborative systems in human-agent environments
- Develop simulations that demonstrate dynamic, collaborative behaviour in human-agent teams

Embedding Artificial Agents in Human Teams

Topics

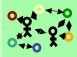
- Adaptation in human–agent teams
- Coordination strategies

Questions

- Implications and opportunities of embedding artificial agents in human teams
- Mechanisms and procedures to harmonize a hybrid team
- Types of human–agent teams from a technological and cognitive stance

Approach

- Interdisciplinary literature study and concept development
- Evaluation in demonstrator setting



What is Adaptation?

- Alteration in structure or habits
- Adjustment to environmental conditions and internal demands

What is a Support System?

- Network of facilities and people who interact for mutual assistance
- Task support
- Team support

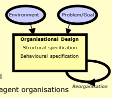
Dynamic Organisations

Topics

- Hybrid organisations
- Autonomous reorganisation

Questions

- Integration of human and artificial agents into one organisational model
- Organisational structures in human-agent organisations
- Relations between organisational goals and structures
- Adaptation of an organisation to environmental changes



Multi-Agent Decision Making Architecture and Methodology

Topics

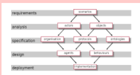
- Collaborative reasoning
- Distributed semantics

Questions

- Trust, commitments, sanctions
- Models for goals, plans and tasks
- Conflicting goals and common plans
- Hierarchy vs. community (p2p)
- Collaboration vs. competition

Approach

- Experimental refinement
- Spyse, Diamond



Virtual characters in games

with UU gaming and HCI group

- Aim: adding 'intelligence' to games
- Special issues
  - Believable/natural behaviour
  - Agents vs avatars for controlling virtual characters
    - 'human in the loop' vs autonomous agents
  - Cognitive modelling challenges
    - Moods
    - Sensing the environment
    - Communication
    - Group dynamics and social behaviour, roles
    - Intention recognition / behaviour prediction

Conclusion

- Over the last decade we have been engaged in:
  - Logic of agency / agents
  - Agent-oriented programming
    - Development of programming language 3APL
    - Agent-oriented software engineering
  - Applications
    - Ontologies for MAS (with IBM)
    - Multi-agent expert systems (with EB)
    - Airport Traffic Planning (with NLR, TUD and UM)
    - Adaptive Support Systems (with TNO)
    - Virtual Characters in Games (with GIVE group and TNO)
    - Intelligent companions (with DECIS, Philips, Berchet)
    - …

Credits: the UU-IS Group

- Frank Dignum
- Frank de Boer
- Jan Broersen
- Martin Caminada
- Mehdi Dastani
- Henry Prakken
- Gerard Vreeswijk
- Marco Wiering
- Huib Aldewereld
- Susan van den Braak
- Jurriaan van Diggelen
- Davide Grossi
- Geert Jonker
- Fernando Koch
- Henk Jan Lebbink
- Cees Pierik
- Birna van Riemsdijk

Software & More Information

- http://www.cs.uu.nl/3apl/

Thank you for your attention!

(And for your support during the last 10 years!!)

□□□□2006

First Call for Papers

8th International Workshop on Deontic Logic in Computer Science (□EON2006)

Special Topic: Artificial Normative Systems

Utrecht, The Netherlands
July 12-14, 2006

http://www.cs.uu.nl/deon2006/

The biennial DEON workshops are designed to promote cooperation among scholars across disciplines who are interested in deontic logic and its use in computer science. These workshops traditionally support research linking the formal-logical study of normative concepts and normative systems with computer science, artificial intelligence, philosophy, organisation theory and law. In addition to these general themes, DEON2006 will encourage a special focus on the topic

ARTIFICIAL NORMATIVE SYSTEMS.

There have been seven previous DEON workshops: Amsterdam, December 1991; Oslo, January 1994; Sesimbra, January 1996; Bologna, January 1998; Toulouse, January 2000; London, May 2002; Madeira, May 2004. Selected papers from each of these workshops have been published internationally. (See the links on the DEON2006 website.)

The Program Committee invites papers concerned with these

WORKSHOP GENERAL THEMES:

WORKSHOP SPECIFIC THEME ARTIFICIAL NORMATIVE SYSTEMS

DEON2006 has a special focus on deontic notions in the theory, specification and implementation of artificial normative systems, such as electronic institutions, norm-regulated multi-agent systems, and artificial agent societies more generally. We also invite studies of hybrid systems, incorporating both human and computer-based agents. Topics of interest in this special theme include, but are not limited to:

- normative systems
- regulated multi-agent systems
- norms in agent societies
- normative agents
- (electronic) institutions
- virtual organisations
- contracts, laws, regulations in artificial societies
- norm violation and punishment
- norm verification

Timetable:

- 27 January 2006: Submission deadline
- 24 February 2006: Notification of acceptance
- 24 March 2006: Deadline for final, camera-ready versions.