Abstract

This paper outlines the formulation of a state of the art intelligent multi agent society that exploits, through semantic enhancement, the data available in an existing digital building model (DBM) software architecture (see (Dibley et al., 2009)). Collectively the agents combine a range of specialist ontologies scoped to sub domains to realise individual and collective skills. Each ontology uses an appropriate knowledge representation to maximise simplicity while retaining adequate expressivity. A shared upper ontology is employed to support general communication and to provide ‘common sense’ knowledge. The agents in general are characterised by the strong notion of agency (as described in the literature), using the BDI (belief, desire, intention) model of agency as a powerful abstraction tool that is suited to applications where the complexity of systems and processes cannot be fully modelled using conventional techniques. The goals of the agents are, typically, to maximize some defined utility, and resulting behaviours are individual and collaborative, aligned and opposed. Knowledge retained, evolved and utilised by agents needs only to be generally locally consistent, and collectively the system can contain redundancy. The BDI model hasn’t yet been exploited in construction industry applications; most existing work employs task/plan rather than goal focussed deliberative agents.

Keywords: multi agent system, BDI, ontology

1 Introduction

A multi layered software architecture has been developed to support the knowledge requirements of a range of software tools. The software developed is referred to in this document as the DBM i.e. a particular implementation cf. a universally agreed architecture. The initial target deployment is commercial buildings for the support of assisting facility management (FM) decisions, but the software could be adapted for application in other settings/roles such as domestic buildings for monitoring the health of the elderly in their homes. The DBM incorporates building information models (BIMs), taking a BIM to be a product model i.e. a data schema instance capturing buildings related entities (geometric data, schedules, geographic and material specification etc), and the relationships between them. Besides utilising BIMs, the DBM encompasses an agent and other layers to provide enhanced facilitation to client tools. Specifically these other (lower) layers of the DBM make available both historical and near real time data from a wide range of sensors that is mapped to the models. For the purposes that DBM is required to satisfy, there exists a high number of hierarchal, simultaneous and collaborative roles that utilise a variety of distributed heterogeneous resources. The multi agent paradigm consisting of individual autonomous, goal seeking entities (that assumes one or
more roles) that communicate using a common fairly abstract language is therefore well suited. The
tagging of resources in lower layers of the DBM with semantic descriptions, and equipping agents
with the ability to reason about that semantic information delivers a level of intelligence. The
development of that collection of intelligent and rational agents is the focus of this paper. The BDI
formalism that specialises the agent paradigm has specific properties that can be used to address the
difficulties of FM knowledge sources, namely potentially incomplete, vague and overlapping
knowledge; this formalism has been adopted in the DBM. A discussion of the technical features of the
agent paradigm and specifically that of the BDI formalism is presented first, followed by discussion
of its suitability to application in the DBM. Next an overview of the methodology used is presented
followed by a discussion of the practical implementation, and an outline of further work. As the paper
aims to adequately cover the aspects of intelligent agency that are novelty utilised in the context of
buildings, lack of space prevents detailed discussion of application details. It is hoped to publish those
details in a journal paper soon.

The multi agent paradigm has been used in AEC industry research but its application has been
limited to the use of the weaker notion characteristics, specifically: autonomy, proactively and
reactivity. No exploitation of the strong notions of agency (see §1.1 for technical features) has been
reported. The weaker notion of agency however matches well with AEC industry characteristics, with
reported applications facilitating decentralised control, authority and information, distributed parallel
activity and supporting the interaction of personnel in virtual organisations. Specifically general MAS
protocols directly support negation and contracting. Applications of MAS in AEC are in engineering
design, management of supply chains, project scheduling and control and concurrent engineering
(Ren & Anumba, 2004).

Rational behaviour in the context of agents is that behaviour consistent with the agent advancing
towards its goals in an optimum way, consistent with its generally constrained knowledge about the
world. An agent’s knowledge is realised by its internal state, held typically symbolically as facts in a
belief repository and by access to other resources such as ontologies. A degree of intelligent
behaviour is exhibited here when the agent draws on mechanisms such as reasoning with its
knowledge, but intelligence can obviously be realised in many other ways. The approach to realise
intelligence here and the focus of the discussion, is the type of system based on rational individuals, in
contrast to so called synthetic ecosystems (“inspired by social behaviour in non-humans, often
insects”) that lack, or place less emphasis on individual rationality, but exhibit “social coherence”
(Parunak et al., 1997).

Central to the requirement of the agent layer is the provision for cooperative working of agents
with specialist skills; these agents take different perspectives on the buildings and related domains,
typically utilising dedicated ontologies and integrating their contributions with the aim of delivering
enhanced functionality. Ontologies used by agents include an IFC (IAI, 2008) derived ontology
describing buildings, materials, process and assets, an engineering ontology and a sensors ontology.
Typically, FM tools interact directly with a database; here the MAS layer resides above, and uses the
services provided by the information layer which incorporates database functionality supporting
entities such as sensor nodes and document repositories.

1.1 Technologies to Support Rational and Intelligent Agents

Many attributes can be used to categorise an agent entity and a useful grouping of characteristics is
that suggested by Wooldridge and Jennings (1995), namely that of weak and strong notion of agency.
The former includes autonomy, perception and appropriate reaction to the environment and an ability
to communicate using a common high level language. The strong notion of agency uses human like
mentalistic attributes (propositional and intentional attitudes) such as belief, desire and intention as an
abstraction and/or other properties e.g. (Shoham, 1993), to model the behaviour of complex systems,
typically where internal mechanisms are not well understood or not easily captured using
conventional techniques. In contrast to the intentional ‘stance’ other stances are, for example, the physical where the laws of physics simply explain behaviour and, the design, where understanding of purpose is adequate (Wooldridge, 2009). The nature of the domain of the DBM, namely a complex combination of systems (building and plant), and influences (environment, and people interacting with that environment), that can only be partially observed by a limited number of sensors, fits well with the functional benefits of the BDI model, a (popular) formalisation of the intentional stance.

Although the selection of specific mentalistic or other attributes to realise an intentional stance/model is contentious (Rao & Georgeff, 1995), the use of belief, desire and intention (BDI) is the most widely adopted for reasons of its basis on “a respectable philosophical model of human practical reasoning” (Georgeff et al., 1998). Belief sets capture an abstract domain based perspective of the world that in practical terms is limited and incomplete. Goals, the realisation of desire, capture purpose, the motivation for a certain behaviour and strategy, while intentions are some future “.... state of affairs that an agent has chosen and committed to .... that tend to lead to action” (Wooldridge, 2009), embodied by plans (static from a so called plan library or dynamically evaluated, possibly ‘from scratch’). The goals, desires and plans capture the motivational, informational and deliberative attitudes respectively of the agent.

For an intelligent agent, beliefs play a role in deliberation as well as utilisation by plans, constituting assumptions that are “... part of the cognitive background” (Bratman, 1992). In this setting, beliefs play a more fundamental role and have a wider scope than variables used in algorithms, so semantic expression, grounded in an upper (‘common sense’) ontology is helpful. In making decisions an agent may need to take significant information “for granted”. Moreover, Bratman (1992) raises the issue of how the context of deliberation should account for a “degree of acceptance” of beliefs; taking a fact for granted in a certain context for example may be acceptable in one scenario but not in another. These fundamental abilities and multiple roles of beliefs equip the agent with an improved ability to deal with vague and incomplete knowledge. A number of assumptions are expected for rational behaviour in the relationship between beliefs and intentions, relating to consistency and completeness. For example holding an intention to reach a given state of affairs while believing that that state of affairs won’t be reached is not acceptable, while it is acceptable to believe that an intention to bring about a fact doesn’t necessarily lead to belief in that fact (Wooldridge, 2009).

Within agent systems where emphasis is placed on self interest, complimenting rational and autonomous behaviour, ‘social norms’ and some level of benevolence should be observed to preserve the integrity of the system while enabling cooperation for the solution of such reliant problems. Social norms are inherent in goals and plans so holistic conformance is easily captured in a closed system. Benevolence, typically during participation in cooperative behaviour, can be tolerated to the extent it is not inconsistent with the agent’s goals or consume ‘significant’ resources. The latter is always finite so some balance should be devised, which is easier in a closed system designed for a specific purpose.

Central to MASs is the ability to communicate using common abstract semantics and (typically asynchronous) protocols, in contrast to (typically synchronous) ad-hoc method invocations in object based languages where the caller has knowledge of the callee’s interface, (and the callee has no control of the invocation). FIPA defines the standard agent communication language FIPA-ACL with formal syntax and semantics. The semantics are based on speech act theory; the acts bring about a change in the state of affairs of those involved. The utterance brings about the so called rational effect of the message (perlocution) e.g. inform to (possibly) bring about a change in beliefs, and request to (possibly) change goals. The other structural message element is the pre condition. The fundamental primitives (performatives) in the standard are inform and request, from which all the other 20 performatives are derived. The semantics are expressed using the modal operators belief and uncertain from the SL (Semantic Language). Consistent with that previously stated, the defined semantics state that the rational effect can be ignored by the callee. Regarding content, the
specification allows the statement of such metadata as the reference ontology; of course that ontology
should be shared between the agents involved in the messaged exchange.

The formalisation of BDI implementation/s is not constrained by its theoretical formularisation,
and in fact it has been stated that the latter is of “little relevance” to implementation of systems due to
incomplete axiomisation and of efficient computation mechanisms (Rao & Georgeff, 1995). Of course
theoretical formalisation offers the advantage of proving appropriate (to the formal specification)
system behaviour through formal proof.

1.2 The Suitability of the BDI Model to the DBM

Traditional algorithmic systems perform well with static knowledge that is completely defined
within that context. However the nature of the building model environment is the opposite to that: it
has high complexity in terms of information resources and the entities described therein, some of
which may be modelled in several contradictory ways, it has an incomplete ‘view of the world’
through limited sensing ability and potentially missing model detail, and the environment is
constantly changing. The characteristics of the BDI abstraction of agency makes it particularly suited
to the BDM environment by virtue of the use of desires to extract strategy, compared to the
algorithmic approach where that strategy is ‘embedded’ in the algorithm at design time. So the BDI
system is better able to adapt to changes in the environment by knowing its direction (perhaps
selecting a different plan to reach an explicit goal) and is able to take a higher level view of data
(beliefs) that can be incomplete, by deriving context from its intentions and desires, assisted with the
provision of ‘common sense’ derived from the core SUMO (Pease, 2008) ontology and reasoner
facility. In summary the BDI abstraction utilised in the DBM environment adds the potential for
increased robustness through flexibility gained by a sense of the agents’ direction and purpose.

2 Methodology

Many MAS ‘implementation’ methodologies have been proposed with various characteristics such as
their coverage of lifecycle, level of detail, similarity to conventional software engineering, availability
of tool support and provision or recommendations of notation. In contrast, methodologies for formal
agency models differ due to their generally dissimilar nature, for example in some formal MAS
models, the specification expressed in a logical representation can be used directly during execution.
In these systems, agents are theorem provers, where goals and beliefs etc. are derived from the logical
representation of the specification, so no refinement of that, as is seen in the analysis and design
phases in traditional software engineering, is needed. Historically formal approaches have evolved
separately and without clear definition as to assisting the implementation, or aspects of non-formal
systems. The methodology developed for the DBM agent layer is non-formal in this (mathematical
basis) sense, in line with other well known MAS ‘implementation’ methodologies, for reasons of ease
of practical realisation. The main motivation for developing a custom methodology was to eliminate
unnecessary detail when applied within the constraints of the DBM application, apply the simplest
mechanisms, support appropriate concepts (goal, plan, etc.) as well as preferred process artefacts at
the appropriate phase, enable the use of UML diagramming and supporting tools, and to directly
support the BDI formalism in the analysis phase as well as in implementation and to map to the
chosen framework (Jadex - see §3.1).

Lack of space prevents a presentation of the methodology, but in summary, its development was
influenced by several existing methodologies. The methodology proposed by Nikras (Nikraz et al.,
2006) is attractive due to its compactness in the analysis phase. The goal hierarchy inspired by MaSE
(Wood & DeLoach, 2000) was integrated, as was pattern application inspired by Tropos (Bresciani et
al., 2004). Some simplifications are brought about by the assumption that goals are not shared, and
that all agents conform to social norms, in pursuit of their goals, that do not compromise the integrity
of the MAS. Additionally links to aspects of ontology development methodology NeOn (Cea et al., 2008), were established to ensure that adequate semantic support is provided for the agents’ micro architecture and message content. At the implementation phase, the methodology specifies iterative refactoring with respect to test case performance logs in order to achieve appropriate commitment, by appropriately dividing plans into sub entities, at computationally expensive points in algorithmic execution.

The methodology currently doesn’t address explicitly consistency and completeness checking through the development lifecycle. The establishment of traceability relations similar to those of Cysneiros & Zisman (2008) may be a suitable approach. However the methodology specifies some manual verification of audited logs of test cases. Verification of strict conformance to the various standards such as the semantics for FIPA messaging is not a central concern as the system developed is closed.

3 Realisation

This section describes the implementation decisions made such as the selection of frameworks and libraries, and the rationale for those choices.

3.1 Agent Architecture

The Jade framework (Telecom Italia SpA, 2008) was selected for the reasons of meeting the functional requirement of delivering an agent infrastructure conforming to a de facto standard (FIPA (IEEE FIPA, 2009)), that is well documented, and is open source software. Additionally a range of works by other authors compliments the framework in several ways, of which the Jadex (Braubach et al., 2003) libraries providing a BDI agent architecture was utilised. Jadex provides a hybrid architecture (deliberative and reactive internal mechanisms driven by events) layer that resides on top of Jade; Jade can be regarded as taking the role of middleware. Reactive behaviour is that characterised by a simple response to conditions excluding any symbolic or practical reasoning. Jadex was found to be suitable due to its scalability and ease of development.

The framework provided by Jade addresses the agent (macro/system) architecture delivering functionality for: agent hosting, lifecycle control, infrastructure services and messaging, but deliberately doesn’t specify the agents’ internal (micro) architecture. The adopted Jadex libraries provide such architecture with a BDI formalisation supporting, unlike other provisions, both goal deliberation (decide what) and means-end reasoning (decide how) together with simple belief base mechanism and query facility. In the DBM the (Jadex) agents’ knowledge representation is currently primitive types, Java class instances, or FIPA-SL0/1 (a family of semantic languages defined by FIPA, see §3.2) sentences. The ontology derived semantic information is captured by a Java objects generated by converting the ontology from its native OWL representation, for easier manipulation in message content and belief bases. The libraries provided by the Jastor project (Szekely & Betz, 2009), are used to generate Java class representation (Java Beans) for agents’ semantic information representation from Protégé authored ontologies. Therefore deliberation phases and pre conditions, event, goal and plan parameters for example that rely on the additional semantic constructs can be accommodated in native Jadex agent architecture without modification of the internal rule engine.

3.2 Messaging

Regarding messaging, the DBM implementation makes a separation between the semantics of the speech acts and that of the content. The content type varies but is tagged with reference ontologies (typically SUMO but additional ontologies are often appended) as well as other metadata. The messaging semantics are currently embodied by rules implemented as Java code in the agents. The
framework Jade Semantic Add-in (JSA) (Bellifemine et al., 2007) was considered for the purpose of message interpretation but its integration is currently postponed as mapping between OWL and FIPA-SL would need to be addressed if it were to be integrated completely, specifically relating to the area of agents’ belief bases. FIPA communicative acts are formally defined in FIPA-SL2 (see below) and subsets using a belief and intention model (Poslad, 2006); JSA implements an interpretation engine that infers messaging intent form potentially alternate message composition, so adds great flexibility to agents’ dialog. However the DBM is a closed system and messages formed by agents, while conforming to the FIPA semantics, are only created in fixed ways so the extra flexibility is not required i.e. message receivers only receive known message formulations so don’t need to perform interpretation at that level.

Relating to message content in inter-agent dialogs, currently expression uses the OWL notation, captured as (type safe) Java classes. There are a number of tools that perform mapping of OWL to class based representation at design time and while there is some limited loss in expressivity, techniques exist that minimise the consequences. The losses are due to the inherent difference in semantics mainly in the areas of satisfiability and completeness (Kalyanpur et al., 2004) which can’t be transformed. The chosen tool was Jastor (as mentioned above) that has the additional ability to generate Java mappings at run time, thereby matching the flexibility that would be gained by using a communication based on OWL ontology entities.

Regarding encoding the message content expressions, FIPA defines FIPA-SL syntax and semantics that can be used as such. FIPA-SL, as part of its specification includes a content reference model defining the entities: predicate, concept, agent action (all instances of which are defined with ontologies in the DBM), entity list which is able to include any of the other entities, and identifying referential expression (IRE) which essentially is a selection expression that specifies constraints. The content language specification enables the expression of appropriate content with respect to the performative e.g. a message entailing a query uses an IRE, inform uses a proposition etc. Jade conveniently provides a library to parse such expressions. The FIPA-SL is actually a family of languages (as mentioned above): SL0 and SL1 cover first order logic (FOL) constructs, SL2 adds action operator feasible (done is defined in SL0), and the belief modal operators belief and uncertain with some restrictions to retain decidability in the modal system (FIPA, 2003). Currently the DBM uses SL0/1 but the increased expressivity of the modal operators are expected to play a significant role in enriching the collaborative problem solving (see §4 below). Finally regarding encoding of message content Jade again provides so called codecs for that purpose, which have advantages over Java and XML serialisation (ease of implementation, human readable and platform independence).

3.3 Deployment

The software is currently deployed in an office using wired sensors (outlined in (Dibley et al., 2009)). In addition a wireless ZigBee (ZigBee Alliance, 2009) based sensor network including the same set of sensors as formerly referenced, with the addition of lux level sensing is being configured, for deployment in a large multipurpose student working/meeting area. Regarding evaluation of the implementation as an MAS, Wooldridge (2009) cites from the literature coherence and coordination as factors for consideration. Assessment of the deployment will aim to quantify the effectiveness of the DBM in terms of data enhancement, and well as an MAS, and those findings will be reported in due course.

4 Further work

The work to follow will target increasing the so called intelligence of agents in the DBM and well as improving the collaborative problem solving capability. Increasing intelligence will focus on adding provision for agents to reason about their own, and other agents’ mental attitudes and possibly about
their deductive ability (Russell & Norvig, 2003). Initially beliefs will be targeted; an agent that is able to carry out such reasoning (typically for means-end) is better able to utilise its beliefs than one that is not e.g. to identify missing information and to potentially discover it. The realisation of equivalent modal system KD45 reasoning using Prolog as presented in (Binas & Ioerger, 2004) seems attractive as a practical implementation. While the goal deliberation provision with meta-goals is currently being used to good effect to activate and deactivate goals, that provision enhanced with the introduction of reasoning about mentalistic attitudes will also contribute towards the improved effectiveness of the agent layer. Improving collaborative problem solving will rely on improvements in the ways agents initially decompose problems, and on increased expressivity gained through FIPA-SL2 based dialog with the use of, for example, the uncertainty modality and action operator feasible.

Another main area to be pursued is that of capturing problems in FIPA-SL2 sentences and attempting a proof with the assistance of other agents. Typically an agent will instantiate a hypothesis from library templates, after reasoning primarily based on the SUMO ontology. Delegate agents will be dynamically created to try to collect evidence for each hypothesis. A blackboard pattern based sequence can be used to coordinate cooperation.

Agent learning has the potential to improve the performance of the DBM agent layer generally, and for example to realise minimal or zero configuration, and will be investigated. Learning is realised when “… the agent observes its interactions with the world and its own decision making process” (Russell & Norvig, 2003). The use of a performance measure allows estimation of utility and plays a central role in learning.

5 Conclusions

A layer of software hosting collaborating, rational, multi skilled agents has been implemented with the objective of delivering practical benefits to FM systems by enhancing the reuse of data available from sensors, both historical (in databases) and in near real time, as well as from other information sources including building plans and schedules. The BDI formalisation is a useful abstraction mechanism that allows modelling of, and to predict the behaviours of complex systems; it is difficult to imagine how such additional functionality could be delivered without some form of abstraction. The technical features and design considerations of the BDI formalism has been outlined and the application to the DBM agent layer has been described, as well as justification for its use. Compared to the weaker notion of agency, particularly when complemented with semantic knowledge and the ability to reason, it has been shown how the BDI formalisation is better able to deal with uncertainties and incomplete knowledge that characterise FM applications. In the DBM the emphasis is on agents collectively providing a good, widely scoped understanding of the changing environment as opposed to very detailed single technology based numerical model approaches. Numerical tools used for specific detail engineering design, while highly suited for that purpose, have associated difficulties if attempting to apply them in this context. Those issues relate to configuration, reliance on expert users, software interfacing and cost implications of licensing. Similarly the use of simple, cheap sensors and intelligence to observe the environment in contrast to linking/integration of embedded control systems in plant such as HVAC units provides a more cost effective solution. Only basic configuration of the system is needed, as incorporated in agent behaviour is a simple form of learning through the development of beliefs by the agents.

Equivalent functionality to at least to weaker notion of agency, like other scenarios in software engineering could be reached by other more traditional means instead of the use of the agent paradigm e.g. using general purpose middleware solutions targeting distributed functionality and threading, or even with the application of frameworks like OGSi, but the BDI formalism offers powerful additional features as has been described. The use of BDI in a construction industry setting seems fairly novel
although the features of autonomy and reactivity in a setting of non static heterogeneous environments have been employed.

References


POSLAD, S., 2006. Review of FIPA specifications. FIPA.


