Multiagent System for Detecting and Solving Design-time Conflicts in Civil Infrastructure

Jaume Domínguez Faus[1], Francisco Grimaldo[2], Fernando Barber[2]

Abstract One typical source of problems in the Civil Infrastructure domain is the distributed and collaborative nature of the projects in which different profiles of engineers contribute with designs devoted to the interest of their field of expertise. Thus, situations in which there are different conflicts of interests are quite common. A conflict refers to a situation in which the actions of an engineer collide with the interests of other engineers. In this paper, we present a multi-agent system that, thanks to the use of ontologies and rules on those ontologies, is able to detect profile-specific conflict situations and solve them according to the preferences of the parties involved in the conflict. The conflict solving is based on the Multi Agent Resource Allocation (MARA) theory. The system is applied to a real use case of an urban development where both the road network and the buildings are designed.

1 Introduction and Related work

Interoperability is an often addressed term when enumerating the problems of distributed systems. In the same way that communication becomes difficult between two people speaking different languages, communication is difficult when dealing with systems relying on data for modeling a problem to solve. This happens because data only describe things and, as any description, they can be interpreted in many ways. Any infrastructure project as, for instance, a road construction involves lots of disciplines ranging from land-use to security regulations, with noise emission, road tracing, water drainage and many others in between.

Jaume Domínguez Faus
Centre for 3D GeoInformation, Aalborg University. 9220 Aalborg, Denmark +45 9940 3699
e-mail: jaume@land.aau.dk
Francisco Grimaldo, Fernando Barber
Departament d'Informàtica, Universitat de València, Av. de la Universitat s/n, (Burjassot) València, Spain 46100
e-mail: francisco.grimaldo@uv.es, fernando.barber@uv.es

The general tendency is to have -when possible- a data model for each discipline that is used by specific software packages to assist the daily engineer's life and split the project works in discipline-experts teams. Problems arise when all the works done by different teams have to be put together. Issues like design clashes, synchronization problems, exceeded budgets, conflicts of interests appear, etc. as a consequence of the decentralized way of working with heterogeneous data models. The detection and solving of such problems is still a prominent manual work and some of them might remain undetected when this process is finished. Once the construction starts, the consequences of mistakes or suboptimal design cause that the infrastructure cost increases a 5-10% of the total budget in average [4]. Most of the efforts done so far have focused on avoiding the collisions by improving interoperability among different data models. It has not been, however, until recently when the conflict-solving has gained attention. This paper presents a new multiagent-based approach for detecting and solving design-time conflicts in the Civil Infrastructure domain. Currently, the Civil Infrastructure software industry focuses on making models that integrate more and more aspects of design disciplines in order to increase interoperability. Perhaps, the most advanced results of these efforts are the most successful standard files (such as CityGML or IFC [6], and AutoCAD's DWG) or the Building Information Model (BIM) servers [4] which combine CAD models with management spreadsheets and other other documents to provide an integral project life-cycle management. However, this distributed and collaborative work has to deal with conflicts that inevitably appear when sub-designs of a project are merged.

Multi-agent Systems (MAS) have been suggested to aid in Civil Infrastructure projects. It is possible to find examples of MAS focused on controlling machinery [9], or on the distinct phases of a project: the tendering procedure [10]; the material supply chain [11]; and the construction phase [8] and [13]. Nevertheless, to the authors' knowledge there is a gap that has not yet been considered satisfactory: the negotiation between designer expertises in the Design phase of the project. Even though it is possible to find some problem-specific works like [1], the situated nature of this collaborative work makes the problem of abstraction of a system to be wicked [5]. However, this abstraction is necessary to capture the negotiation as a design conflict-solver in the software packages normally used by the engineers in their daily work. Thus, more research is needed in this field for MAS to be a real option.

The use of ontologies has been proposed as a means to give sense and semantics to the data in several contexts. In geospatial and civil infrastructure information, ontologies are not widely used. Although it is possible to envisage ontological structures in some data models (e.g. CityGML) they are hardly used in a formal and explicit manner. We propose the use of ontologies to support automatic conflict detection and of the Multi Agent Resource Allocation (MARA) [3] for its solving at a semantic level. In section 2.1 we present the ontological approach we propose to represent the world semantics, and the rules that are used to detect conflicts. Further below, in section 2.2 the negotiation mechanism used to solve conflicts is introduced. Finally, in section 3 we describe a use case in which the system was applied in order to illustrate its usefulness.

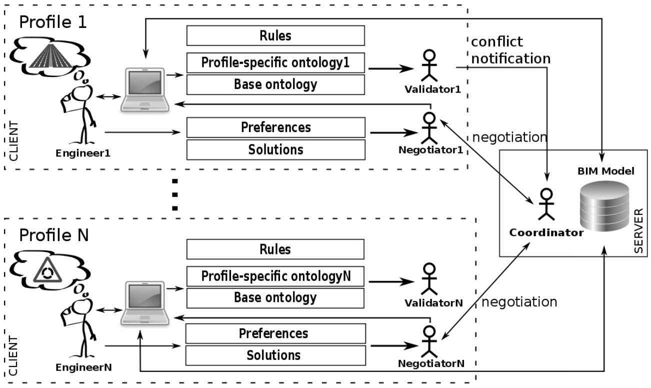

Fig. 1 Overview of the system

2 Architecture of the System

The multi-agent system we propose for detecting and solving design-time conflicts in the Civil Infrastructure domain is depicted in figure 1. It follows a distributed architecture approach allowing the engineers of different profiles to design, through their client interfaces, a common BIM model that is stored at the server. This collaborative work is carried out with the assistance of a set of agents: the Validators, the Negotiators and the Coordinator. The Validator agents are in charge of semantically detecting conflicts and errors within the model by using the ontological knowledge of each field of expertise. In turn, Negotiator agents aim at solving conflicts by expressing the preferences of the engineers in a negotiation protocol that is initiated by the Coordinator under conflict notification. Following, we review the details about these agents, which have been implemented as part of an agent society in JADE [2].

2.1 Semantic Conflict Detection

We propose using OWL [12] ontologies for the semantic abstraction of the data beyond the pure classical attribute/value pair. As shown in figure 1, our ontologies are structured in layers in which each layer provides an extra level of abstraction. At the lowest level, the Base Ontology defines the basic concepts needed by any geospatial data model. The base class `Feature` refers to the most basic object that traditionally forms geospatial data models such as GIS or CAD systems. A `Feature` is composed of a `Geometry` and of a set of `Attributes` defined by its name, its type, and its value. `Features` can be related to each other through the generic relation `hasRelationship` and its inverse relation `isRelationshipOf`. This pattern has proven to be flexible and suitable for many uses. Besides, by using inheritance, classes can be arranged in a hierarchy (e.g. `Conflict` and `Error` are particular types of `Problems`). Therefore, this ontology acts as the first layer of abstraction allowing the creation of Profile-specific ontologies on top of it.

Profile-specific Ontologies are meant to define the concepts of interest for each profile. These concepts can be specific `Features` providing particular properties and/or semantic meaning (e.g. a `Building` or a `Parcel`) and also specific relationships defining how certain `Features` relate to each other (e.g. `isLocatedAt` relates a `Building` with the `Parcel` where it is placed). This second level of ontologies allows to separate the categorization of the different interests involved in civil infrastructure projects in order to ease the management of the knowledge. Note, however, that this does not necessarily prevent a concept to be shared among different profiles in case several profiles need it.

As the project progresses, the different engineers include new designs or edit the existing ones and the model changes continuously. In this dynamic context, the Validator agent automatically assists in the correctness of the model as a whole by periodically checking a set of rules defined for the profile. We propose using SWRL [7] rules as a way of supporting the semantic consistence and ontological reasoning. These rules are ontological expressions with an antecedent and a consequent that allows the Validator agent to detect and infer problematic situations. The problems found are categorized between `Errors` or `Conflicts`, acording to the classes defined in the Base Ontology. `Errors` are situations in which the model is not correct due to missing or wrong values in the `Feature`'s properties and, thus, they are notified and solved manually by the engineer through its client interface. On the other hand, `Conflicts` capture the situations where the designers' interests collide and they are solved through the negotiation protocol explained next.

2.2 Conflict Solving Protocol

We propose using a Multi-Agent Resource Allocation[3] (MARA) approach to analyze the possible alternatives that solve the `Conflicts`. The MARA model provides agents with a general mechanism to make socially acceptable decisions. In this kind of decisions, members are required to express their preferences with regard to the different solutions that have been previously proposed by all the members for a specific decision problem. Our MARA approach uses ContractNet-like protocol as the allocation procedure. Figure 2 depicts the process for the case of a `Conflict` between two profiles. When the `Conflict` is detected by one Validator agent, it is notified to the Coordinator agent. Then the Coordinator distributes the `Conflict` to all the Negotiators in a Call For Proposals. The negotiators respond with their alternatives, if any, and the Coordinator collects all the proposals. In the collection, invalid or repeated solutions are filtered out and the set of remaining solutions is distributed again to request the preferences. Each Negotiator then expresses its utility on each of the solutions at hand by giving it a value ranging from 0 (lowest) to 10 (highest). The Coordinator agent then picks the winner solution which is the one that maximizes the global utilitarian social welfare represented by the solution that accumulates highest utility among the negotiators. Finally, the winner solution is then broadcasted to all the clients.

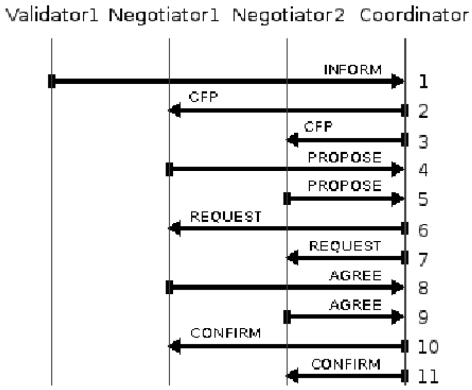

Fig. 2 JADE[2] console showing the Conflict Solving protocol

3 Urban Development Use Case

In order to simulate the daily work of engineers in the design phase, an urban development use case was selected. This project consists of the development of the Strømsø area in the city of Drammen, Norway. Traditionally an industrial area, after decades of growth, Strømsø became the downtown of Drammen while keeping the original industrial aspect. The authorities want to adapt it to the new residential reality. In general, the development goal is the construction of residential buildings to increase the number of inhabitants. In the initial phase, the project defines where to place buildings according to their characteristics (number of residents, floors and footprint). Further phases of the design deal with other more detailed aspects. We focused on the building placing problem to show our MAS approach.

To avoid future traffic jams, it was agreed that there should not be more residents than the capacity of the road. Thus, the location of a building is constrained to the capacity of the road that serves the building. The current usage of the road is obtained by the sum of the inhabitants of the buildings that are associated to that road. So, in addition to their geometry, buildings and roads specify the amount of inhabitants and the road capacity respectively in their attributes. There are two engineering profiles identified: 1) The designer that places buildings in a location of her/his choice (Building profile), and 2) The road designer that detects which road is connecting the building to the road network and checks whether the road is capable to hold all the buildings connected to it (Road profile).

For each profile there is a designer that is developing the model, and each designer has: 1) a validator agent that checks the model according to the semantics (expressed by her/his ontology and rule-set settings) and initiates the negotiation; and 2) a negotiator agent that performs the negotiation on behalf of the engineer.

As introduced in section 2.1, `Feature`s constitute the most generic object that can be defined in our ontological model (unlike pure geometry-based models in which basically only the geometry is known). On top of it, we identify three specific concepts of interest for this use case: the `Road`, the `Parcel` and the `Building`. `Parcels` are `Features` that define an area in which `Buildings` can be placed. `Buildings` are `Features` representing the residential entities where people live in. In turn, `Roads` are `Features` representing the parts of the road network. Beyond specifying the type of a `Feature`, these classes define more attributes that are required to describe their characteristics such as the capacity of a `Road` or the inhabitants of a `Building`. Since these concepts are relevant for both profiles, the previous classes are defined in both Building and Road Profile-specific Ontologies.

Since what a particular type of `Feature` means depends on the profile that is looking at it and, in turn, it is expressed through the relationships that it establishes with other `Features`, each Profile-specific Ontology defines a particular set of relationships the profile is interested in. The layered design of the ontologies allowed the Road profile to define the relationship called `roadServesTo` and its inverse `isServedByRoad` which state what `Road` serves any other `Feature` and vice versa. On the other hand, the Building profile defines the relationship `holds` and its inverse `isLocatedAt` which establishes which `Feature(s)` a given `Parcel` holds and, inversely, where a particular `Feature` is located.

SWRL rules have been defined for each profile so that the corresponding Validator agent can detect the errors and conflicts that appear in the model and that are related to its field of expertise. Regarding the errors, for example, each `Road` must specify its capacity in order to check if it can hold the potential traffic. If a `Road` is missing this attribute, then the model is not complete and the validator agent infers an `Error`. Equation 1 shows the rule used to infer a `RoadCapacityError`, a specific type of `Error` defined in the Profile-specific Ontology for the Road profile. This rule could be read as: if an element *r* happens to be a `Road`, and the result of the operation *isMissingAttribute* for this road and the attribute name "capacity" resolves to true, then *r* is also a `RoadCapacityError`. The operation *isMissingAttribute* is an example of how it is possible to extend the general logic operations of ontologies with user-defined operations. This mechanism is allowed in SWRL rules by means of the use of Built-ins. Similar rules were used by the Building profile to detect when a building does not declare the amount of inhabitants.

On the other hand, the two profiles involved in this use case may also come into conflict. That is the case when the building designer places a building in a parcel where the road connecting to that parcel cannot hold the new population of the building. Equation 2 shows the rule defined in the Building profile to detect this kind of `RoadExhaustedConflict`. This rule is actually a compound rule that: retrieves the `Parcel` *p* where the `Building` *b* is placed, gets the `Road` *r* serving that parcel and computes whether the road is overloaded with the buildings that are connected to it through the operation *isRoadExhausted*. This rule leans on the inference done by another rule about which road serves a parcel (equation 3). This latter rule explores the relationships of the ontology to get the `Geometry` *gp* of the `Parcel` *p* and selects the closest `Road` *r* by means of the operation *closestRoad*.

$$Road(?r) \wedge isMissingAttribute(?r, "capacity") \rightarrow RoadCapacityError(?r) \quad (1)$$

$$isLocatedAt(?b, ?p) \wedge isServedByRoad(?p, ?r) \wedge Road(?r) \wedge isRoadExhausted(?r) \rightarrow RoadExhaustedConflict(?r, ?b) \quad (2)$$

$$Parcel(?p) \wedge hasGeometry(?p, ?pg) \wedge closestRoad(?pg, ?r) \rightarrow isServedByRoad(?p, ?r) \quad (3)$$

The Conflict going to be solved is detected by the Validator agent (see figure 1) who provokes the initiation of the conflict solving protocol described above and depicted in 2. When the Coordinator is notified, he broadcasts the Call For Proposals to the Negotiators acting as proxies of the engineers. The engineers at the clients receive a message informing that a new Conflict solving sequence has been started and the Coordinator is waiting for their proposals. Knowing the details of the conflict (i.e. the Road is exhausted) they think on how to fix it. Different solutions like, e.g., enlarging the Road for more capacity; or reducing the amount of inhabitants of one or several Buildings; or maybe relocating a Building in another Parcel served by a Road with more availability for new residents; are then applied temporarily to the model. A recording system allows to capture the changes to the model. The engineers encapsulate sequences of changes (such as "on Building number 32, set the value of the Attribute 'inhabitants' to 30 from 40") into a Solution and provide all the alternative Solutions they have. All the alternatives are then proposed to the Coordinator who evaluates them and discards repeated or invalid ones. The viable Solutions are then sent back to the clients so the engineers express their preferences on each of them by grading each with a value ranging from 0 to 10. The grades are then sent back to the Coordinator who takes the winner solution as described above. The winner solution is then notified to all the clients and applied to the model.

4 Conclusions

In this paper we presented a system designed to support collaborative work in Civil Infrastructure projects that is able to assist in the detection and solving of semantic Errors and Conflicts. These semantic problems, which also involve geometric problems, are so common that they are normally accepted so long they can be in-field detected and corrected. However, this is not always the case and they may eventually lead to project delays and to overheads. Thus, it is important that the models are delivered free of problems as much as possible. A semantically perfect model without problems or ambiguities eases the automation of the tasks, which translates to a more efficient usage of resources. Conflicts are a special case of problem which are especially difficult to solve. Negotiating is the natural mechanism to reach an agreement on how to solve them. Our system provides a structure for this negotiation by means of suggesting alternatives and picking the preferred one among all the parties -the Profiles- involved in the conflict. The preferred alternative is is the one that maximizes de global welfare.

Acknowledgements This work was supported by Norwegian Research Council, Industrial PhD scheme case no: 195940/I40 through Vianova Systems AS, Norway; the Spanish MICINN, Consolider Programme and Plan E funds, as well as European Commission FEDER funds, under Grants CSD2006-00046 and TIN2009-14475-C04-04. It was also partly supported by Development and Planning Department of the Ålborg University (AAU), the Vice-rectorate for Research of the Universitat de València (UV) under grant UV-INV-AE11-40990.62. Authors also want to thank Dr. Erik Kjems from the AAU for his help without which this paper would not be possible.

References

1. C. Anumba, O. Ugwu, L. Newnham, and A. Thorpe. Collaborative design of structures using intelligent agents. *Automation in Construction*, 11:89–103, 2002.
2. F. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-agent Systems with JADE*. John Wiley & Sons Ltd, 2007.
3. Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaitre, N. Maudet, J. Padget, S. Phelps, J. A. Rodriguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.
4. C. Eastman, P. Teicholz, R. Sacks, and K. Liston. *BIM Handbook, a guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors.* John Wiley & Sons, Inc. New Jersey,, 2008.
5. G. A. Fitzpatrick. *The Locales Framework: Understanding and Designing for Cooperative Work*. PhD thesis, University of Queensland, November 1998.
6. T. H. Kolbe, G. Gröger, and L. Plümer. Citygml: Interoperable access to 3d city models. *Geo-information for Disaster Management*, pages 883–899, 2005.
7. M. O'Connor, H. Knublauch, B. G. S.W. Tu, M. Dean, W. Grosso, and M. Musen. Supporting rule system interoperability on the semantic web with swrl. *4th International Semantic Web Conference (ISWC), Galway, Ireland, Springer Verlag*, pages 974–986, 2005.
8. F. Peña-Mora and C.-Y. Wang. Computer-supported collaborative negotiation methodology. *Journal of Computing in Civil Engineering*, pages 64–81, April 1998.
9. Z. Ren and C. Anumba. Multi-agent systems in construction–state of the art and prospects. *Automation in Construction*, 13:421–434, 2004.
10. M. Schnellenbach and H. Denk. An agent-based virtual marketplace for aec-bidding. *Proceedings of the 9th International EG-ICE Workshop Advances in Intelligent Computing in Engineering, Darmstadt, Germany*, pages 40–48, 2002.
11. C. Udeaja and J. Tah. Agent-based material supply chain integration in construction. *Perspectives on Innovation in Architecture, Engineering and Construction, CICE, Loughborough University*, pages 377–388, 2001.
12. W3C OWL Working Group. OWL 2 web ontology language document overview. Technical report, W3C, Oct. 2009.
13. X. Xue, Y. Ji, L. Li, and Q. Shen. Cognition driven framework for improving collaborative working in construction projects: Negotiation perspective. *Journal of Business Economics and Management*, 2010.