

SWARM MODELLING FRAMEWORK TO IMPROVE DESIGN SUPPORT SYSTEMS CAPABILITIES

SUBMITTED: October 2019

REVISED: April 2020

PUBLISHED: August 2020

EDITOR: Esther Obonyo

DOI: [10.36680/j.itcon.2020.023](https://doi.org/10.36680/j.itcon.2020.023)

*Gabriele Novembri, Assistant Professor,
Department of Civil, Constructional and Environmental Engineering – DICEA, Sapienza – University of Rome;
gabriele.novembri@uniroma1.it*

*Francesco Livio Rossini, research fellow,
Department of Civil, Constructional and Environmental Engineering – DICEA, Sapienza – University of Rome;
francesco.rossini@uniroma1.it*

SUMMARY: *The disruptive development of ICT technologies can be a decisive element in bringing the productivity of the construction sector closer to the highest performing ones, such as automotive. We base the proposed research on the improvement of existing tools with artificial intelligence techniques, with the goal of having a self-adaptive model regarding the objectives to achieve. So, we present a general framework based on the Swarm Simulation Modelling approach. Building objects, goals, constraints and design solutions will be represented as a Multi-Agents System able to communicate, interact and integrate over the existing BIM systems, ensuring a reactive and proactive behaviour. Finally, the model presents many interconnected intelligent agents, linked to building object. They can sense external ‘perturbations’ and react by re-organising their structure to satisfy imposed constraints. We can find the near optimal solution via the DCOP approach, autonomously.*

KEYWORDS: *Swarm agent, Agent-Based simulations, BIM, Project Management, Artificial Intelligence, Project Management, Design Support Systems.*

REFERENCE: *Gabriele Novembri, Francesco Livio Rossini (2020). Swarm modelling framework to improve design support systems capabilities. Journal of Information Technology in Construction (ITcon), Vol. 25, pg. 398-415, DOI: [10.36680/j.itcon.2020.023](https://doi.org/10.36680/j.itcon.2020.023)*

COPYRIGHT: © 2020 The author(s). This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



1. INTRODUCTION

Present-day buildings are complex systems that have to comply with an articulated framework of rules, requirements and codes of practice, ever more extensive and detailed (Bertelsen and Sacks, 2007). Complexity and quantity of sub-systems of different disciplines and professional skills often generate difficulties that turn the design process into a highly unpredictable workflow. Inconsistencies and errors affecting design activity lead in many cases to indeterminacy and uncertainty about characteristics, performances and behaviour of the final product (Singh et al., 2011). Quality and reliability of the design often generate a building process characterised by opposing choices, sudden stop-offs, restarts and rethinks often increment time required to get the final product, intensifying the always conflicting relationships between clients and contractors (Lu et al., 2016).

Assessed and esteemed time and costs comparison show the inefficiencies, error propagation and unexpected budget growth affecting the building sector. These aspects are typically better managed in other industrial sectors, like manufacturing (Barbosa et al., 2017). We can often find causes can be often in goals vagueness and uncertainties, considering that, in some circumstances, the final product of the building process results outdated because of the context of rapid changes (Fazli et al., 2014).

Several reasons, can explain this situation such as the slow adoption of e-procurement systems combined with the low level of standardisation and prefabrication, and the often-inadequate workforce skill. This non-exciting picture, according to some influential research, till now continues to generate the constant and impressive decline of construction sector productivity, in contrast with the clear parallel growing demand (Barbosa et al., 2017). Approximately 50% of the errors reported during the construction phases originate from design phases (Bentley and Building Research Establishment, 1981), where the involved stakeholders cannot govern the required information to assure the correct realisation of the design intent, represent one of the key factor to provide adequate quality and productivity levels to the building process (Otey, 2014).

According to (Matthews et al., 2015; Vähä et al., 2013), a real improvement of the effectiveness and efficiency of the building process can be reached in through the evolution of the tools nowadays available. Identifying industrialisation of building processes and prefabrication as a 'hard' component of the building process, it is clear the need of 'soft' innovations in the support and decisional tools available to designers and stakeholders (De Grassi, 1984). Buildings can rightly be considered complex systems. (Baccarini, 1996). Although there is no accepted definition of complexity and how its value can be measured, it seems undeniable that buildings characterised by the traits that distinguish this systems. Dynamic nature, instability and continuous rapid and often radical changes in most cases unpredictable are undeniable characteristics of the building process (Bennet, 1991).

A new paradigm is therefore required to manage this kind of complexity, so we will consider buildings as systems that include many individual agents, represented by the conceptual entities involved in the representation's creation on which it bases the design process. The entities of the model interact with each other according to rules that organise, at a local level, the interaction among them.

The interaction between building objects is mainly local as 'no system-wide set of rules determining the interaction. The only rules are the rules at the level of the agent itself' (Stacey, 2001).

On this basis, an advanced 'Swarm' simulation prototype based on Distributed Artificial Intelligence (DAI) and Multi-Agent-Based Simulation (MABS) is proposed, because nowadays appear a suited way that may approach the complexity of building design and process management. (Bonabeau et al., 1999; Eberhart et al., 2001; Kennedy, 2006).

A prior goal of the research is the improvement of the cognitive capabilities of Design Support tools, in terms of automation in defining the rules to impose to building objects, the communication among the model and Designer and, to provide stakeholders of tools capable to auto-organise building properties to satisfy determined goals.

So, we make The proposed Swarm Simulation Model up of agents equipped with an inventory of possible actions related to the building entity represented. We can undertake action as an answer to some activation conditions or triggering conditions. The primary goal of the agents is to achieve their own goals even though agents operate to a higher-level goal.

The development of this study implies the clarification of original ideas, to be analysed to better define the field of investigation. So, we structure the paper in original sections, ideally gathered into three macro-chapters:

- 1) clarification of the problem: there are encompassed section from 2 to 4. The relevant aspect of these parts are related to the different problem that always recurs into the design process like the choice of a better representation framework, the complexity of building sector, and the limits of most-common methodologies;
- 2) methodology: in these sections, from 5 to 7, we describe the way used to apply the Swarm-simulation in the BIM environment, via the connection of data from different sources (BIM database, agent modelling repositories etc);
- 3) further improvements and conclusion: the in-depth analysis on the Distributed Constraint Optimisation Problem (DCOP) is to be intended like the next step of the research, that will be focused on the application of statistical approaches to solve best-solution comparison.

2. CREATIVITY AND STANDARDISATION

The construction industry has traditionally suffered from a low level of productivity, linked to a variety of motivations of a predominantly organisational nature. Thus Several strategies have been adopted in the past to react to this negative trend, mainly trying to turn the building process in prefabricated components assembling activity, to emulate industrial processes that in the same period had achieved impressive productivity increments. But this effort doesn't seem to have had the expected impact on, despite the interest and investments made by important companies from other more productive industrial sectors (Kieran and Timberlake, 2004).

However, similarities are merely similarities. The design approach used in manufacturing sectors such automotive is nowadays based on a wide range of sophisticated simulation and physical prototypes construction, testing and destruction to measure performance starting from the initial phases of design activity (Elverum and Welo, 2014).

It carries production out in high-tech plants using advanced automation processes, which ensures high levels of productivity. The enormous investments that the automotive industry or, more generally, the manufacturing industry can mobilise for production plants, simulations, destructive and non-destructive tests, are justified by the resulting impressive mass production. Despite the massive research and development activity, time spent on design phases, important investments, simulations and tests on prototypes that the automotive industry can mobilise, millions of cars are recalled each year for more or less serious defects (Lucky and Takim, 2015). Today, the building process product is undoubtedly as complex as the products of the automotive industry or the typical industrial products. However, unlike in industrial production, the design activity in the building industry cannot be based on investments, design times, quantity and quality of prototypes to be tested and destroyed, highly industrialised production sites and, above all, mass production that makes similar approaches affordable.

So, each building is a prototype, always different or, in other words, a unique object whose design and construction undoubtedly cannot trigger investments comparable to an industrial product or, even less, allows destructive tests to be carried out. So, construction sites are accordingly characterised by low investments, comparable only to a small fraction of the total cost of the result. The designer is also required to guarantee the building's operability for a lengthy period, compared with other manufacturing products. In the building sector, defects may appear after a very long time, often decades, so Life Cycle Assessment (LCA) is so becoming a crucial aspect in the building design and construction process, to prevent high costly reworking activities (Sharma et al., 2011).

According to this approach, the simulation of a complex system does not require the knowledge, representation or formalisation of the complete system but only the exact definition of the agent's behaviour. Dimension and nature of problems that can be approachable with this system depend on the Agent number and type that the system comprises. We build the proposed swarm simulation system in a bottom-up way, adding different agents as the complexity of the system growth or the context change providing the scalability and flexibility level require addressing complex systems simulation.

Swarm Simulation systems are based, intrinsically, on parallelism and a-synchronicity, suitable for distributing systems in which agents, regardless of their location, can simultaneously interact with the context (Chen et al., 2018).

In robotics today we are witnessing a real renaissance, based on distributed intelligence and situatedness. The new paradigm for intelligent robotics focuses on physical embodiment and the absence of the representation. According to (Brooks, 1991), the traditional approach of artificial intelligence was based on a Top-Down approach in which representation played a fundamental role. An intelligent agent (e.g. a robot) constructed a

representation based on the context. We based the choices adopted using the 'artificial' representation ignoring the actual world.

Distributed intelligence systems, on the other hand, pursue a bottom-up approach, considering the agents located in the actual world (Situativeness) without the need to interact with abstract representations, but experimenting directly, through sensors (Embodiment), the world that surrounds them, reacting to the stimuli transmitted by the sensors.

Agents' intelligence does not derive only from the computational engine but also from the ability to process the signals coming from the external world, (Intelligence) to interact with the actual world and its internal components. (Emergence). This approach may appear inappropriate to model complex systems comprising buildings in the process aimed to their construction because during the greatest part of the process the object does not exist and thus there is no context where an agent can be placed.

In the development of the proposed prototype, an opposed approach is used. The agents will be located in the symbolic representation of the context or within the project environment.

3. DESIGN ACTIVITY AND REPRESENTATION

The design activity is aimed at the realisation of «artefacts that have desired properties and is carried out using a representation of things to be built» (Simon, 1988). The design activity represents probably one of the most studied activities to understand its very nature, characteristics, modalities under which is carried out and tools used to represent and communicate the design choices.

It has developed countless studies and experiences starting from Zwicky's General Morphological Analysis (Zwicky, 1967), to create tools to handle the design encountering soon in the first computers the perfect 'toolbox' usable to build the envisaged instruments. Those experiences have always been characterised by two distinct and conflicting approaches: to the utopian visions of Yona Friedman and Nicholas Negroponte who theorised the so-called 'Architectural Machine', was opposed the 'pragmatic' approach of Skidmore, Owings and Merrill (SOM) whose Building Optimisation Program could be called a 'Perfect Slave' (Llach, 2015).

Negroponte refused Coons' idea of design as an iterative process in which a 'creative stage' and a 'mechanical stage' were alternated and it verified the design choices against performance metrics. Friedman and Negroponte, on the other hand, envisaged a cohabitation between two intelligent species in a symbiotic relationship, that would create an 'extended designer' in «an ecology of mutual complementation design, augmentations, and substitution, even imagining the substitution of the architect considered being an 'elitist middlemen» (Negroponte, 1973).

The unifying trait of some experiences starting from Zwicky, Engelbart's augmented architect (Douglas C. Engelbart, 1962) to the most recent approaches (Novembri, Rossini, & Fioravanti, 2017) are, in contrast, aimed to the creation of a formal model building object to allow and support the design activity. Formal representation present, explicitly, in the above approaches used to develop, test and define satisfactory solutions representing «an acceptable compromise between divergent objectives» (Simon, 1996).

Design solutions are considered as a state of the formal representation and used, by the designer and stakeholders, to verify the compliance to performance requirements and constraints at the basis of design activity. In this approach the role of representation in the building process and in problem-solving is crucial. Many unsolvable problems can be easily solved using a different representation. So, «that representation makes a difference is a long-familiar point. We all believe that arithmetic has become easier since Arabic numerals and place notation replaced Roman numerals, although I know of no theoretic treatment that explains why» (Simon, 1996).

Starting from the 'Zwicky Box' and the design support system envisioned by Engelbart which may be considered one of the most brilliant definitions of today's BIM systems, to the sophisticated representation techniques inspired by Artificial Intelligence (AI) the representation have always had a crucial role in the research activity aimed at the creation of support systems for the design activity (Fioravanti, Novembri & Rossini, 2017).

It carries the design activity out its tests alternating representation changes and transformations to moments against the adopted goals (Visser, 2010). It verifies design choices acceptability against needs, mandatory constraints and 'rules of thumb' at the basis of the design activity.

The designer, using his models, defines performances that the building objects to fulfil goals of the project activity, defining the following construction phases preconditions. The project therefore also represents a fundamental

communication tool among various actors in the process and should provide all the required information to verify the decisions made and to ensure the correct and harmonious development of the following phases of construction.

4. REPRESENTATION: LIMITS AND DEVELOPMENTS

The designer uses sketch, drawings, and digital models to describe and explore the proposed solutions, so to different stages of the process correspond different representations, that are considered most appropriate to show and analyse the relevant information for that stage (Hornecker, 2007).

However, it is necessary to understand whether the representation that is made by the designer using the available tools develops, at the end of the process, into a project. Or, in other words, if the representation used and shared among involved stakeholders represents the 'artefacts', properties and performances required, to be correctly understood and spread to the various operators of the process. We should note it that the current representation systems do not appear this goal. Comprehensive prefiguration of the artefacts, Design Intent and Rationale must be got by reinterpreting the received information, considering that the designer original intentions rarely correspond to the original ones (Crilly, Good, Matravers, & Clarkson, 2008) (Crilly et al., 2008).

There are several strategies to verify the feasibility of projects, such as Integrated Project Delivery - IPD integrated with BIM systems, or the benefits - in terms of financial management of the project - that can be recorded with methods such as Earned Value Management - EVM (Jones, 2014). These methodologies, although successful in high-level practice, are consistent with model analysis processes, but do not trigger 'self-generative', autonomous and critical processes within the model itself. This is, therefore, a strong limitation of the state-of-the art because it shows how the employable technologies are oriented towards 'ex-post' analysis regarding the possibility, given by the proposed methodology, of obtaining an overall reactivity of the model, in an autonomous way, regarding the goal of achieving imposed quality standards.

So, although the BIM systems are equipped with reactive capabilities, these are strictly limited to reporting the violation of imposed constraints, mainly geometric. The lack of semantic recognizability of the current systems, in fact, does not make it possible to recognise, directly in the modelling areas, the violation of rules related to the achievement of objectives, or rules that are not strictly geometric.

It is therefore clear that current design tools cannot completely gain knowledge and information involved in a project. The relationship between the knowledge owned, and the tools used to support it has been analysed for a long time. According to Perkins, (1993) it appears natural to consider the notes taken by a student because of a future use leads to consider the knowledge owned as distributed between the student himself and his notes, laboriously catalogued and organised. We must move from 'Person-Solo' to 'Person-Plus', with his tools and the catalogued and organised knowledge living outside.

By analogy, it does not seem possible to consider a 'Designer-Solo' but rather, in order to share the complete design, seems more reasonable to consider the complex made up by the designer and his knowledge. These are to be integrated and completed by the set of documentation, regulations and experiences gained in previously developed projects.

The design activity carried out by the 'Designer-Solo' in fact uses an ecosystem of tools aimed at adopted choice, representation, check and long-term storage. The complex made up by the designer 'internal' knowledge and 'external' knowledge can be defined, by analogy, as 'Designer-Plus'.

This intimate union and collaboration between designer and the tools used in architectural design immediately appeared one possibility offered by first computers, that represented a revolutionary new tool (Singh et al., 2011).

The intimate connection between computer and designer that originally appeared to conflict with the original vision of Artificial Intelligence aimed to the substitution of the designer, appears today as one of the most promising ways and substantially refers, perhaps with great delay, to Engelbart's visionary concept of the 'Augmented Architect'. Several experiences have been historically carried out according to this approach to develop design support tools to enhance the designer's capabilities (Carrara, Kalay, & Novembri, 1994). So, if it doesn't exist a 'Designer Solo', in the same way, it does not seem possible the existence of a 'Design Solo'.

The attempt to encompass in the project the so-called 'Design Intent' and 'Design Rationale' (Regli, Hu, Atwood, & Sun, 2000) answers to this consciousness and, emphasise the design tools inadequacy as designer solution

communication vehicle to share the design information, govern the construction process and improve the collaboration among involved stakeholders. Thus, it seems impossible to take into consideration a project without dealing with the part of the designer knowledge strictly related to it, to elaborate and share the envisaged solutions.

Available support systems, often sophisticated, are nowadays mainly based on parametric modelling. These are gradually replacing the old one, deeply changing the building objects representation methods and, increasing the semantic level of the representation (Sacks et al., 2017). Building objects can now be recognized by current BIM systems as logical entities with attributes, default values and, limited, with some reactive abilities to the modification introduced by the Designer.

The geometric modelling approach still appears yet to be most widely used. This approach emphasises the paradoxical lack of even the smallest levels of common-sense knowledge as also sophisticated tools still allow creating unreasonable objects impairing information exchange and sharing (Bloch & Sacks, 2018). To overcome these difficulties, we propose a so-defined Swarm Modelling Approach, to create a pro-active tool to support design activity and to represent, manage and transmit the knowledge needed for a correct interpretation of contents in the building process.

5. SWARM SIMULATION MODELLING APPROACH

Distributed intelligence and multi-agent systems appear today to be a promising approach adaptable to attain the level of accuracy and 'intelligence' of design support tools previously envisaged. The success achieved by this approach also in buildings (Minar, Burkhart, Langton, & Askenazi, 1996) has generated significant interest due to the ability to simulate complex systems combined with top levels of modularity, scalability and flexibility.

Physical embodiment and the absence of the representation appears to be one of the major reasons for this success. In contrast with the traditional approach of Artificial Intelligence, based on the representation of the context where agents live and work, the representation of the context is no longer necessary. Agents react to external signals coming from the context where they are situated. Reality is experienced and changed by Agents respectively through sensors and actuators (Drogoul, Vanbergue, & Meurisse, 2002). In this way, Agents perceive the reality interacting with each other and to the context, possibly changing it.

This approach may seem unsuitable to model building object's properties and behaviour not in the context, because, during the design activity, they still don't exist. Agents are symbolic entities embodied in a symbolic computational system interacting with the external world sending and receiving messages with other Agents, experimenting and changing the context using their sensors and actuators. It creates the absence of representation from an informatics point of view. Interaction with the environment does not require to agents to build a symbolic representation of the surrounding context because it's unnecessary, and the Agent doesn't know if the context in which they embody it it's real. The knowledge of the world owned by the agent derives only from sensors, which receive and send signals interacting with a world that can be symbolic and in ongoing development.

Based on this principle, we adopt an approach that could be called recursive. The almost-detailed abstract and symbolic representation of the context in the world where agents are situated, act, interact, send and receive messages from sensors and where, through actuators, take actions aimed to change the context in which they live (Abar, Theodoropoulos, Lemarinier, & O'Hare, 2017).

BIM models, in this sense, make up a 'real-symbolic' world - the BIM World, where the 'building digital twin' is located, representing an ideal context in which it can embody agents. Every entity of the BIM model are linked to an agent, that we call 'mirror agent' that complement and enhance the entities structure with the ability to interact with other BIM entities, change the model to cooperate with the design activity (Novembri, Rossini, & Fioravanti, 2017).

When the Agents in the 'BIM World' interact with each other applying a Swarm behaviour, the simulative Agent-model is created. Agents dynamically react to inputs originating by the designer and, with continuous monitoring involving the fulfilment of the objectives of all agents verifying the proposed solutions, suggest or directly undertaking actions required to comply with design goals and rules, using the designer's choices as a basis to optimisation processes activation (Rogalska, Bożejko, & Hejducki, 2008), aimed to improve the adopted choices to comply, with the highest degree, with goals.

6. HYBRID AGENT PARADIGM

The growing interest in agent-based simulation systems nowadays there is not a universally accepted definition of the term Agent. Despite the wide popularity this system is achieving, structure and characteristics that Agents must have aren't widely codified.

The term Agent is normally used to refer to an abstraction, an idea or a concept that can represent any element whatever of an identifiable complex system like a BIM model. Agents are capable of autonomous behaviour taking, autonomous actions and can autonomously decide the most appropriate action to adopt to achieve the objectives for which they were designed (Van der Hoek & Wooldridge, 2008). An agent is normally equipped with a repertoire of actions that can be undertaken as a function of some conditions of activation. The primary goal of an agent is to achieve the set aim, even though agents operate in the interests of a user. The behaviour of the system derives from the behaviour and interaction of the community agents (Agency). Agents can cooperate with other agents to achieve their goals and are located in a context in this case represented by the BIM Model (Rossini, Novembri, & Fioravanti, 2017). An agent does not, however, have full control over the environment in which it is situated but can influence it. The same action activated twice in apparently identical conditions, can produce different results or even fail because Agent-based systems must allow that failure can be a possibility.

Different typologies of agents have been proposed to get desired autonomy and interaction levels varying over a wide spectrum, ranging from a purely reactive behaviour characterised by a reaction to stimuli with no reasoning to a purely deliberative behaviour, where the reaction results from a complex reasoning process. The so-called 'Hybrid Agents structure', that can provide both behaviours, appears to be the most suitable structure to adopt in the Swarm Model definition (Wooldridge, 2009), because allow both merely reactive than complex reaction, also toward the same input. We design hybrid Agents to attain an intermediate behaviour between the two extremes and represent one of the most promising Agent systems, based on 'Subsumption Architecture' (R. Brooks, 1986). The lower levels of the structure are purely reactive (i.e. 'switch behaviour') and automatically react to input from the context, whilst higher levels tend to 'proactivity', and can inhibit lower levels preventing immediate reactions. The subsumption architecture creates in this way a multilevel system (fig. 1) in which the higher layers use the lower-level competencies (Rodney Allen Brooks, 1999).

Agents are equipped with a utility function allowing the self-assessment of performing the actions, concerning a goal represented by the attainment of preservation of the desired state: an agent may, however, have as a goal the achievement and maintenance of an aim, also after it has been achieved. Regarding the Agent theoretical definition, there is no universally accepted software architecture or software development approach for multi-agent systems implementation. A distinction must be made between different approaches that can be used for this aim: Multi-agent systems differ from Object-Oriented (OO) programming (McCarthy & Levin, 1965) and distributed or concurrent software systems (Stojcev, 2000) which they are often inappropriately identified. In the OO approach Object are typically defined in terms of given methods and attributes, whereas an agent instead is defined in terms of the behaviour that this can express. The distributed software or competitive systems are characterised by pre-set synchronisation and coordination protocols whereas, in Multi-agent systems, the interaction between agents is autonomously established during the simulation. Object-Oriented and Agent paradigms share however many aspects, even if the two approaches are conceptually different. Since decades, Object-Oriented (OO) is a very popular programming paradigm (Booch, 1980). Objects are software entities that encapsulate some state and perform actions and methods over their internal state. This paradigm implements encapsulation, composition, inheritance, delegation and message passing. Objects remain passive and are activated only when a message processing is needed. So, the object existing in the system is irrelevant, unless some messages are sent to it. To communicate with an Object by sending a message, you know that the Object exists in the system.

The Actors paradigm was defined by Carl Hewitt (Hewitt, Bishop, & Steiger, 1973), and has been made popular by the Erlang language used at Ericsson, with significant success to build highly concurrent and reliable Communication system (Armstrong, 1996). Actors and Agents entities differ in some aspects. The Actor-based is a programming approach, whereas Agent-based is a modelling approach. So, The Actor Model represents a way to implement a Hybrid Actor-Agent Model. To this aim, the AKKA.net toolkit has been used. (Roestenburg, Bakker, & Williams, 2016).

Actors paradigm is similar to OO as this approach implements the same concepts but, in this case, Actors have control over their internal state. Actors like Objects communicate to each other exchanging messages, although in

this case is not strictly necessary to know if the Actor exists since messages can be broadcasted. Once an Actor receives a message, it may send messages, create new actors and change its local state (Ricci, 2016). Actors process messages are taken from a mailbox in a sequential way: to complete a task, actors can store information locally, waiting for other messages to complete the information needed to accomplish the task. The key difference between Actors and Object in the OO paradigm is autonomy of entities involved. In an OO system, an object exposes methods and attributes to other objects that can invoke the method itself, whereas Actors are software elements that autonomously decide, based on the received messages, to activate procedures and methods. In the Actor paradigm, the Actor is the fundamental element of a concurrent computation model used to develop parallel and distributed systems.

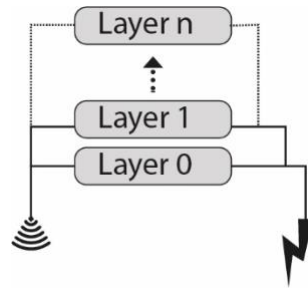


Figure 1. A simplification of the hierarchical scheme of 'Subsumption architecture'.

7. PROPOSED SYSTEM STRUCTURE

AS-BIM (Agent Swarm Building Information Modelling), is a general support system designed to interact with BIM tools to create a complex, dynamic, symbolic, and proactive representation of building objects aimed to support design activities. AS-BIM comprises a BIM system interlaced with Multi-Agent System (MAS).

The MAS system is composed by Agents in the BIM model, to integrate and enhance the BIM building objects representation providing them with the interaction, reactivity and proactivity abilities (Park et al., 2013) characteristic of multi-agent systems.

Agents living inside MAS interact and are closely interlinked with BIM objects through the Advanced Programming Interface (API) which represents, borrowing a typical robotic terminology, Agents sensors and actuators used to interact with the BIM system (fig. 2).

Provide to BIM objects, the Agent's typical behaviour represents a complex task, because of the need to coordinate simultaneous interaction between Designer, BIM system itself and external MAS. The API of existing BIM systems impose several severe restrictions. The autonomous activity of an external MAS aimed to react in real-time to the designer choices, undertake action in order BIM model modification and integration is so normally strictly disciplined since concurrent access to the BIM model data can easily generate race condition conflicts. (Gharachorloo et al., 1992). To guarantee the more compliant behaviour, the system works hence in distinct ways: the easiest consists in satisfying any request raised by the designer via the interface of the BIM-tool system, that waits for the task completion. Here, it establishes a Synchronous mode of interaction.

However, the designer does not use continuously the system's resources. The BIM system thus alternates moments when it is busy responding to the user and moments when it is in a state known as IDLE. Tasks running on a computer typically occupy only a part of processing time on the CPU. When the CPU has completed all tasks is the IDLE state, other tasks can use that. When the MAS, that run concurrently and asynchronously with the BIM system, needs to interact with the BIM World, raises a request that it will append them to the queue of the tasks to be undertaken among Designer's tasks request. So, whenever is possible, it passes control to the Agents interested to perform transactions in BIM context. This solution allows to avoid race conditions between Designer and Agents of the MAS and attains an efficient asynchronous partnership. It should be noted that a MAS is intrinsically transparent to the Network, in how an Agent can live anywhere in any computational resource connected to the Network, and thus can have access to high computational power.

Agents of MAS and BIM objects are only linked with a counterpart called Mirror Agent, that replicates part of the information placed in the BIM system integrated with those necessary to provide the expected behaviour. Families

and instances of the BIM System are linked to the actors by the Unique_ID identifier. (Autodesk Developer Guide, 2017).

Actor in this manner can represent non-editable behaviour of family and instances in the BIM system, acting as a cache memory of the Actor system too. When the required information is not available, Agents try to find it in the BIM system: if got, it is duplicated in the Agent memory. So, multiple access to the same information does not, therefore, require continuous interaction with the BIM system. If it changes the BIM object from which the information was derived, the system will refresh data in the Agents memory to avoid inconsistency between Agents and BIM systems.

This digital tool interacts with the BIM system and the designer: it carries these tasks out by the BIM Worker Agent (BWA) and the User Interface Agent (UIA) respectively. The BWA, located into the BIM API, perform the transactions requested by the agents in the BIM system, while the UIA performs a similar task interacting with the user, showing messages received from the Agents and translating the user's actions into messages for the MAS. The Request Handler Agent (RHA) serialises the requests of the agents managing the queue of events raised to the BIM System.

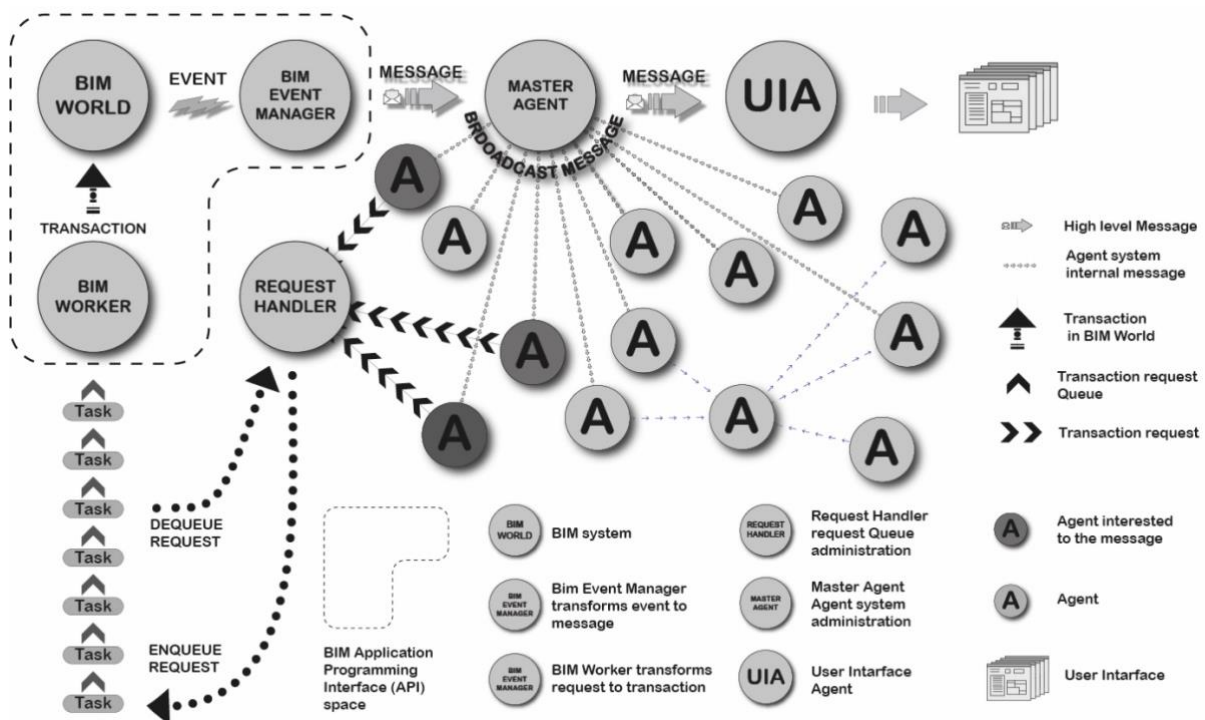


Figure 2. AS-BIM conceptual framework, with evidence on different ways in which a message become an agent, that interact with the whole system.

Each message sent by the Agents concerning information request or BIM model modification is interpreted and inserted in the system queue waiting to be satisfied. BIM Event Manager (BEM) is a key element of the system. It subscribes and intercepts the events of interest raised by the BIM system, sending them to the Master Agent. The typology of events intercepted by BEM will depend on the design phase and the user goals. Master Agent (MA) starts the MAS system and creates Agents using existing prototypes, connecting them through UniqueID to the respective BIM objects (fig. 3). Once the Agents system has been created, the BEM remains in 'listening mode' of BEM messages.

Two distinct situations can occur when a message is received: if the message is of general interest, it is broadcasted to all agents; otherwise, messages are of interest only for a specific Agent and specifically sent to them. Agents interested or the specific agent involved by the message respond activating them to its repertoire of actions sending messages and interacting with the other agents of the system and the user, requiring eventually to operate in the BIM System through the Request Handler and BIM Worker Agent.

8. THE ANSWER IS IN THE QUESTION: DISTRIBUTED CONSTRAINT OPTIMISATION PROBLEMS (DCOP)

The Swarm-Model goal is the development of an integrated representation of Building Objects, design goals and solutions that develop during the design activity to the last solution (Carrara et al., 1994). Design goals representation makes up a key element to ensure design choices, evaluation or distance quantification from optimal values.

Goals are strictly related to object representation as expected, desired and allowed values. They can be specified only for attributes explicitly represented or the existence of which is derivable. To this aim, the representation of Goals has been objectively and unambiguously defined as a set of Design Requirements, that represent constraints consisting as a set of desired or mandatory values defined for Objects Attributes. Performances are defined by the values achieved against the attributes affected by Design Requirements. The attribute set affected by a Design Requirement is the scope of itself: requirements may be added, edited, and deleted during the design process, as it develops towards a more detailed representation.

A direct correspondence thus subsists between characteristics and functions explicitly represented, and it to verify those to. Goals, through design requirements, represent a means to backtrack from characteristic to the original request. In this context, generalized values may be assumed by an attribute such as desired or mandatory.

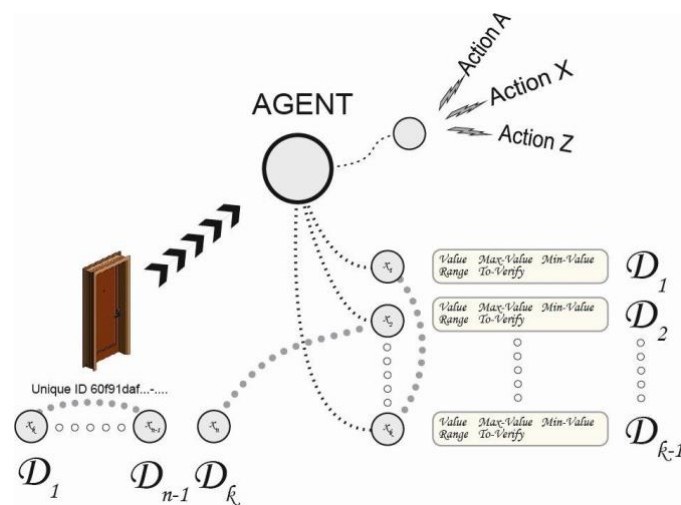


Figure 3. the connection among BIM object, that encompasses information and properties linked to the instance's geometry, and the Action-Message-Agent system.

Procedures that can obtain computable values. To predict and compare performances achieved by the emerging design solutions against the stated goals, object attributes will be Variables and Design Requirements as Agents. Optimal or near optimal solution suggested to the designer and choices we can thus treat check against constraints as a Distributed Constraint Optimisation problems (DCOP), (Petcu and Faltings, 2005). It represents attributes affected by design requirements as variables. It assigns each variable to an Object Mirror Agent, constraint affects whose attribute or to a Constraint Agent depending on the constraint typology as shown below.

The Object Mirror or Constraint Agent control attributes, values, defining acceptable and invalid ones. However, attributes Values are not independent so agents must coordinate their choices to determine workable solutions. We have considered two types of constraints defined as Hard Constraints and Soft Constraints.

It defines a Hard constraint as a request for a joint attribution to a set of variables of values belonging to an allowable domain. It is not always possible to assign compatible values to all variables, but it is often necessary to adopt not satisfying solutions violating some constraint minimising, as much as possible, the distance from a complete solution. If compliance with constraints is mandatory, a Constraint Satisfaction Problem (CSP) occurs, and it defines related constraints as 'hard' constraints. The aim of CSP is to find an assignment for all the variables that satisfies all constraints. This problem occurs when the feasibility of solutions needs to be tested to identify absurd and unrealistic solutions. Hard constraints are usually handled and hosted in Object Mirror Agents, to maintain Objects' attribute values within the admissible values, that cannot be derogated without violating the model's coherence.

Situations where constraints of compliance is not mandatory and the constraint violation leads however to quite appreciated valid solutions, the problem is characterised by so-called ‘Soft Constraints’. In these cases, we normally define a utility function for mapping all joint variable attributions to a scale of values which represents the cost of a violation of a constraint.

In these cases a CSP occurs (Carrara and Novembri, 1986), that is typical for constraints derived from design goals, which can be satisfied at different levels identifying so-called satisfying solutions (Simon, 1988). We should note it to this regard that a CSP can be considered as a Constraint Optimisation Problem (COP) where the utility function, in case of non-compliance, produce a relevant growth of costs in terms of computational time.

They will achieve design solution definition and proactive behavior both basing on Distributed Constraint Optimisation (DCOP) approach, leading the problem to the search of a variable assignment that satisfies all constraints and that optimises a global utility function, minimising its cost.

Formally a DCOP problem (Fioretto et al., 2014) is represented by a tuple $P = \langle A, X, D, F, \alpha \rangle$ where:

- $A = \{ a_1 \dots a_n \}$ is a finite set of agents.
- $X = \{ x_1 \dots x_m \}$ is a finite set of variables, with $n \geq m$.
- $D = \{ D_1 \dots D_m \}$ is a set of finite domains for the variables in X , where D_i represents the domain of the variable X_i .
- $F = \{ F_1 \dots F_k \}$ is a finite set of cost functions, with (1), where $x_i \subseteq X$ is the set of variables relevant to f_i usually defined as the scope of f_i .

$$f_i : X_{x_i \in X_i} D_j \rightarrow \mathbb{R}^+ \cup \{\perp\} \quad (1)$$

The number of variables in its scope is the value of a cost function. Each cost function f_i represents a factor in a global aim cost function f_i . Cost function represent the constraints of the problem but may be regarded as utility or reward functions.

- $\alpha : X \rightarrow A$ is a total and onto function, from variables to agents, which assigns the control of each variable $x \in X$ to an agent $\alpha(x)$.
- $\alpha(f_i)$ will denote the set of agents whose variables are involved in the scope of f_i where $\alpha(f_i) = \{\alpha(x) | x \in X^i\}$.

An assignment of a valid value for a subset of variables of X represent a partial assignment, where the assignment itself is complete if for each variable a valid assignment is given. For an assignment σ a cost function f_i is deemed satisfied by σ if $f_i(\sigma_{X^i}) \neq \perp$. A complete assignment is a solution of a DCOP if it satisfies all its cost functions. The goal in a DCOP is to find a solution that minimises or maximise the total cost expressed by its cost functions (2):

$$\sigma^* := \arg \min_{\sigma \in \Sigma} Fg(\sigma) = \arg \min_{\sigma \in \Sigma} \sum_{f_i \in F} f_i(\sigma_{X^i}) \quad (2)$$

Where Σ is the set of all viable solutions. So, given an agent a_i , $L_{a_i} = \{x_j \in X | \alpha(x_j) = a_i\}$ denotes the set of variables controlled by agent a_i or its local variables, and (3) denotes the set of its neighbouring agents.

$$N_{a_i} = \{a'_i \in A | a_i \neq a'_i \exists f_i \in F, x_\gamma, x_\zeta \in X^j, \alpha(x_\gamma) = a_i \wedge \alpha(x_\zeta) = a'_i\} \quad (3)$$

A cost function f_i is said to be hard if $\forall \sigma \in \Sigma$ we have that $f_i(\sigma_{X^i}) \in \{0, \perp\}$. Otherwise, the cost function is said to be soft. Constraints can be defined with regarding subset of variables.

Each agent operates asynchronously and knows only about constraints in which they involve it. Solution algorithms have been defined mainly using binary constraint graph using Boolean variables that can only assume True/False values.

The figure (fig. 4) shows a simple example of a Constraint Graph with four Variables $V = \{x_1, x_2, x_3, x_4\}$ with validity domain $D = \{0, 1\}$ and four constraints with their cost functions $f = (d_i, d_i)$ whose values are represented in the table. Two variables x_i, x_j are neighbours if a constraint exists between them. It controls each variable by an Agent which can handle multiple variables too.

Real-World applications require that global aim functions return a range of allowable values not limited to the Boolean True/False and the constraint binary nature imposed by the algorithm may appear unrealistic. It may be proved however that general constraint graph may be mapped to a binary constraint network based on Boolean True/False variables using additional variables and constraints (Bacchus and Van Beek, 1998).

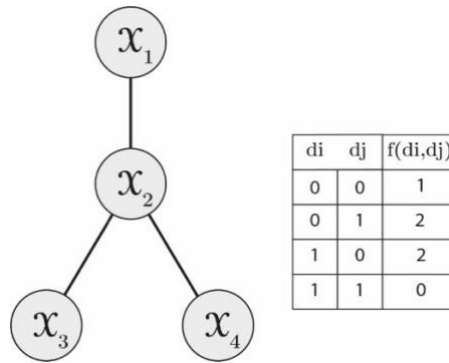


Figure 4. Constraint Graph with for variables.

Any DCOP problem can be automatically converted to Binary Constraints formulation using basically two different methodologies called respectively dual and hidden transformations. Dual Transformation derives from studies carried out on relational databases and was originally introduced by Dechter and Pearl (Peirce, 1931; Rossi et al., 1990).

According with Bacchus (Bacchus et al., 2002), during the dual transformation, new variables representing the original problem constraints are defined. As shown in (figure 5) the new variables c_1, c_2, c_3, c_4 are created. The constraint between $c_1 (C_1(x_1, x_3, x_6))$ and $c_2 (C_2(x_1, x_3, x_4))$ establishes that the tuple $c_1 \leftarrow (0,0,1)$ is incompatible with the tuple $c_2 \leftarrow (0,1,0)$ to x_3 variable, as are assigned two conflicting values. Among the dual variables that share original variables is so defined a so-called dual constraint, that prevents solutions with not consistent values of the variables shared among constraints.

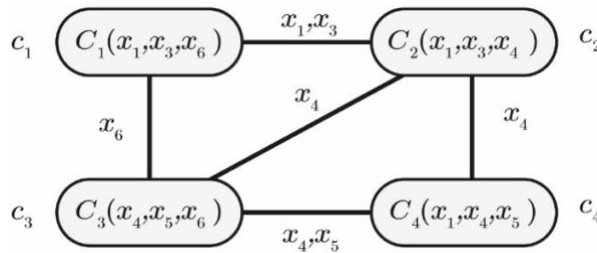


Figure 5. An example of dual transformation. A problem with four constraints based on three variables is transformed into a problem with four dual constraints.

In the hidden transformation, in addition to system and dual variables defined analogously to the dual transformation, 'Hidden constraint' defined between dual variables and each of the original variables in the constraint are added. These new variables guarantee that the original variable's value is the same, as the value assigned by the dual variables. The problem thus becomes characterised by ten variables, six original and four additional (fig. 6).

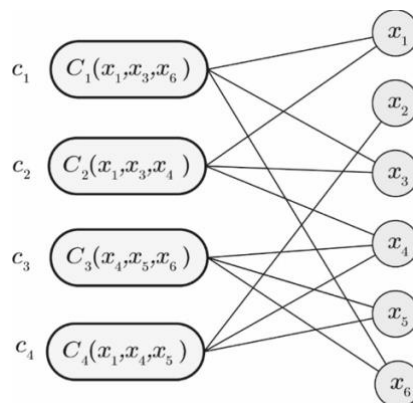


Figure 6. An example of dual transformation. A problem with four constraints based on three variables, is transformed into a problem with four dual constraints.

The constraint established between the dual variable $c_1 (C_1(x_1, x_3, x_6))$ and the x_1 constraint establishes that the dual variable value $c_1 \leftarrow (0,0,1)$ is compatible with the assignment $\{x_1 \leftarrow 0\}$ but $c_1 \leftarrow (0,0,1)$ is incompatible with $\{x_1 \leftarrow 1\}$.

In a first pre-processing phase, agents are ordered in an acyclic parent-child edges-based Depth-First Search tree (DFS) in which an agent is identified as the tree root where, thanks to the transformations described above, all agents have a single parent.

In the first step of the algorithm, each Agent selects a value in the validity range and communicates it to the neighbour agents following the constraint graph edges. Two messages are sent to other agents concerning respectively the adopted value (VALUE Message) and the threshold value (THRESHOLD Message). Threshold values represent the limit beyond which the backtrack procedure is activated. Agents at lower graph level receive the VALUE message only from higher level agents, while the THRESHOLD message is sent only between child-parent chain (fig. 7).

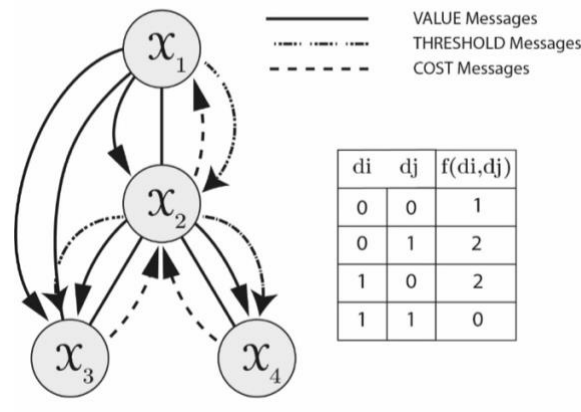


Figure 7. Communication among agents in the optimization procedure.

Once an agent has received any message, he checks if controlled variable value changes are necessary, computes costs related to the choices taken and, eventually, adjusts his backtrack threshold by sending the new value to neighbour Agents and the threshold value messages to its children. Cost messages containing the cost calculated by the Agent plus costs received by its children is then sent to Agent's parent.

Value messages go from the root to the tree leaves using the DFS tree while COST messages feedback go up from the leaves following the DSF tree while it sends threshold messages down the DSF tree to reduce redundant search.

DCOP algorithms performance depends on the variable's admissible values. Threshold and allowable ranges play a key role in this regard, and most times the variables correspond to BIM system object parameters. Admissible values domain definition is often infeasible a priori but depends on the values assumed during the search of a solution by the other variables.

Attempts have been made in order to dynamically define acceptable values ranges according to designer's preferences or based on already taken choices in the design process using statistical predictive techniques (Novembri et al., 2017), to provide the basis to use for a 'Heuristic Repair' process of the achieved solutions. The need for the heuristic approach is given by the large number of variables present in a building project, and allow designer to reduce the brute force needed for computing, and speed up the validation process as well. Thanks to this methodology, it is possible to start from a given result, to trace the entire causal chain and verify the plausibility of the values checked. In this way, besides having a methodology of iterative verification of information, it is possible to lighten the processes of parallel calculations that, if there had not been a heuristically determined starting point, it would have had to perform.

Based on the designer's preference and typically adopted we will define solutions by learning to leverage designers' tendency to adopt, where possible, analogous design solutions in comparable contexts.

It will carry heuristic repair out using the *min-conflict* approach (Minton et al., 1992) that, in most cases, has shown a successful behaviour (Armstrong, 1996). The initial solution defined on commonly accepted variable's values is revised by subsequent trials where values that minimise the number of possible constraints violations is adopted. The heuristic approach seems, compared to the systematic backtracking, better suited for the building design solutions search (Fig. 8).

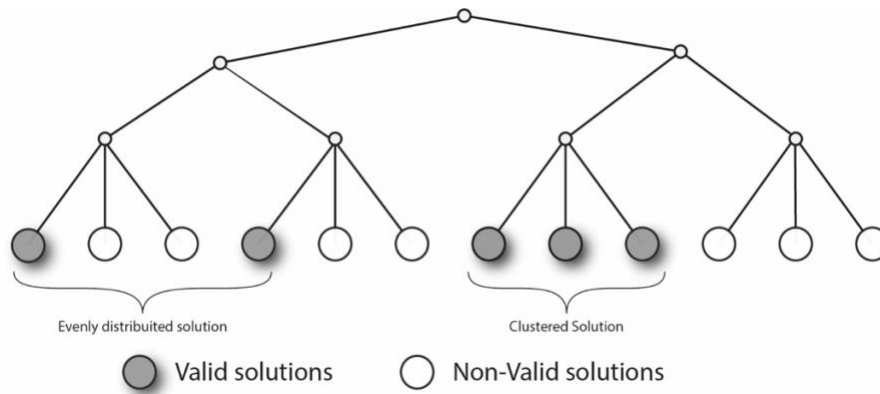


Figure 8. Clustered and evenly distributed solutions.

'Artifact' make up actually nearly decomposable systems (Simon, 1996), in which the design is characterised by subproblems often weakly interacting to each other (fig. 9). The composition of the partial solutions leads to additional constraints seem to generate DCOP solutions tree characterised by the presence of clustered solutions.

```

Procedure BACKTRACK (VARS-LEFT, VARS-DONE)
  var TEMP-VAR
  If (all variables are consistent)
  then
    solution found
    STOP
  Assign to TEMP-VAR var From VARS-LEFT in conflict
  Remove TEMP-VAR from VARS-LEFT
  Push TEMP-VAR onto VARS-DONE
  Let VALUES = list(allowable values)
  order VALUES upward-order
  order VARS-LEFT according to number of conflicts with variables in VARS-LEFT
  For each VALUE in VALUES, until solution found
    If VALUE not conflicts with any variable that is in VARS-DONE
    then
      Assign VALUE to TEMP-VAR
    else
      Call INFORMED-BACKTRACK (VARS-LEFT VARS-DONE)
    end if
  end for
end procedure

Main Heuristic Correction
  Assign to all variables an initial value.
  Let VARS-LEFT = list of
  Let VARS-DONE = nil
  Call INFORMED-BACKTRACK (VARS-LEFT VARS-DONE)
End Heuristic Correction
  
```

Figure 9. Pseudocode formalisation of Heuristic correction procedure.

9. CONCLUSIONS

The path undertaken to increase productivity in the construction sector is the digitisation of processes. We move towards implementing the tools used in the processes, through the systematic application of AI techniques, to automate the mechanisms of modelling and solution of design issues.

One of the major limitations of the current tools is the lack of automatic *reactivity* of the models, in the terms in which a BIM model can automatically adapt its elements and components, to changes imposed by the designer, or to a change of input conditions because of the compliance of design goals. There have been, in fact, many developments of model analysis tools (e.g. the continuous evolution of clash detection tools) or in the field of knowledge sharing and collaboration. But, regarding the capabilities of the modelling tools, there is still a reactivity behaviour linked to simple geometric rules, or parametric algorithms managed, in parallel, by visual programming applications (i.e. Dynamo for Revit; Grasshopper for Rhinoceros etc).

So, the proposed system prototype is based on a new approach called Building Swarm Modelling, in which the information managed through a traditional BIM system is enriched and completed in order to provide building objects with a typical behaviour of Multi-Agent systems aimed at verifying the choices made by the designer from the early stages of the design activity while providing a proactive behaviour to the overall system.

Constraints and characteristics of building objects are represented by agents placed in the virtual model represented by the BIM model with which the Agents interact through the API of the system. Agents can manage what we have defined as Hard Constraints through which are represented the common-sense Knowledge, the so-called Thumb rules and mandatory constraints arising from the regulations. While hard constraints are aimed to ensure the consistency of the information embedded in the BIM model with proper objects, soft constraints are the design objectives for which we can define a degree of dissatisfaction.

The multi-agent system, through Distributed Constraint Optimisation problem-solving techniques, can test the choices made by the designer but, at the same time, independently suggest, during the project activity, different solutions based on the search for high levels of satisfaction of the hard and soft Constraints.

The next steps of the research will be the implementation of this framework in one or more case studies, to test the proactivity of this system, regarding the design constraints. After that, we will focus the developments towards a higher autonomy of the modelling tool, which will have to be equipped with machine learning systems able, first, to select the experience gained from previous projects, to understand the conditions of similarity of information. This is necessary for training the tool to the imitation mechanisms. Then we can verify the cognitive possibilities of a digital tool able, therefore, not only to govern the self-organisation of the model but, further, to produce experience automatically, from similar cases.

REFERENCES

- Abar, S., Theodoropoulos, G. K., Lemarinier, P., & O'Hare, G. M. P. (2017). Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review*, 24, 13–33. <https://doi.org/10.1016/j.cosrev.2017.03.001>
- Armstrong, J. (1996). Erlang—A Survey of the Language and its Industrial Applications. 96.
- Autodesk Developer Guide. (2017). Revit API Developers Guide. Retrieved 28 January 2017, from <https://knowledge.autodesk.com/search-result/caas/CloudHelp/cloudhelp/2016/ENU/Revit-API/files/GUID-F0A122E0-E556-4D0D-9D0F-7E72A9315A42-htm.html>
- Baccarini, D. (1996). The concept of project complexity—A review. *International Journal of Project Management*, 14(4), 201–204.
- Bacchus, F., Chen, X., van Beek, P., & Walsh, T. (2002). Binary vs. Non-binary Constraints. *Artif. Intell.*, 140(1–2), 1–37. [https://doi.org/10.1016/S0004-3702\(02\)00210-2](https://doi.org/10.1016/S0004-3702(02)00210-2)
- Bacchus, F., & Van Beek, P. (1998). On the conversion between non-binary and binary constraint satisfaction problems. 310–318.
- Barbosa, F., Woetzel, J., Mischke, J., Ribeirinho, M. J., Sridhar, M., Parsons, M., ... Brown, S. (2017). Reinventing construction through a productivity revolution. McKinsey Global Institute, Available at: www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/reinventing-construction-through-a-productivity-revolution (Accessed 6 June 2018).[Google Scholar].
- Bennet, J. (1991). *International construction project management: General theory and practice*. Butterworth-Heinemann.
- Bentley, M. J. C., & Building Research Establishment. (1981). *Quality control on building sites*. Garston, Watford: BRE.
- Bertelsen, S., & Sacks, R. (2007). TOWARDS A NEW UNDERSTANDING OF THE CONSTRUCTION INDUSTRY AND THE NATURE OF ITS PRODUCTION. 12.
- Bloch, T., & Sacks, R. (2018). Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models. *Automation in Construction*, 91, 256–272. <https://doi.org/10.1016/j.autcon.2018.03.018>



- Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). From natural to artificial swarm intelligence.
- Booch, G. (1980). Object-oriented analysis and design. Retrieved from <http://kmvportal.co.in/Course/OOAD/OOAD%20by%20KMV.docx>
- Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1), 14–23.
- Brooks, Rodney A. (1991). Intelligence Without Reason. Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1, 569–595. Retrieved from <http://dl.acm.org/citation.cfm?id=1631171.1631258>
- Brooks, Rodney Allen. (1999). *Cambrian intelligence: The early history of the new AI*. MIT press.
- Carrara, G., Kalay, Y. E., & Novembri, G. (1994). Knowledge-based computational support for architectural design. *Automation in Construction*, 3(2), 157–175. [https://doi.org/10.1016/0926-5805\(94\)90017-5](https://doi.org/10.1016/0926-5805(94)90017-5)
- Carrara, G., & Novembri, G. (1986). Constraint-bounded design search. *Computer-Aided Architectural Design Futures*, 146–157. Elsevier.
- Chen, Y.-S., Yang, H.-X., Guo, W.-Z., & Liu, G.-G. (2018). Promotion of cooperation based on swarm intelligence in spatial public goods games. *Applied Mathematics and Computation*, 320, 614–620. <https://doi.org/10.1016/j.amc.2017.10.022>
- Crilly, N., Good, D., Matravers, D., & Clarkson, P. J. (2008). Design as communication: Exploring the validity and utility of relating intention to interpretation. *Design Studies*, 29(5), 425–457.
- De Grassi, M. (1984). *Organizzazione e controllo del processo edilizio*. Vestro Editore.
- Douglas C. Engelbart. (1962, October). AUGMENTING HUMAN INTELLECT: A CONCEPTUAL FRAMEWORK.
- Drogoul, A., Vanbergue, D., & Meurisse, T. (2002). Multi-agent based simulation: Where are the agents? *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, 1–15. Springer.
- Eberhart, R. C., Shi, Y., & Kennedy, J. (2001). *Swarm intelligence*. Elsevier.
- Elverum, C. W., & Welo, T. (2014). The role of early prototypes in concept development: Insights from the automotive industry. *Procedia CIRP*, 21, 491–496.
- Fazli, A., Fathi, S., Enferadi, M. H., Fazli, M., & Fathi, B. (2014). Appraising Effectiveness of Building Information Management (BIM) in Project Management. *Procedia Technology*, 16, 1116–1125. <https://doi.org/10.1016/j.protcy.2014.10.126>
- Fioretto, F., Le, T., Yeoh, W., Pontelli, E., & Son, T. C. (2014). Improving DPOP with branch consistency for solving distributed constraint optimization problems. 307–323. Springer.
- Gharachorloo, K., Adve, S. V., Gupta, A., Hennessy, J. L., & Hill, M. D. (1992). Programming for different memory consistency models. *Journal of Parallel and Distributed Computing*, 15(4), 399–407.
- Hewitt, C., Bishop, P., & Steiger, R. (1973). A Universal Modular ACTOR Formalism for Artificial Intelligence. *Advance Papers of the Conference*, 3, 235. Retrieved from <https://eighty-twenty.org/files/Hewitt,%20Bishop,%20Steiger%20-%201973%20-%20A%20universal%20modular%20ACTOR%20formalism%20for%20artificial%20intelligence.pdf>
- Hornecker, E. (2007). Sketches, drawings, diagrams, physical models, prototypes, and gesture as representational forms. *Second International Workshop on Physicality*, 87.
- Kennedy, J. (2006). Swarm intelligence. In *Handbook of nature-inspired and innovative computing* (pp. 187–219). Springer.
- Kieran, S., & Timberlake, J. (2004). *Refabricating architecture: How manufacturing methodologies are poised to transform building construction*. McGraw-Hill New York.
- Llach, D. C. (2015). *Builders of the Vision: Software and the Imagination of Design*. Routledge.

- Lu, Q., Won, J., & Cheng, J. C. P. (2016). A financial decision making framework for construction projects based on 5D Building Information Modelling (BIM). *International Journal of Project Management*, 34(1), 3–21. <https://doi.org/10.1016/j.ijproman.2015.09.004>
- Lucky, B. O., & Takim, S. (2015). Manufacturing defects in the automobile industry, a case study of the remote causes and effects of Toyotas transmission malfunctions in cars. *International Journal of Engineering and Applied Sciences*, 2(8).
- Matthews, J., Love, P. E. D., Heinemann, S., Chandler, R., Rumsey, C., & Olatunji, O. (2015). Real time progress management: Re-engineering processes for cloud-based BIM in construction. *Automation in Construction*, 58, 38–47. <https://doi.org/10.1016/j.autcon.2015.07.004>
- McCarthy, J., & Levin, M. I. (1965). *LISP 1.5 programmer's manual*. MIT press.
- Minar, N., Burkhart, R., Langton, C., & Askenazi, M. (1996). The swarm simulation system: A toolkit for building multi-agent simulations.
- Minton, S., Johnston, M. D., Philips, A. B., & Laird, P. (1992). Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*, 58(1–3), 161–205.
- Negroponte, N. (1973). *The Architecture Machine: Toward a More Human Environment* (First. edition). Cambridge, Mass: The MIT Press.
- Novembri, G., Rossini, F. L., & Fioravanti, A. (2017). Construction time and cost optimization using A.I. and statistical methods, through Bayes-Point Machines. Maggioli S.p.A.
- Otey, M. J. M. (n.d.). A review of the design intent concept in the context of CAD model quality metrics. 10.
- Park, C.-S., Lee, D.-Y., Kwon, O.-S., & Wang, X. (2013). A framework for proactive construction defect management using BIM, augmented reality and ontology-based data collection template. *Automation in Construction*, 33, 61–71. <https://doi.org/10.1016/j.autcon.2012.09.010>
- Peirce, C. S. (1931). *Collected papers of Charles Sanders Peirce*. Harvard University Press.
- Perkins, D. N. (1993). *Person plus: A distributed view of thinking and learning*.
- Petcu, A., & Faltings, B. (2005). A scalable method for multiagent constraint optimization.
- Regli, W. C., Hu, X., Atwood, M., & Sun, W. (2000). A survey of design rationale systems: Approaches, representation, capture and retrieval. *Engineering with Computers*, 16(3–4), 209–235.
- Ricci, A. (2016). Programming with event loops and control loops – From actors to agents. *Computer Languages, Systems & Structures*, 45, 80–104. <https://doi.org/10.1016/j.cl.2015.12.003>
- Roestenburg, R., Bakker, R., & Williams, R. (2016). *Akka in Action* (1 edizione). Shelter Island: Manning Publications.
- Rogalska, M., Bożejko, W., & Hejducki, Z. (2008). Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling. *Automation in Construction*, 18(1), 24–31. <https://doi.org/10.1016/j.autcon.2008.04.002>
- Rossi, F., Petrie, C. J., & Dhar, V. (1990). On the Equivalence of Constraint Satisfaction Problems. 90, 550–556.
- Rossini, F. L., Novembri, G., & Fioravanti, A. (2017, July 9). BIM and Agent-Based Model Integration for Construction Management Optimization. 111–118. <https://doi.org/10.24928/2017/0267>
- Sacks, R., Ma, L., Yosef, R., Borrmann, A., Daum, S., & Kattel, U. (2017). Semantic Enrichment for Building Information Modelling: Procedure for Compiling Inference Rules and Operators for Complex Geometry. *Journal of Computing in Civil Engineering*, 31(6), 04017062. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000705](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000705)
- Sharma, A., Saxena, A., Sethi, M., & Shree, V. (2011). Life cycle assessment of buildings: A review. *Renewable and Sustainable Energy Reviews*, 15(1), 871–875.

- Simon, H. A. (1988). The Science of Design: Creating the Artificial. *Design Issues*, 4(1/2), 67–82. <https://doi.org/10.2307/1511391>
- Simon, H. A. (1996). *The Sciences of the Artificial*. MIT Press.
- Singh, V., Gu, N., & Wang, X. (2011). A theoretical framework of a BIM-based multi-disciplinary collaboration platform. *Automation in Construction*, 20(2), 134–144. <https://doi.org/10.1016/j.autcon.2010.09.011>
- Stacey, R. (2001). *Complex Processes of Responding in Organizations: Learning and Knowledge Creation*.
- Stojcev, M. (2000). Andrews Gregory R.: *Foundations of multithreaded, parallel and distributed programming*, Addison-Wesley, Reading, Massachusetts, USA. *Facta Universitatis-Series: Electronics and Energetics*, 13(3), 384–387.
- Vähä, P., Heikkilä, T., Kilpeläinen, P., Järviluoma, M., & Gambao, E. (2013). Extending automation of building construction—Survey on potential sensor technologies and robotic applications. *Automation in Construction*, 36, 168–178. <https://doi.org/10.1016/j.autcon.2013.08.002>
- Van der Hoek, W., & Wooldridge, M. (2008). Multi-agent systems. *Foundations of Artificial Intelligence*, 3, 887–928.
- Visser, W. (2010). Visser: Design as construction of representations. *Collection*, (2), 29–43.
- Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.
- Zwicky, F. (1967). The morphological approach to discovery, invention, research and construction. In *New methods of thought and procedure* (pp. 273–297). Springer.