applied sciences

MDPI

Article

A Multi-Agent System in Education Facility Design

Barbara Strug and Grażyna Ślusarczyk *

Institute of Applied Computer Science, Jagiellonian University, 30-348 Kraków, Poland; barbara.strug@uj.edu.pl
* Correspondence: grazyna.slusarczyk@uj.edu.pl

Abstract: This paper deals with a multi-agent system which supports the designer in solving complex design tasks. The behaviour of design agents is modelled by sets of grammar rules. Each agent uses a graph grammar or a shape grammar and a database of facts concerning the subtask it is responsible for. The course of the design process is determined by the interaction between specialised agents. Space layouts of designs are represented by attributed graphs encoding both topological structures and semantic properties of solutions. The agents work in parallel on the common graph, independently generating layouts of different design components while specified node labels evoke agents using shape grammars. The agents' cooperation allows them to combine a form-oriented approach with a functional-structural one in the design process, where the agents generate the general 3D form of the object based on design requirements together with the space layout based on the functional aspects of the solution. Based on the given design criteria, the agents search for admissible solutions within the design space that constitutes their operating environment. The proposed approach is illustrated by the example of designing kindergarten facilities.

Keywords: multi-agent system; graph grammar; shape grammar

1. Introduction

This paper deals with a multi-agent system that supports the designer in solving complex design tasks. The concept of grammar-based intelligent agents being generative subsystems that control the multi-functional design process is used. Intelligent agents communicate with each other and autonomously control the development of design subtasks. The behaviour of the design agents is modelled by sets of grammar rules. Each agent uses a graph grammar or a shape grammar and a database of facts concerning the subtask it is responsible for. Agents solving subproblems of the same kind (i.e., designing different buildings or layouts on different floors) work according to the same grammar. The course of the design process is determined by the interaction between cooperating agents.

The growing interest in MAS research is due to their ability to solve problems that may be too large for a centralised single agent, to ensure reliability, and to tolerate uncertain data and knowledge. Multi-agent systems (MASs) provide a good solution for distributed control in different types of applications [1].

In all agent systems, which are combined with the generative power of grammars, agents implement grammar rules of the same type, which are either shape or graph grammar rules. As in practical design applications, different parts of solutions can have different characteristics, so agents need to have different task-specific knowledge and play different, specific roles. Therefore, in the presented approach, where designing of educational facilities is concerned, agents use two different types of grammars, both shape and graph grammars.

In this paper, the proposed approach is applied to the conceptual stage of designing 3D forms of buildings and layouts of spaces in educational facilities. Educational facilities are composed of many different components (i.e., various types of ground spaces and buildings with different zones) the design of which should be based on various functional,

Citation: Strug, B.; Ślusarczyk, G. A Multi-Agent System in Education Facility Design. *Appl. Sci.* **2023**, *13*, 10878. https://doi.org/10.3390/app131910878

Academic Editor: Jürgen Reichardt

Received: 19 August 2023
Revised: 25 September 2023
Accepted: 28 September 2023
Published: 30 September 2023

organisational, and aesthetic requirements. Thus, designing such a facility requires integrating a form-oriented approach with a functional-structural one. The cooperation of specialised agents allows them to generate the structure of the whole grounds design and at the same time incorporate the generic 3D form of the buildings while maintaining various requirements and diverse aspects of the design process.

As various elements of educational facilities have different characteristics, it is desired to design them independently only with occasional cooperation when common elements are being processed or common resources are considered [2]. The manager agent generates the arrangement of the whole facility's grounds to be compatible with the given criteria and invokes agents to design various ground spaces and buildings with their floor layouts. Space layouts of ground designs are represented by attributed graphs describing both topological structures and semantic properties of solutions. The graph grammar-based agents work in parallel on the common graph, independently generating layouts of different design components like sports facilities, playgrounds, and recreational areas while specified graph node labels representing buildings evoke agents using shape grammars. The agents designing the space arrangements use graph grammars [3] to transform graphs representing design projects, while parametric shape grammars [4] are used by agents to create forms of buildings composed of cuboids. Once the generic form of a building design is created, the graph grammar-based agents generate floor layouts. The agents' grammars jointly generate a set of graphs representing design task solutions with properties specified by graph attributes. The special set of graph node labels, which allow for the activation of different agents working on different parts of the graph [5], ensure that no two agents work at the same time on the same part of the design. Based on the given design criteria, the agents search for admissible solutions in the design space that constitutes their operating environment.

In the presented approach, in one agent-based system, different types of grammars are combined. The agents of the system use two types of grammars and operate on both graphs and shapes at the same time. Thus, the proposed multi-agent system allows for a flexible approach to the design of complex facilities with different component parts. When requirements are added during the design process, the system can be equipped with additional agents, which can use various types of grammars. This supports the generation of alternative facility models with various arrangements of spaces, which can be easily adapted to suit different needs by changing the design requirements. Generating design alternatives expands the designers' way of thinking and increases their creativity. The presented approach is illustrated by the example of designing kindergarten facilities.

The main contribution of this paper consists of defining a multi-agent system capable of designing educational facilities with the help of grammars of different types. The inclusion of different types of grammars, a method of intercommunication between agents and common control, are also the authors' contributions introduced in this paper. Due to the specific characteristics of the problem of educational facilities design, which usually combine different types of elements, including both building designs and exterior designs, we believe that the proposed MAS can achieve this purpose as multi-agent systems are well-suited for solving problems that involve many perspectives and require integration of different techniques.

The paper is organised in the following way: In the Section 2, a literature review is presented, briefly summarising the research in agent systems related to design problems and in grammar and rule-based systems. In Section 3, the structure of the design system based on many agents is described and the main definitions are introduced. Section 4 presents the way in which the designs are represented in the multi-agent system proposed in the paper. In Section 5, grammars used as the rule systems for design agents are defined and examples of such grammars in the context of educational facility design are depicted. In Section 6, a design of kindergarten facilities with the use of a multi-agent system and the agents' grammars described in the previous sections is presented. In Section 7, a brief discussion of the results is presented and is followed by the conclusions drawn from the research as well as an indication of the possible ways of further development of this research.

2. Related Work

Multi-agent systems have been applied to various problems, including market simulation, monitoring, system diagnosis, and remedial actions [6]. An MAS designed for the organisation of agents acting in dynamic and uncertain environments where it is required to enable improvisation by sharing knowledge and adapting behaviour, is presented in [7].

Nowadays, agent technology determines the new paradigm in computer-aided engineering. Multi-agent systems have been proposed in many engineering fields like customer service management [8], collaborative intelligent manufacturing [9,10], distributed control [11,12], distributed production [13], supply chain management [14], and industrial applications [15]. They are less frequently used in the area of engineering the design of products [16,17], where important decisions concerning the options of design solutions have a great impact on the quality of the design process. The application of a multi-agent system supporting the decision-making process in design for recycling is described in [18]. In [19], a multi-agent system is used for product design and manufacturing. It shows that multi-agent systems play an important role in developing and analysing models of interactivity in design.

When the system is decomposed into individual agents' tasks, it can be treated as an optimisation problem with constraints upon the decision variables [20], where the system performance greatly depends on the decisions made by the interacting agents working on the basis of local information and with limited communication capabilities. Agents turn out to be effective in optimisation problems like flexible job shop scheduling [21] or traffic light coordination [22].

As among the main characteristics of agent systems are adaptability and flexibility, they are well-suited to deal with the dynamic nature of design which is particularly important in conceptual design stages. A multi-agent system called ADEA (Agents-based Design Activity Analysis) for modeling, simulating, and analysing design activity is proposed in [23]. The design space is modelled as a network of design parameter agents. The agents of the ADEA platform formalise the necessary design roles, characterising the design activity as well as the relationship between design parameters in the design space. Despite their advantages, multi-agent systems are scarcely used for urban and architectural design.

On the other hand, various types of grammars, especially shape and graph grammars, are widely used in engineering and architectural design [24,25]. Therefore, certain types of grammars are combined with agent-based systems in order to obtain effective design tools [26]. Grammatical models of multi-agent systems modeling eco-systems are described in [27]. A graph grammar-based multi-agent system supporting the design of house estates is presented in [28]. In [29], a shape grammar-based multi-agent system for designing inner hood panels is proposed. In this system, each shape grammar rule is associated with one instantiation agent that controls the application of this rule. The multi-agent shape grammar system which automatically generates product form designs compatible with user preferences is described in [30]. Different rules of a vehicle shape grammar are assigned to several system agents. Each agent selects a shape grammar rule to implement and chooses parameters on the basis of a preference function. An architecture of a multi-agent rule-based system for supporting urban and architectural design is presented in [31]. In this architecture, each agent explores a part of a solution using its own shape grammar.

In the proposed approach, the multi-agent system is applied to the architectural design problem which includes parts with different characteristics. Therefore, although each agent explores a part of a solution using its grammar, the types of these grammars can be different.

3. A Multi-Agent Design System Structure

In practical applications, where solution parts have different characteristics, different agents with specific roles and knowledge can be effectively used.

The proposed multi-agent system, which supports the design of educational facilities, consists of a manager agent and several task-specific grammar-based agents acting simultaneously and performing different subtasks by communicating through a common message buffer. The manager controls the design process and maintains all necessary information such as constraints, currently created designs, completed designs, and allowable agent interactions. The task-specific agents are each responsible for a separate part of the design and return their solutions to the manager. The overall design solutions are handled by a single manager agent, whose role is similar to that of the human design team manager [32].

At the beginning of the design process, the designer formulates a set of design requirements on the basis of a given design task and environmental conditions. These requirements, together with legal norms and standards, constitute the design criteria for a given project and are passed on to the manager agent. The constraints handled by the manager and sent to the invoked agents correspond to the requirements determining the designer's preferences, like the number of playgrounds or the existence of a parking lot. They are stored as the values of graph attributes and constrain the use of grammar rules by agents. The constraints provided by the manager agent also allow for specifying the parameters of shapes in shape grammar rules. As the agents can apply only rules which comply with the imposed restrictions, the generated solutions always fulfill the design criteria. If some constraints cannot be met, the solutions are not created. The information flow within the system is shown in Figure 1.

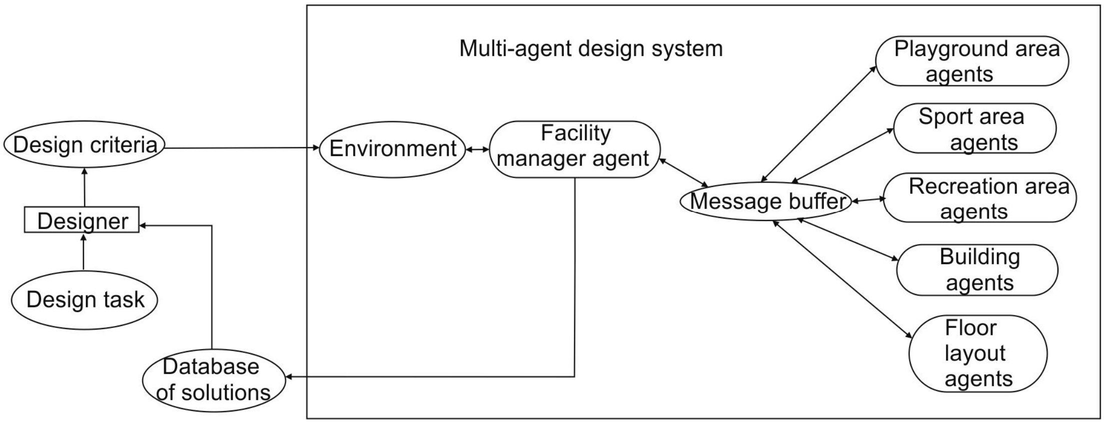

Figure 1. Information flow in the proposed multi-agent system.

The exchange of messages between agents in the system is developed based on a table architecture. Every message sent in the system has a specific type and associated data. There are two types of message templates: In the first type, a message consists of a special label, called a communication label, and a set of variables with values determining the parameters for the grammars of the invoked agents, like the number of building floors, areas, sizes, and the orientation of spaces. The second template type is in the form of a graph generated by an agent.

A system of agents is situated within an environment understood here as a design space composed of all possible designs in which the design process takes place. On the basis of the given design task and specified design criteria, admissible design solutions are searched for in the design space. The grammar-based multi-agent system, which supports the design process, can be defined formally as follows:

Definition 1. *A grammar-based multi-agent design system MADS is a tuple MADS = (Env, M_B, Ag_1,...,Ag_n), where:*

1. *Env is an environment in the form of a design space.*
2. *M_B is a message buffer used by agents.*
3. *Ag_1,..., Ag_n are grammar-based design agents.*

The distinguished facility manager agent is responsible for designing the arrangement of the whole ground. Layouts of the facility can be composed of six types of spaces intended for a kindergarten building, yard, playgrounds, recreational areas, sports facilities, and parking lots. The manager agent delegates tasks to the agents responsible for designing the areas created by him. The number and types of areas are specified in the design criteria. The manager agent also activates the agent responsible for generating a building form. Once this form is created, the agents designing the building floor layouts are activated.

All of the design agents, except for the agents designing the building forms, are equipped with graph grammars generating attributed graph representations of design solutions and task-oriented databases which together represent their local knowledge. The behaviour of the agents designing buildings is controlled by parametric shape grammars. Grammar rules encode the structural aspects of possible solutions, while values of graph node attributes correspond to the design requirements. Thus, the agent invoked for a node with a specified label knows the requirements related to his subtask.

The agents work in parallel on the common graph, independently generating layouts of different design components. Each type of agent is responsible for one design subtask and works on an independent part of the common graph representing a design. Each one of them is associated with a single grammar consisting of rules for designing a particular part of a facility. Several parts of a graph being generated can be developed by agents associated with the same grammar. Eventually, the manager agent combines the designs generated by all of the agents and places the obtained solution in the database. The created solutions can be visualised using a given interpretation together with the building forms.

The basic steps of the design process for the proposed multi-agent system are as follows:

1. The graph grammars for agents responsible for designing different types of outdoor spaces and building floor layouts are created.
2. The shape grammar for the agent responsible for designing the building form is created.
3. The templates of messages and communication labels corresponding to the labels of the initial elements of the grammars created in steps 1 and 2 are defined.
4. The graph grammar for the manager agent, which allows him to invoke task-specific agents, is created, taking into consideration the specified set of communication labels.
5. The manager agent reads the design requirements specified by the designer for a given kindergarten design task.
6. The manager agent invokes other agents, passing them the parameters for their tasks.
7. The manager agent combines the solutions generated by all of the agents and puts the obtained one into the database of solutions.
8. Steps 5–7 are repeated a predefined number of times.
9. The generated solutions are presented to the designer.

4. Design Representations in a Multi-Agent Design System

A design agent is a computational entity capable of acting autonomously in order to satisfy the requirements of the assigned design task. Each agent has a diverse goal to achieve. It has the ability to modify the design and exchange information with other agents. It takes decisions about design actions on the basis of its knowledge. In the perception process, it reads the message and stores the attributes describing the design requirements, like the number of rooms needed. In the decision-making process, the agent selects an action to be taken in the next step. It either sends a message, if the solution being generated is completed or does not satisfy the required conditions, or selects a grammar rule and applies it as a new design step. The agent decision-making process is based on its local knowledge, consisting of a grammar and a task-oriented database.

Let I denote a set of the agent's internal states composed of the content of long-term and short-term memory. A formal definition of a grammar-based design agent is as follows.

Definition 2. *A grammar-based design agent is a tuple Ag = (I, A, K_L, π, χ), where:*

1. *I is a set of the agent's internal states.*
2. *A is a set of possible actions.*
3. *K_L is a domain knowledge consisting of a grammar and a task-specific database.*
4. *π: $M_B \rightarrow I$ is the perception function, which changes the agent's state depending on the content of the message buffer.*
5. *χ: I x $K_L \rightarrow A$ is the decision-making function, which specifies the action to be taken based on the internal state and the domain knowledge.*

Layouts of kindergarten facilities are internally represented in the form of labelled and attributed hierarchical graphs [33], which encode both the topological structures and semantic properties of educational facilities. Nodes of these graphs represent different types of spaces, which are denoted by specified node labels. Moreover, nodes have attributes specifying properties of represented spaces such as the length, width, area, orientation or required equipment. Graph edges represent spatial relations between areas. In each graph node representing a component of the given object, subordinate nodes and edges, which describe the inner structure of that component, can be nested. Thus, hierarchical graphs allow for expressing hierarchical dependencies between different parts of objects.

Let $\sum$ be a set of graph labels and *At* be a set of node attributes.

Definition 3. *A hierarchical attributed labelled graph H is a system H = (V, E, s, t, lab, atr, par), where:*

1. *V and E are disjoint finite sets of nodes and edges.*
2. *s, t: E → V are functions assigning source and target nodes to edges, respectively.*
3. *lab: $V \cup E \rightarrow \sum$ is a node and edge labelling function.*
4. *atr: $V \rightarrow 2^{At}$ is a node attributing function.*
5. *par: $V \cup E \rightarrow V \cup \{\perp\}$ is a parent assigning function (symbol ⊥ indicates that a given node or edge has no parent), specified in such a way that no edge or node can be its own ancestor.*

In our approach, all of the design agents, except for the agents designing building forms, are equipped with graph grammars generating attributed graph representations of specified areas. They work in parallel, deriving the common graph and independently generating layouts of different design parts. The agents' behaviour is controlled by grammar rules which transform generated graphs or shapes.

Ground layouts of kindergarten facilities consist of various arrangements of required areas. Each layout can be composed of six possible types of areas representing spaces intended for a kindergarten building, yard, playgrounds, recreational areas, sports facilities, and parking lots. At first, the manager agent generates an initial non-hierarchical graph representing a ground layout. If the specified requirements are fulfilled by this layout, the agent activates other agents which start from the initial nodes with communication labels and apply grammar rules to generate graphs representing the layouts of particular areas. The returned graphs are located in the respective nodes of the initial graph, which becomes a hierarchical one. Two possible layouts of kindergarten facilities with the graphs representing them are presented in Figure 2. The labels *Building*, *Playground*, *Recreation*, and *Sport* activate the agents designing the areas with specific purposes. The layout presented in Figure 2b contains two areas labelled *Playground*, as one of them is intended for younger children, while the second one is for older children. If at the beginning of the design there is a requirement for two playgrounds, only the layout from Figure 2b meets this condition.

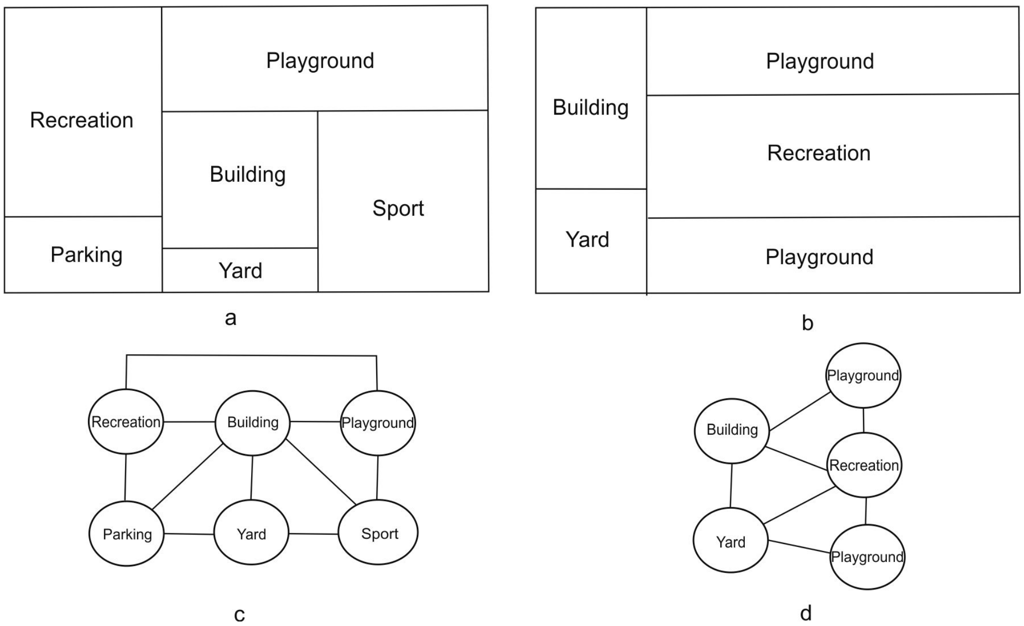

Figure 2. Two layouts of the facility and their graph representations.

5. Agents' Grammars

All of the design agents, apart from the one generating a building form, are equipped with graph grammars generating graphs representing structures of various education facility areas. These grammars operate on graphs that are not hierarchical. The successive design actions are performed by applying grammar rules chosen by design agents. In this way, the design agents control the design process, driving it towards solutions that are compatible with the design criteria.

Let us denote by L_H a set of attributed graphs and define π: $L_H \rightarrow \{TRUE, FALSE\}$ as a design predicate. A graph grammar is composed of a set of graph nodes with terminal and non-terminal labels, a set of rules, an embedding transformation, and an axiom being its initial graph.

Definition 4. *A graph grammar G is a system G =* $(\sum, P, C, X)$*, where:*

1. $\sum = \sum_N \cup \sum_T$ *is a set of non-terminal and terminal graph labels.*
2. *P is a set of rules of the form p =* (l, r, π)*, where:*
   - *l is an attributed graph containing at least one node with a non-terminal label and r is an attributed graph.*
   - π *is a design predicate determining the rule applicability.*
3. *C is an embedding transformation.*
4. *X is an initial graph, called an axiom.*

The application of the rule p to a graph H consists of substituting r for a graph isomorphic with l and replacing its connections with the edges coming to nodes of r, according to the embedding transformation defined for this rule in C. In our approach, neighborhood-controlled embedding (NCE) graph grammars [34] are used. It means that a graph l contains only one node and embedding is specified by a binary relation on a set of node labels and a set of nodes of r. For each pair $(x, \mu) \in C$, where x is a node of r and μ is a node label, an edge (x, y) is added between x and each node y labelled by μ, which is incident with the removed node. An agent can select the grammar rules leading to the design having the required properties by checking the conditions imposed on the values of graph attributes expressed by π. During the application of the rule, the values of the attributes of r are established on the basis of the values of attributes of l.

Six selected rules of an NCE graph grammar generating layouts with various arrangements of playground facilities are presented in Figure 3. Non-terminal labels of nodes start with a capital letter, while terminal labels are written in lower case letters. The area of the spaces and the material used for the different types of beddings are specified by the graph attributes. The rules *p1* and *p2* generate arrangements of areas with beddings filled with grass (nodes labelled by *Lawn*), rubber mat (*Rubber*), synthetic grass (*Synthetic*), and areas to be planted with trees (*Trees*). The first rule can be applied when the value of the attribute *area* of the node *v1* is greater than 50 m^2. Rules *p3* and *p4* generate arrangements of sandpits, carousels, and outdoor boards located on the synthetic grass, while rules *p5* and *p6* generate arrangements of swings, activity towers, and spring rockers located on rubber mats. For clarity, the embedding transformation is depicted only for rules *p3* and *p4*.

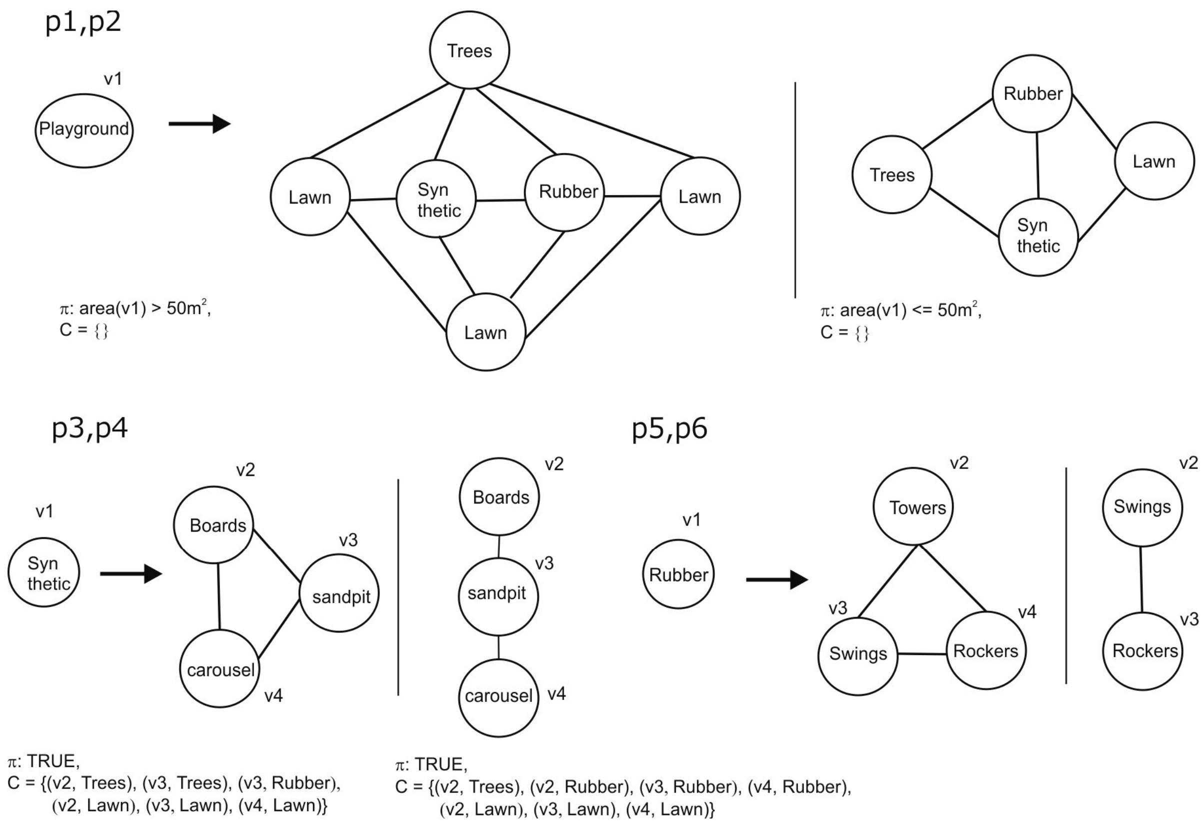

Figure 3. Selected rules of an NCE graph grammar generating layouts of a playground.

The agent designing a recreation area uses a graph grammar generating graphs representing various arrangements of trees, flower beds, shrubs, paths, benches, brooks, bridges, and stones. For a graph representing a recreation area arrangement, which is nested in a node labelled *Recreation*, and a visualisation of this arrangement see Figure 7.

An agent which generates a building form is equipped with a shape grammar generating 3D forms composed of parameterized cuboids. This agent selects the rules of the grammar and the parameters of cuboids on the basis of the requirements concerning a design building, like the number of floors and their areas.

Definition 5. *A shape grammar G is a system G = (M, S, R, I), where:*

1. *M is a set of markers.*
2. *S is a set of basic shapes called terminal shape elements,* $S \cap M = \emptyset$.
3. *R is a finite set of shape rules of the form* $\Theta \rightarrow \Psi$, *where* Θ *and* Ψ *are shapes whose components belong to S or M, and* Θ *must contain a marker.*
4. *I is an initial shape containing element* Θ, *such that* $\exists\, \Theta \rightarrow \Psi \in R$.

The application of the rule r to a generated shape K consists of finding a fragment of K isomorphic with Θ such that $\exists\, r\!: \Theta \rightarrow \Psi \in R$ (with an exact correspondence between the basic shapes and markers of Θ and the replaced fragment), finding a geometric transformation allowing to map Θ to the replaced fragment (scaling, rotation, translation, mirror reflection, and their combinations), applying the same transformation to the shape Ψ of the right side of the rule r, and replacing the part corresponding to Θ with the transformed Ψ.

In the parametric shape grammar, new shapes are created by assigning specific values to the parameters of basic shapes. Parameters can describe the size, location, and orientation of objects. In parametric rules, some object parameters may not have a specific value, which allows for the definition of many similar rules with one generalised rule. Assigning specified values to the parameters gives a specific rule. Each parameter can have a defined range, relationships can be defined between the parameters, and dependencies between the parameters of shapes of both sides of a rule can be established.

Four selected rules of a shape grammar generating 3D forms of kindergarten buildings are shown in Figure 4. The parameters of new shapes added by rules specify that they cannot be longer, wider, or deeper than the shapes of the left-hand sides, the angle of rotation of the new shape relative to the existing one should be in range [0°, 90°], and the protrusion should be between zero and half the length of the existing shape. The parameter ranges are shown for rules *p3* and *p4*. In Figure 5, two building forms generated by the specialised agent using the shape grammar from Figure 4 are shown.

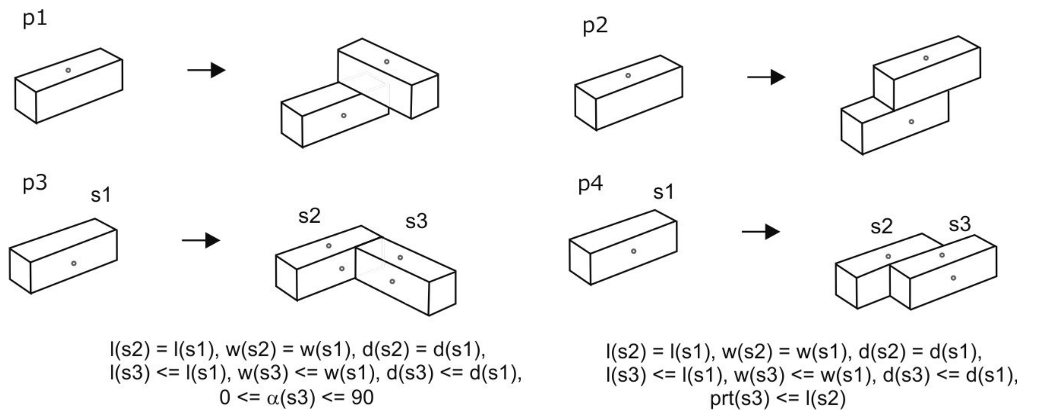

Figure 4. Selected rules of a shape grammar generating 3D forms of kindergarten buildings. S1 denotes shapes of the rule left-hand sides, while s2 and s3 denote shapes of the rule right-hand sides.

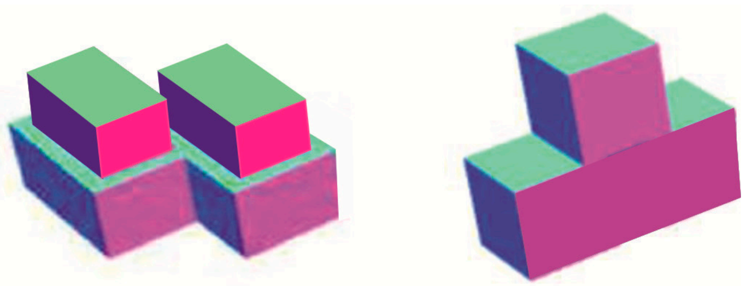

Figure 5. Two building forms generated by the agent using the shape grammar.

Once the form of a building is generated, the outlines of its floors are created as the intersection of this form with planes at given heights. Then relations between different floors, which describe the location of stairs and lifts, are specified. The obtained 2D shapes, together with functional aspects of solutions, like the location of stairs, the number and types of rooms, become a starting point for agents generating floor layouts.

A visualisation of the layout of the kindergarten facilities shown in Figure 2b is presented in Figure 6a. A 3D view of these grounds with a building form with two floors generated by the agent using its shape grammar is shown in Figure 6b. One of the possible floor layouts of the ground floor of the kindergarten building, which is generated by the agent specialised for this task (activated by the label *GroundFloor*) with the use of a graph grammar, is shown in Figure 6c. It is composed of an entrance hall, cloakroom, hall, playroom, dining room, kitchen, and a bathroom. A 3D view of this floor is presented in Figure 6d.

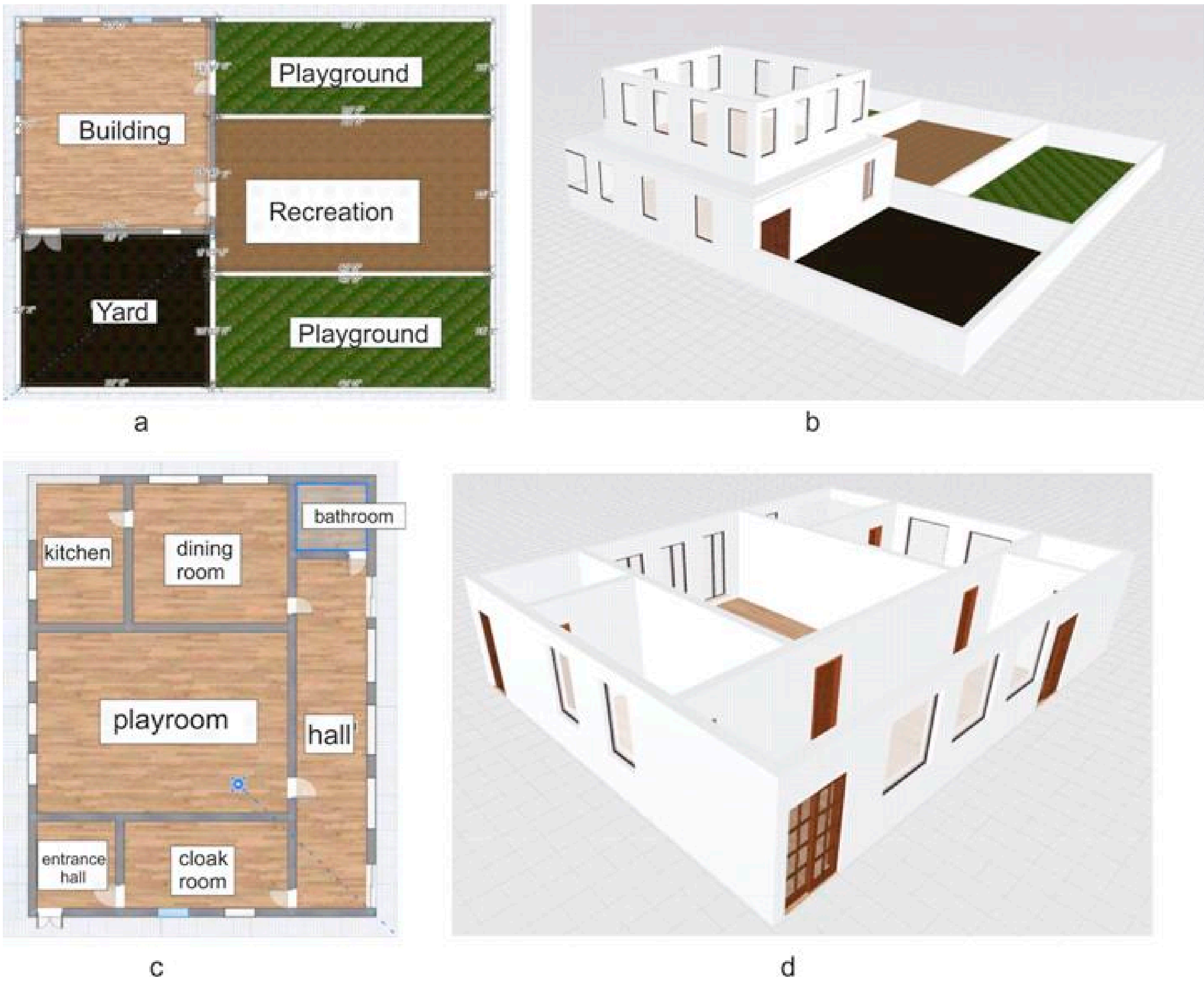

Figure 6. (**a**) A view of kindergarten grounds; (**b**) a 3D view of these grounds; (**c**) a detailed view of the ground floor; (**d**) a 3D view of the ground floor.

6. Designing of Educational Facilities Using Different Types of Agents' Grammars

The main goal of the considered design task is to provide a well-organised space for the educational facility with suitably arranged, different purpose zones and a building model with arranged floor layouts.

Educational facilities, on one hand, are similar to other architectural designs, but they have some distinctive characteristics. The main feature of such facilities is the fact that they usually combine different types of elements, including both building designs and exterior designs. In contrast to an apartment or office building, the exterior design has its own specific requirements that are strongly correlated with the building requirements. Moreover, there is a very high diversity of exterior elements that can be used. Another important characteristic is the diversity of the target users and thus, their requirements. The design has to take into account the age of the users, ranging from very young children to teachers of different ages, kindergarten caregivers, and parents. While accessibility for users of different abilities is nowadays expected in all buildings, the educational facilitates add another layer in this aspect. Depending on the target group age, different types of playground components have to be placed. Moreover, with integrated facilities becoming more and more widespread, children with special mobility needs have to be considered. Making such facilities as inclusive as possible requires special types of design objects placed in an appropriate way.

The design of an object composed of various components can be performed by different agents that each possess their own types of grammars. Each agent works independently on its own part of a design, but it can also call other agents which are able to solve particular design subproblems. Such triggering is performed using a specified set of labels.

Let $L = \{l_1 \ldots l_n\}$, $L \subset \sum$ be a set of communication labels, where each l_i invokes one agent which starts the generation of a subtask solution from an attributed node labelled by l_i. With each label l_i, a set of variables with values determining the design requirements is associated. These values are assigned to attributes of the graph node with label l_i in the agent's perception process. The values of attributes associated with the label *Building* specify parameters controlling the generation in the shape grammar. It should be noted that agents, which solve similar subproblems, can work using the same grammar. Such agents are referred to as agents of the same type. Each type of agent has its own activating communication label and, therefore, the number of communication labels equals to the number of different grammars used by agents of the system.

At the beginning of the design process, on the basis of functional requirements and constraints concerning the educational facilities and specified by the designer, the manager agent using its graph grammar generates an initial graph which represents the arrangement of required areas. Some nodes of this graph are labelled by elements of L. These elements determine the agents which will be activated to work simultaneously on the solution. The set L consists of the labels *Building*, *Playground*, *Recreation*, *Sport*, *GroundFloor*, and *Floor*. The values of the node attributes in this graph correspond to the design requirements related to the individual areas.

Each agent has a database of facts concerning the subtask it is responsible for (i.e., architectural norms, available plants, and equipment). Starting from the node with the communication label, the agent performs the derivation process using grammar rules. After each step of derivation, it evaluates the obtained partial solution with respect to the given requirements and facts contained in its knowledge database. On this basis, it selects the grammar rule which is to be applied in the next step. When the graph labelled only by terminal labels is generated, an agent sends it to the message buffer and the manager agent nests it into the graph node labelled by the element that activated the agent. The agent generating a building form sends the information about the number of floors in the design. Then the manager agent nests the graph composed of the node labelled *GroundFloor* and the required number of nodes labelled *Floor* in the graph node labelled *Building*. The nesting operation causes the initial graph generated by the manager agent to become hierarchical. The labels *GroundFloor* and *Floor* activate the agents that design layouts of the ground and other floors of the building using their graph grammars.

Graphs corresponding to a subtask solution generated by agents are located as subgraphs in nodes of the initial graph generated by the manager agent. Once the building form is generated by an agent, which uses a shape grammar, the subgraph consisting of as many nodes as the building has floors is placed in the node labelled *Building*. Eventually, the manager agent combines the designs generated by all of the agents into one solution and sends it together with the building forms to the database of solutions. The created graphs can be visualised using a given interpretation, which assigns geometrical objects to graph nodes.

A hierarchical graph, which is a result of nesting a graph representing a recreation area generated by one of the agents in a node labelled *Recreation,* and nesting a two-node graph representing two floors of the building shown in Figure 6b, with the form generated by the other agent, in a node labelled *Building* of a graph from Figure 2c, is shown in Figure 7a.

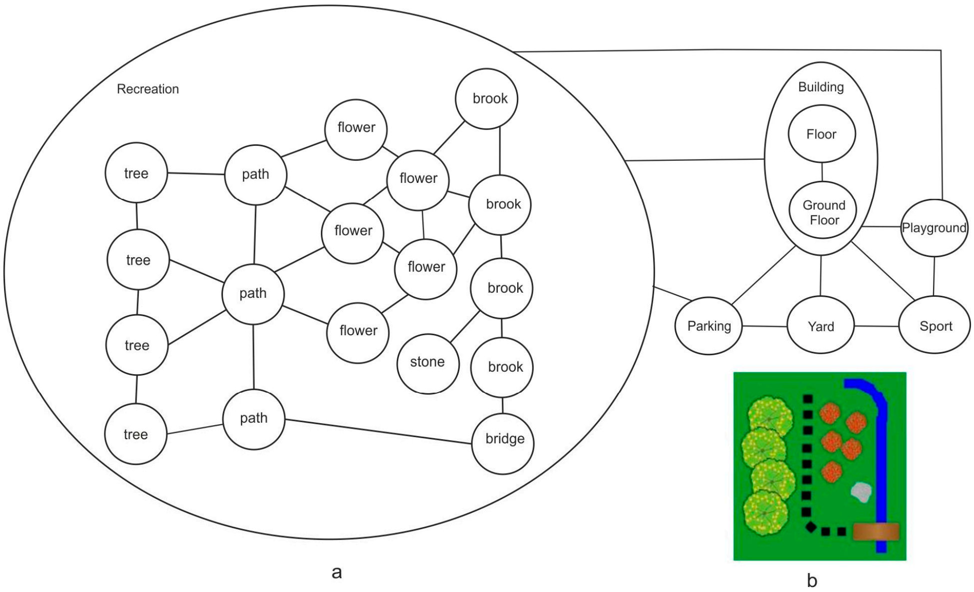

Figure 7. (**a**) A hierarchical graph generated by the manager agent by combining agents' solutions; (**b**) a recreation area arrangement.

A visualisation of a solution which consists of a two-story building, a facility layout with two playgrounds, a recreation area, and a yard is shown in Figure 8. In the top-right panels (number 2) different parts of the solution, like the building with the grounds, facility layout, recreation area, sports area, or building floor layouts, can be visualised. In the bottom-right panel (number 3) the area which can be currently modified by the designer is shown. The presented playground is composed of a tower with a slide, rocket, swings, springboard, and a seesaw. In panel number 1, the hierarchy of all possible components that can be placed in the playground is listed.

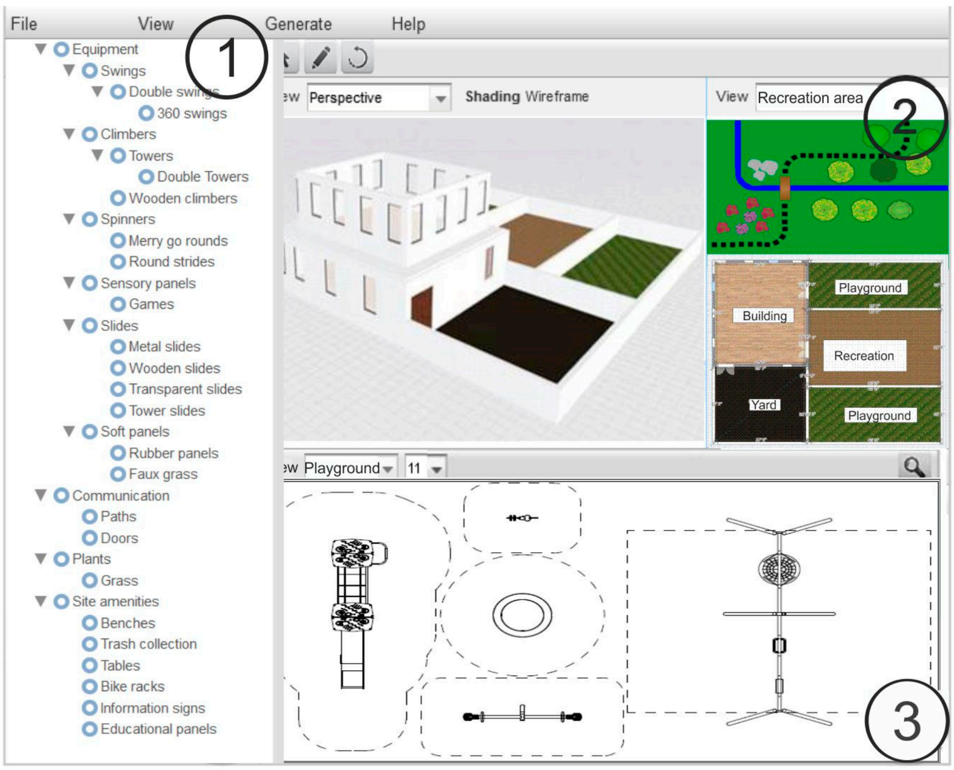

Figure 8. A visualisation of a generated solution.

After the predefined number of solutions is generated, they are shown to the designer for evaluation. It should be noted that agents generate only solutions which are compatible with the initial requirements, but they are only suggestions. The final decision about the best solution must be made by the designer who takes into account such aspects as functionality, aesthetics, price, and suitability for the user. When the designer wants to generate modified designs, he can use agents with different values of attributes specifying requirements.

7. Discussion

The approach presented in this paper is based on the multi-agent approach to solving design problems. Each agent is controlled by a formal rule system (a grammar) which describes all possible actions a given agent can perform. By separating the grammars among agents instead of using a single set of rules, the system is capable of incorporating additional rule sets when needed. It makes it very easy to add other types of equipment to the design as well as change the types of facilities to be designed. Moreover, the system may be easily adapted to a required type of design by selecting appropriate agents from the set of all available and thus limiting the complexity of the system. It should be noted that the presented approach is general enough to be also applicable in other design tasks, like, for example, designing house estates.

The method proposed has two main drawbacks, the first one is the need to design the grammar for each new agent, although this can be addressed by either using a visual grammar designer or allowing for the use of a language for rule defining, for example, an adaptation of the RuleMl or JSON rule engine. Another problem that may appear is related to the communication between agents. As in the proposed approach a message buffer is used, for each new agent, a set of messages has to be defined and messages accepted by other agents have to be appropriately adjusted. Both problems, while not trivial, can be addressed.

For evaluation purposes, designs generated by the proposed multi-agent design system were compared to the existing designs of kindergarten facilities. The comparison showed that the generated solutions resemble real ones reaching the level of human-made designs. Such a subjective evaluation method enables an intuitive assessment of generated results from a human perspective.

8. Conclusions

In this paper, a new approach to a distributed design with the use of many agents which can be controlled by various types of grammars is presented. A graph-based multi-agent design system supports the conceptual stage of designing building forms, layouts, and the arrangement of spaces in educational facilities. The actions of each agent are defined by one grammar, either a graph grammar or a shape grammar. Cooperating agents simultaneously perform different design subtasks, while communication between them is realised by a set of specified communication labels and a message buffer. The main advantage of such an approach is the possibility of adding specialised agents with appropriate grammars when different types of spaces are to be designed.

The obtained solutions, which consist of the proposed forms of buildings and layouts of facilities areas, that can be interpreted as designs, are evaluated by the designer. On the basis of this evaluation, the grammars of agents can be modified in order to generate more satisfying solutions. By changing grammar rules or adding some new ones, the agents can be adapted to the changing requirements of the design task.

Thus, defining a multi-agent system capable of designing educational facilities with the help of grammars of different types, introducing a method of intercommunication between them, and common control are the authors' contribution to the research domain introduced in this paper.

In the future, we intend to extend this approach by incorporating agents equipped with non-visual descriptive grammars. Such agents could compute quantitative or qualitative information about designs generated by other agents, estimate the costs of buildings, the planned gardens or playgrounds, and evaluate the accessibility of needed plants and equipment. In the present version of the system, the agents cooperate but they do not negotiate with each other. In the future, we would like the agents to have more influence on the other agents' decisions.

Author Contributions: Conceptualisation and methodology, examples and visualisation, writing—original draft preparation, review and editing: B.S. and G.Ś. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study does not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xie, J.; Liu, C. Multi-agent systems and their applications. *J. Int. Counc. Electr. Eng.* **2017**, *7*, 188–197. [CrossRef]
2. Grabska, E.; Strug, B.; Ślusarczyk, G. Educational Facility Design with Graph Grammar Systems. In Proceedings of the 26th EG-ICE International Workshop on Intelligent Computing in Engineering (EG-ICE 2019), Leuven, Belgium, 30 June–3 July 2019; Geyer, P., Allacker, K., Schevenels, M., De Troyer, F., Pauwels, P., Eds.; Volume 2394, pp. 193–203.
3. Rozenberg, G. *Handbook of Graph Grammars and Computing by Graph Transformation*; Volume 1: Foundations; World Scientific: Singapore, 1997.
4. Grasl, T.; Economou, A. From topologies to shapes: Parametric shape grammars implemented by graphs. *Environ. Plan. B Plan. Des.* **2013**, *40*, 905–922. [CrossRef]
5. Grabska, E.; Strug, B.; Ślusarczyk, G. A Multiagent Distributed Design System. In Proceedings of the 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAAMS 2009), Salamanca, Spain, 19–20 April 2009; Demazeau, Y., Pavón, J., Corchado, J.M., Bajo, J., Eds.; Advances in Intelligent and Soft Computing. Volume 55, pp. 364–373.
6. Catterson, V.M.; Davidson, E.M.; McArthur, S.D.J. Practical applications of multi-agent systems in electric power systems. *Eur. Trans. Electr. Power* **2012**, *22*, 235–252. [CrossRef]
7. Keogh, K.; Sonenberg, L. Designing Multi-Agent System Organisations for Flexible Runtime Behaviour. *Appl. Sci.* **2020**, *10*, 5335. [CrossRef]
8. Ong, S.K.; Lou, P.; Nee, A.Y.C. Multiagent-based Pay-Per-Use (PpU) distributed manufacturing. *Int. J. Comput. Appl. Technol.* **2011**, *40*, 114–121. [CrossRef]
9. Leitão, P. Agent-based distributed manufacturing control: A state-of-the-art survey. *Eng. Appl. Artif. Intell.* **2009**, *22*, 979–991. [CrossRef]
10. Monostori, L.; Rancza, J.; Kumara, S.R.T. Agent-based systems for manufacturing. *Ann. CIRP* **2006**, *55*, 697–720. [CrossRef]
11. Hadeli, T.; Valckenaers, P.; Kollingbaum, M.; Van Brussel, H. Multiagent coordination and control using stigmergy. *Comput. Ind.* **2004**, *53*, 75–96. [CrossRef]
12. Sun, J.; Zhang, Y.F.; Nee, A.Y.C. A distributed multiagent environment for product design and manufacturing planning. *Int. J. Prod. Res.* **2001**, *39*, 625–641. [CrossRef]
13. Trentesaux, D.; Dindeleux, R.; Tahon, C. A multicriteria decision support system for dynamic task allocation in a distributed production activity control structure. *Int. J. Comput. Integr. Manuf.* **1998**, *11*, 3–17. [CrossRef]
14. Forgeta, P.; D'Amoursa, S.; Frayret, J.-M. Multi-behavior agent model for planning in supply chains: An application to the lumber industry. *Robot. Comput.-Integr. Manuf.* **2008**, *24*, 664–679. [CrossRef]
15. Marik, V.; McFarlane, D. Industrial adoption of agent-based technologies. *IEEE Intell. Syst.* **2005**, *20*, 27–35. [CrossRef]
16. Ming Chao, K.; Norman, P.; Anane, R.; James, A. An agent-based approach to engineering design. *Comput. Ind.* **2002**, *48*, 17–27. [CrossRef]
17. Moon, S.K.; Simpson, T.W.; Kumara, S.R.T. An agent-based recommender system for developing customized families of products. *J. Intell. Manuf.* **2009**, *20*, 649–659. [CrossRef]
18. Dostatni, E.; Diakun, J.; Grajewski, D.; Wichniarek, R.; Karwasz, A. Multi-agent system to support decision-making process in design for recycling. *Soft Comput.* **2016**, *20*, 4347–4361. [CrossRef]
19. Ostrosi, E.; Fougčres, A.-J.; Ferney, M. Fuzzy agents for product configuration in collaborative and distributed design process. *Appl. Soft Comput.* **2012**, *12*, 2091–2105. [CrossRef]
20. Lin, X.; Shroff, N.B.; Srikant, R. A tutorial on cross-layer optimization in wireless networks. *IEEE J. Sel. Areas Commun.* **2006**, *24*, 1452–1463.
21. Ennigrou, M.; Ghedira, K. New Local Diversification Techniques for Flexible Job Shop Scheduling Problem with a Multi-Agent Approach. *Auton. Agents Multi-Agent Syst.* **2008**, *17*, 270–287. [CrossRef]

22. Bazzan, A.L.C. A Distributed Approach for Coordination of Traffic Signal Agents. *Auton. Agents Multi-Agent Syst.* **2005**, *10*, 131–164. [CrossRef]
23. Choulier, D.; Fougères, A.-J.; Ostrosi, E. Developing multiagent systems for design activity analysis. *Computer.-Aided Design* **2015**, *59*, 201–213. [CrossRef]
24. Haakonsen, S.M.; Ronnquist, A.; Labonnote, N. Fifty years of shape grammars: A systematic mapping of its application in engineering and architecture. *Int. J. Archit. Comput.* **2023**, *21*, 5–22. [CrossRef]
25. Kolbeck, L.; Vilgertshofer, S.; Abualdenien, J.; Borrmann, A. Graph rewriting techniques in engineering design. *Front. Built Environ.* **2022**, *7*, 815153. [CrossRef]
26. Grabska, E.; Grzesiak Kopeć, K.; Ślusarczyk, G. Designing Floor Layouts with the Assistance of Curious Agents. In Proceedings of the 6th International Conference, Reading, UK, 28–31 May 2006; ICCS 2006, Part III, LNCS 3993. Alexandrov, V.A., Albada, G.D., Sloot, P.M.A., Dongarra, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 883–886.
27. Paun, G.; Salomaa, A. (Eds.) *Grammatical Models of Multi-Agent Systems*; Gordon and Breach: Amsterdam, The Netherlands, 1999.
28. Ślusarczyk, G. Graph-based representation of design properties in creating building floorplans. *Comput.-Aided Des.* **2018**, *95*, 24–39. [CrossRef]
29. McCormack, J.P.; Cagan, J. Designing inner hood panels through a shape grammar based framework. *AIEDAM* **2002**, *16*, 273–290. [CrossRef]
30. Orsborn, S.; Cagan, J. Multiagent Shape Grammar Implementation: Automatically Generating Form Concepts According to a Preference Function. ASME. *J. Mech. Des.* **2009**, *131*, 121007. [CrossRef]
31. Santos, F.; Lopes, P.; Paio, A.; Eloy, S.; Reis, J.; Rato, V.M. A multi-agent expert system shell for shape grammars. In Proceedings of the 17th Conference on Computer Aided Architectural Design Research in Asia (CAADRIA); 2012.
32. Olson, J.T.; Cagan, J. Interagent Ties in Team-Based Computational Configuration Design. *Artif. Intell. Eng. Des. Anal. Manuf.* **2004**, *18*, 135–152. [CrossRef]
33. Strug, B.; Ślusarczyk, G.; Paszyńska, A.; Palacz, W. A Survey of Different Graph Structures Used in Modeling Design, Engineering and Computer Science Problems. In *Graph-Based Modelling in Science, Technology and Art*; Zawiślak, S., Rysiński, J., Eds.; Springer: Cham, Switzerland, 2022; pp. 243–275.
34. Janssens, D.; Rozenberg, G. Graph grammars with neighbourhood-controlled embedding. *Theor. Comput. Sci.* **1982**, *21*, 55–74. [CrossRef]