



From Floorplan to Navigation Concepts: Automatic Generation of Text-based Games

Reza Arabsheibani¹, Ehsan Hamzei¹, Kimia Amoozandeh¹, Stephan Winter¹, and Martin Tomko¹

¹Department of Infrastructure Engineering, The University of Melbourne, Parkville, VIC 3010, Australia

Correspondence: Reza Arabsheibani (reza.arabsheibani@student.unimelb.edu.au)

Abstract. Text-based games are environments in which defining the world, the representation of the world to the player (hereafter, agent) and agent interactions with the environment are all through text. Text-based games expose abstract, executable representations of indoor spaces through verbally referenced concepts. Yet, the ability of text-based games to represent indoor environments of real-world complexity is currently limited due to insufficient support for complex space decomposition and space interaction concepts. This paper suggests a procedure to automate the mapping of real-world geometric floorplan information into text-based game environment concepts, using the Microsoft TextWorld game platform as a case. To capture the complexities of indoor spaces, we enrich existing TextWorld concepts supported by theoretical navigation concepts. We first decompose indoor spaces using skeletonization, and then identify formal space concepts and their relationships. We further enhance the spectrum of supported agent interactions with an extended grammar, including egocentric navigation instructions. We demonstrate and discuss these new capabilities in an evacuation scenario. Our implementation extends the capabilities of TextWorld to provide a research testbed for spatial research, including symbolic spatial modelling, interaction with indoor spaces, and agent-based machine learning and language processing tasks.

Keywords. symbolic modelling, indoor space, text-based games, space concepts

1 Introduction

Modeling indoor space is necessary for indoor location-based services, including navigation assistance (Karimi, 2015). Indoor space models can be classified as *geometric* and *symbolic* models (Afyouni et al., 2012). While the *geometric* models use coordinate-based geometric representations of space, the *symbolic* models define the environment using abstract *concepts* and their relationships (incl.

connectivity and containment) (Karas et al., 2006). While current geometric models are widely used in simulation, symbolic representation can only be used if the simulation environment is capable of handling abstract concepts and performing qualitative spatial reasoning.

Text-based games are simulation environments in which the definition of the world, representation of the world to the agent (i.e., human or software agent), and the agent's interaction with the environment are all through text. Text-based games simulate the environment using linguistic concepts with well-defined semantics, such as *rooms* and *doors* rather than through interaction with graphic or geometric information. When agents require information about these concepts (rooms and objects in rooms) and their affordances, they interact via text by issuing an action command, such as *look*. In response, the simulation environment provides natural language descriptions (typically template-based) containing information about visible objects in the room (Figure 1). The capability of text-based games to expose the environment to agents using concepts instead of graphical scenes is appealing for research in indoor space interaction, including navigation. The expressive properties of current text-based games are, however, limited, and their suitability for experimenting with real-world navigation instructions needs to be assessed.

There are limitations to authoring a complex world in current text-based games since they have been developed for fiction games with simple, schematic environments. This is suitable for automatically created game environments or hand-crafted game environments. The repetitive interaction of agents with large amounts of automatically generated worlds enables agents to learn complex skills (Jansen, 2021) and solve complex tasks. However, the concepts available in text-based game models lack the expressive power needed for modeling real-world indoor spaces. Moreover, there is no automatic procedure for mapping generic geometry floorplan data formats into descriptions that are interpretable by text-based games and game agents.

```

West of House
You are standing in an open field west of a
white house, with a broader front door.
There is a small mailbox here.

-> Open Mailbox
Opening the small mailbox reveals a leaflet.

-> Take leaflet.
Taken.

-> Read leaflet
"WELCOME TO ZORK!"

```

(a) Playing Games

```

You are in a room called the backyard.

> Look
To the south, there is a closed screen door.
To the west, there is a closed wooden door.
On the eastern side, there is an exit with no door.

-> Open wooden door
You open the wooden door. Now you can see a table in
the other room.

-> Go west
You are in a room. An open wooden door leads east,
and a steel door to the north. There is a notebook
on the table.
You can read the notebook, go through its pages,
and write down something.

```

(b) Observation and Navigation

Figure 1. Examples of user interaction with a text-based game: (a) Opening scene of Zork, a classic text adventure game. (b) illustration of how players can observe the environment and navigate among rooms using only text-based commands, as the game provides no graphical interface.

The main research question addressed here is “*How can we computationally model indoor environments that enable textual interaction to support agent navigation?*” Therefore, our modeling focuses on (1) using descriptive, conceptual text definitions of indoor spaces; (2) the ability to represent concepts of the environment that are relevant to navigation; and (3) supporting the interaction of agents with the modelled environment. This paper addresses the hypothesis that a symbolic representation of an indoor space supporting navigation applications can be computed from the spatial information contained in floorplans in a way that schematic geometry (Rüetschi and Timpf, 2004) enables mapping between real-world geometric information and text-based game concepts. We use symbolic models to represent the environment (e.g., locations, objects, and possible actions) through textual descriptions.

The main contributions of this paper are:

- The definition of new indoor space concepts that mediate automatic mapping of geometric information into text-based games descriptions;
- The assessment of space decomposition approaches based on their applicability to the text-based games;
- The enriched user interaction, scenario creation, and definition of new navigation grammars (incl. the introduction of an egocentric reference frame) in a text-based game environment, using TextWorld as a case.

The organization of this paper is as follows: Section 2 elaborates on the history and general specifications of text-based games. We discuss gaps in current implementations of text-based games, their navigation applications, and the potential use of text-based games in the spatial community. Moreover, we elaborate on the indoor space decomposition methods and geometry-to-graph conversion approaches. Section 3 summarizes abstract concepts of the environment. In Section 4, we will propose an approach

for automatically converting geometric information of the floorplan into a navigation graph, generating a simulated environment from the navigation graph, and improving TextWorld’s user interaction grammar. Proposed concepts implemented in TextWorld are demonstrated in Section 5. Section 6 explains the data, software and libraries, and codes repository. Finally, Section 7 discusses findings and future works.

2 Background

2.1 Text-based Games

Text-based computer games trace back to the 1960s when the only communication form with mainframes was terminals (Nelson et al., 1997). Z-machine is perhaps the oldest low-level interpreter able to parse texts and create interactive fictional text-based games (Koirikivi, 2015). Inform, introduced by Nelson (2006), is a natural-language-based programming language for creating interactive games. Inform7 is the last generation of the Inform family of text environments (Nelson, 2011). Although there are other text-based games authoring systems such as *TADS*¹ and *Twine*², they are not as generally adopted as Inform7 by research communities (Jansen, 2021).

In Inform7, all *objects* are concepts definable by descriptions, and *grammar* is the set of actions related to the properties of the defined objects. Inform7 models the environment based on an *object tree* (Figure 2), a simplified hierarchy of objects with their properties and relevant grammar. The relation between Inform7 *objects* and *actions* is framed by *rules*. The atomic unit of text-based games is a *room*, and the connectivity of *rooms* is then encoded in *directions* (typically only supporting cardinal directions). A third basic concept of text environments is that of *region*.

¹www.twinery.org

²www.tads.org

The adjacent rooms with shared properties can be aggregated into a *region*. Inform7 also defines a small number of spatial rules and rudimentary reasoning capabilities at its core, e.g. the reverse direction (for example, since north and south are reverse directions, if room A is at the north of room B, then room B is at the south of room A) and containment relationships (for example, when room A is inside room B, if the agent exits room A, then the agent is in room B).

In text-based games, the agent's movement inside a room is considered trivial. This major conceptual gap limits the applicability of text environments for modeling interaction in real-world settings. Consider a real-world example: a customer cannot see every object inside a supermarket if they remain stationary. In contrast, the representation of a supermarket as a *room* in a text-based game generally means that an agent can see and interact without locomotion with all objects within the supermarket.

TextWorld is an environment the Microsoft Research team introduced as a sandbox for training reinforcement learning agents (Côté et al., 2018). TextWorld provides an operational and logical framework for creating Inform7 worlds. To generalize an operational logic for the interaction between the agent and TextWorld, *Ceptre*, a functional language that uses linear logic formulas to reason about games, has been introduced (Martens, 2015). For example, if P is the player (or agent), r and r' are rooms, $at(P, r)$ means player P is inside room r , and $north_of(r', r)$ means r is located to the north of room r' , then using defined datatypes and predicates it is possible to define the `go north` rule as: `go/north::at(P, r) & north_of(r', r) -> at(P, r')`. *Ceptre* introduces characters, locations, and objects as substantial types and defines predicates (such as character locations, sentiments, and actions) that enable state transitions in TextWorld. At its lowest level, TextWorld translates this logical framework to Inform7 code for compiling. The TextWorld structure can predict the next state when the agent chooses a possible action.

2.2 Text-based Games for Spatial Research

Text-based games and interacting agents may support a range of spatial research through their appealing properties:

- **Symbolic Modeling:** Text-based games rely on symbolic modeling of the environment, capturing the interrelationship of space subdivisions or hierarchies of spatial entities. In the case of indoor navigation, geometric models of indoor spaces work only for agents who are aware of (or able to determine) their current position in the environment, e.g. a robot with multi-radar sensory information of distances to immediate barriers (Zhou et al., 2022). TextWorld overcomes this by providing programmatic means to interact with an environment defined symbolically and

to develop agents with capabilities to interact with the symbolic environment.

- **Abstraction:** Text-based games expose an environment with fully deterministic, abstracted responses to agent actions or state transitions at a given level of detail. For example, if a door connects the current room to a room in the north, and the agent interacts (e.g., by issuing the `open the door` and `go north` command), the agent's location will deterministically change to the northern room. This process excludes lower-level reasoning about required motions such as walking toward the door, turning the key, and pulling the door. The abstraction capability of text-based games allows reasoning about the concepts and actions at a higher conceptual level of environmental interaction, liberating from lower-level physical interaction details. This level of abstraction allows focusing on hierarchical spatial reasoning, spatial learning, and problems related to navigation.
- **Extensibility:** In contrast to 3D virtual environments and simulators, agents in text-based games interact with objects using textual commands, without the need for grounding agent-object interaction. 3D virtual environments, e.g., AI2-Thor (Kolve et al., 2017), use photos to create scenes and include physics laws for the interaction of agents and objects. Action space is limited in 3D modeling, and extending agent-object or object-object interactions is expensive and complex (Jansen, 2021). The definition of new actions for available concepts in text-based games follows the same logic and templates of already defined actions. For example, opening a container in 3D environments requires a new agent-object interaction definition. At the same time, it is a straightforward extension to define a rich collection of action phrases in TextWorld (Hausknecht et al., 2020).
- **Quest Definition:** Text-based games expose the ability to simulate scenarios (*quests*) for the interacting agents. The scenarios can be constrained by time or the number of interactions. The agent can receive rewards when completing the *quest* or be otherwise penalized. Text-based games can support environments that change dynamically over time (e.g., as fire can extend in a building), and also, success states for the agent can vary, as a pedestrian may need to change the planned path.
- **Route Instruction Following:** Text-based game agents can be used for exploring the interpretability of route instructions. Route instructions are verbal descriptions for guiding a wayfinder that capture a sequence of states and actions (Frank, 2003). Similarly, text-based game quests are also sequential decision-making engines that consist of states and actions (Hausknecht et al., 2020). A similar semantic structure of route instructions and quests in TextWorld

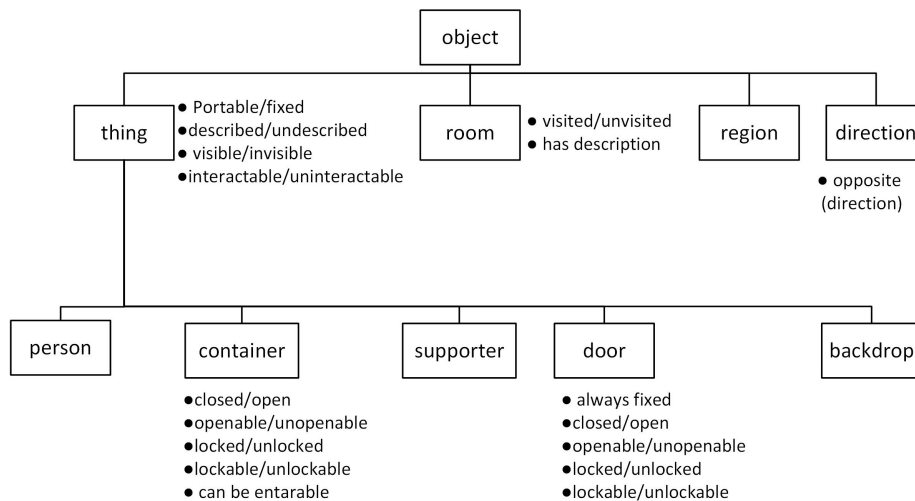


Figure 2. Inform7 object tree (Aikin, 2009). *room*: atomic spatial partition, *region*: aggregation of rooms, *direction*: cardinal and/or ordinal directions, *door*: connecting rooms, *backdrop*: scenery that may extend across rooms, e.g. sky, *person*: man, women, and pet, *container*: objects can be put inside containers, and *supporter*: a horizontal surface on which things can be put, e.g. table. Note: this is an abridged concept tree (e.g., the concept *Person* holds additional attributes not shown here for legibility).

can be used for assessing route instructions in guiding wayfinders in partially-observable environments. TextWorld offers agent interaction capabilities based on natural language commands from agents and descriptions of the environment back to the agents. The agents' navigation in the simulated world proceeds through this command-based interaction. Agents can follow provided route instruction based on rules (Ammanabrolu and Hausknecht, 2020), supervised learning (Adhikari et al., 2020), or reinforcement learning (Tuli et al., 2022). All agent navigation in TextWorld is allocentric and supports eight cardinal and ordinal directions (Figure 7) (Paillard, 1991). So the agent must be aware of the relative direction of the next room while navigating. For example, when the agent issues a command `go north`, if there is a room north of the current room, the agent's location will change accordingly but fail if such a room does not exist. People more commonly navigate using an ego-centric reference frame, especially in indoor spaces where recognizing cardinal directions may not be a straightforward task.

Currently, text-based games support only concepts of rooms, regions, and eight directions. Therefore, to use the abstract capabilities of text-based games in spatial domains, their concepts must be extended to capture the complexity of real-world indoor spaces and richer interactions. A model that can capture complex environments' spatial details enables the definition of more complex quests, reasoning, and instruction generation and following. Here we suggest an automatic procedure to generate text-based game descriptions directly from geometric information.

2.3 Geometry to Graph

Floorplans are often represented using simple 2D geometries representing the outlines of indoor spaces. Thus, rooms may be represented as polygons or polylines, doors or stairs as lines, and landmarks as points. These polygons and points may be annotated with additional information about the spaces, doors, or landmarks. However, such a geometric representation does not capture connectivity between room walls and is thus inadequate for routing and navigation.

Graph representations of spatial environments are expressive structures enabling the modeling of objects and their relationships, such as topological relationships (Marshall et al., 2018). Graph models of indoor environments are highly varied. In the *structure graph* introduced by Roth et al. (1982), nodes represent doors or corners, and edges represent walls of an indoor space (Figure 3b). Deriving the graph representation for navigation purposes from the *structure graph* is challenging due to the lack of explicit definitions of indoor concepts such as containers (*rooms*). Hence, Lee and Kwan (2005) defined the *primal space* and *dual graph* (Figure 3c). *Primal space* in 2D divides one universal polygon into a partition of pairwise disjoint polygons (Ledoux and Gold, 2007, p. 3). *Dual space* assigns a dual vertex to each polygon in the primal space, and if the polygons share an edge, the corresponding dual vertices will be connected by a dual edge. However, if two rooms share their boundary without direct accessibility in the dual graph of an indoor environment, the dual graph still connects them via an edge.

To overcome the accessibility problem in the *dual graph*, *accessibility graph* and *navigation graph* have been suggested (Yang and Worboys, 2015) (Figures 3d and 3e). *Navigation graph* supports room, door, vertical connectiv-

ity, intersection, and landmark. The level of detail of elements modelled in the navigation graph can be used for instruction follower agents. One possible shortcoming is that the navigation graph has no orientation information, while a navigation agent requires orientation guidance.

From a computational perspective, various standard data formats have been provided for indoor space modeling. CityGML³ is a standardization that addresses elements for both outdoor and indoor spaces using 3D coordinates. IndoorGML⁴ is another open data model designed by OGC to represent indoor spaces. IndoorGML supports the definition of indoor space based on geometric information of structure graphs and dual graphs. Industry Foundation Classes (IFC) is an ISO-accepted open standard that can support structure graphs. Geometric and semantic information of the indoor features is defined in the IFC, while the containment relationships of spatial objects can be modelled using the predefined hierarchy, the connectivity of features is not directly defined. Moreover, Indoor Mapping Data Format (IMDF) is a standard that defines indoor spaces based on a hierarchical model that contains venues, buildings, floors, rooms, and objects. The geometric information of features is defined using GeoJSON polygon types in IMDF model⁵. IMDF can support accessibility and navigation graphs.

2.4 Abstract Concepts of Environment

To model indoor spaces symbolically, e.g. in a text-based game, a schema of indoor entities is required. Elements of the schema must be abstract enough to link the computational modeling of the space in a text-based game and real-world indoor space. Elements can be distinguished based on visual and/or functional aspects. Lynch (1964) argues that the operation of an individual in different environments relies on a conceptual *image* of the agent about the environments. An object's *imageability* is defined as "*the quality in a physical object which gives it a high probability of evoking a strong image in any given observer*" (Lynch, 1964, p. 9). *Imageability* is thus reduced to the perceptible visual aspect of physical objects. The socially shared image of an environment then approximates a collection of individuals' images that thus lead to a shared pattern of the environment (Lynch, 1964, p. 40). From the spatial functional perspective, Tomko and Winter (2013) have introduced a formally grounded extension to Lynchian elements that links the dimensionality of city elements and their *accessibility*, i.e., the affordance of spatial objects that specifies if an agent can *enter* the inside of the object.

In a complementary manner, *Image schemata* are a cognitive model that maps observed visual sensory data to a recurring abstract form of knowledge about the world, i.e. patterns (Johnson, 2013). The conceptualization of indoor

³www.opengis.net/doc/IS/CityGML-1/3.0

⁴www.indoorgml.net

⁵<https://apple.com/indoor>

spaces based on image schemata in geographic information systems was introduced by Raubal et al. (1997) and Frank and Raubal (1999), and applied broadly in spatial cognition research (Kuhn, 2007; Rüetschi, 2007; Hedblom et al., 2019).

Rüetschi and Timpf (2004) sketched a formal model of spatial concepts in human wayfinding focusing on qualitative spatial configurations rather than on metric information (the notion of topology defines, metrics refines), which was helpful to avoid issues with the limited device positioning accuracy (if available at all). They extended Johnson's image schemata and defined six concepts based on spatial and functional properties (Table 1).

This study adopts schematic geometry to ground a floorplan in Inform7 concepts. Because schematic geometry is designed for navigation purposes, it is a generic design that enables the implementation of the complexities of indoor spaces. We apply the skeleton graph as the primary space decomposition approach. Skeletonization of the space creates a graph that supports navigation and is flexible in terms of spatial cognition and abstract reasoning (Afyouni et al., 2012) (see Figure 3f). A skeleton graph can model the adjacency and connectivity of nodes (i.e., rooms) connected via edges (i.e., doors), thus fitting with the capabilities of TextWorld. Such skeleton-based approaches fit the case of navigation in narrow corridors (Russo et al., 2014) and support the representation of *natural movement* within indoor environments (Mortari et al., 2019).

3 Conceptual Framework of Indoor Space Concepts

A challenge of text-based games for the simulation of real-world indoor spaces is the constrained domain of concepts and related actions, as well as the lack of support for the agent's movement inside a room (or parts of the room, possibly differentiated by intervisibility), as the concept of more atomic partitions is not defined. We extend TextWorld's knowledge base to cover these shortcomings. The newly defined concepts and their specification are summarized as follows (Table 2 illustrates the defined concepts in an example.):

- **IndoorArea:** The basic new concept for the definition of an atomic indoor space is `IndoorArea`. An `IndoorArea` is *enterable* and affords object containment. An `IndoorArea` is part of a `IndoorRoom`. The agent can interact with the objects inside it if the agent is located in the same `IndoorArea`. An `IndoorArea` can be adjacent with other neighbouring `IndoorAreas`, but the movement between adjacent `IndoorAreas` that belong to the different `IndoorRoom` require explicit actions.
- **IndoorRoom:** The aggregation of `IndoorAreas` form `IndoorRoom`, a container of

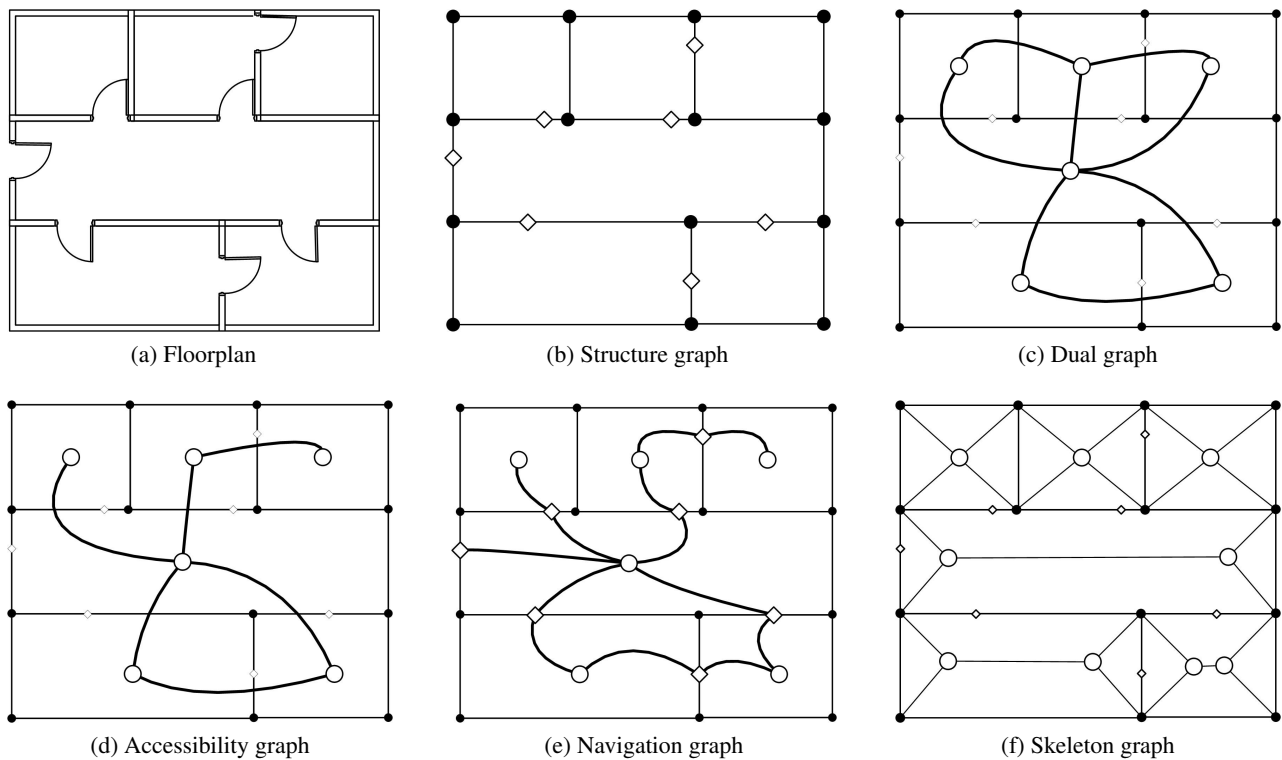


Figure 3. Graphs for modeling indoor spaces. (a) Floorplan containing geometric information of rooms and doors, (b) Structure graph containing coordinates of points of doors and polygons of rooms, (c) Dual graph in which nodes correspond to rooms and edges signify rooms adjacency, (d) Accessibility graph in which nodes correspond to rooms and edges signify doors or gates between adjacent rooms, (e) Navigation graph in which nodes correspond to both rooms and doors and edges represent connectivity relationship, (f) Skeleton graph in which nodes represent intersections and edges correspond to locus equidistant to two points in the boundary.

Table 1. concept of space based on wayfinding (Rüetschi and Timpf, 2004)

element	definition	example
room	enterable and bounded container	kitchen, hallway
region	bounded area or surface affording support	3rd floor, shopping area
collection	spatially or functionally similar things	train station
gateway	a walk-able link that is consciously experienced	door
ulink	unconscious link	open kitchen
item	objects that does not fit to other concepts	landmarks

which boundaries correspond to real-world boundaries such as walls. Every `IndoorRoom` has at least one `IndoorArea`, and every `IndoorArea` relates to only one parent `IndoorRoom`. The `IndoorRoom` is partitioned by `IndoorAreas`. The movement inside `IndoorRoom` does also not require explicit actions, but movement between `IndoorRooms` requires interactions with actions and objects, e.g., a door.

- **IndoorFloor:** The collection of horizontally collocated `IndoorRooms` form `IndoorFloor`. Object containment is not the direct affordance of `IndoorFloors`.
- **Door:** The link that connect two adjacent `IndoorAreas` in two different `IndoorRoom` par-

ents is a `Door`. By definition, doors are located at the boundaries of two `IndoorAreas` and therefore two parent `IndoorRooms`. Doors are *walk-through-able*. Agent must interact explicitly with `Doors` and when they do so, both `IndoorArea` and `IndoorRoom` related to the agent's location will be changed.

- **ULink:** The link that connects two adjacent `IndoorAreas` within the same `IndoorRoom` is a `ULink`. They are *walk-through-able* and the agent experience `ULinks` implicitly. When an agent uses a `ULink`, the `IndoorArea` of the agent changes while the `IndoorRoom` remains the same.
- **Landmark:** All objects located inside an `IndoorArea` are `Landmarks`. They are not

enterable or *walk-through-able*. Landmarks can be visible or invisible, intractable or non-intractable, and movable or fixed. Signage and fixtures are examples of Landmarks. Signage are landmarks fixed in a location that shows an informative text to assist agents' navigation, and agents can only see them without any other further interaction. Fixtures are landmarks fixed in a location that agents can interact with them but cannot move (such as an ATM). A Landmarks can be visible, intractable, and movable, e.g., an apple on a table. Each Landmark belongs to one and only one parent IndoorArea.

All defined concepts must have defined basic properties, including name and description and can have optional properties such as printed name (an alternative, human-consumable label). The extended concept definitions above enable a conceptual model of indoor spaces. Elements of indoor environments with their properties and affordances can be represented through instances of these concepts.

The mapping of abstract concepts of navigation spaces to the extended concepts introduced earlier, and finally to the core text-based game concepts, is shown in Figure 4. Later, we show how a template-based mapping is applied to instantiate these concepts based on floorplan data.

The concept of Room is the basic spatial partition in TextWorld, so we map IndoorArea extracted from skeleton graph into TextWorld's Room. IndoorArea is the basic partition of our defined concepts, so they cannot contain any other concept while they must be contained in at least one IndoorRoom. IndoorRoom maps a real-world room surrounded by physical walls into TextWorld's Rooms that support containment of other Rooms (mapped from IndoorAreas). IndoorFloor maps between horizontally colocated rooms with shared properties using TextWorld Region concept.

Navigating between IndoorAreas might need interaction (e.g., opening a closed door) or be an implicit experience, such as walking through an open gate. Doors map explicit connections and Ulinks map implicit connections. Ulink use the Door of TextWorld concepts which is always open while Door use those openable and closable Textworld Doors.

Landmark map all objects inside IndoorAreas into TextWorld concept of Thing. Things can have different properties, such as portable or fixed, described or undescribed, and intractable or not-interactable.

To automatically map the information derived from the floorplan into TextWorld understandable format, we use templates shown in Table 3. IndoorArea and IndoorRooms use the template of general concepts, as those are new concepts introduced to TextWorld.

Properties schema of new concepts can be defined using the template of the property of concepts, for example, the parent attribute of the IndoorArea, which is an

IndoorRoom. Moreover, the affordances of new concepts must follow the specified template. Visibility, openness, and enterability are examples of concepts' affordances. Concepts can be instantiated by passing a name and a human-consumable description to the template. The properties of instances can also be specified by passing the initialization value.

4 Computational Implementation

The TextWorld engine uses a knowledge base and themes as inputs for a generator that translates the world specifications into Inform7 descriptions. Then, the engine interactively presents the world to the user. This study extends TextWorld's structure (Figure 5) in three domains:

- Defining new concepts (IndoorArea, IndoorRoom, Ulink) based on schematic geometry to ensure the availability of generic concepts allow for capturing the complexities of indoor spaces;
- Extracting abstract objects, instances, properties, and their corresponding descriptions from a geometric floorplan, thus providing a third modality for world creation to the two available currently in TextWorld (random world creation and handcrafted worlds);
- Extending the expressivity of the TextWorld engine by extending direction instructions with egocentric agent navigation concepts, extending the set of available actions and predicates for new concepts, and introducing action aliases.

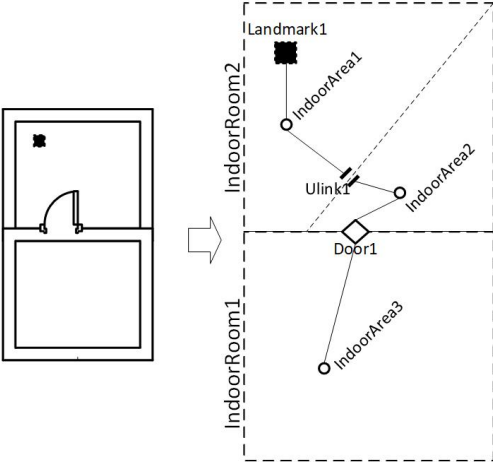
Figure 6 shows the general workflow of the proposed implementation to automatically generate text-based game descriptions from the geometric information of an architectural floorplan. The workflow starts with the geometries of floorplan elements, including rooms, doors, and landmarks, as well as initial information about the Quest, such as the origin and destination location of the agent, the agent's initial orientation, and the maximum number of interactions in a quest (stopping condition).

Next, a geo-processing step recognizes areas and assigns them to corresponding doors and landmarks based on applying the templated instantiations of geometries to text-based game concepts. Finally, descriptions that are understandable for TextWorld is generated and demonstrated.

4.1 Generating Skeleton Graph from Floorplan

The skeleton graph extraction from floorplans relies on medial axis transformation (MAT) (Lee, 1982). The MAT of a polygon is the set of loci from which at least two points on the polygon's boundary are equidistant (Lee, 1982). In other words, skeleton points are an aggregation of circles' centroids inscribed inside a polygon that

Table 2. An example of newly defined concepts in action

Floorplan and Decomposed Floorplan	Concepts	World definition
<p data-bbox="193 443 288 465">Floorplan</p> 	<p data-bbox="694 421 826 443">IndoorRooms</p> <p data-bbox="694 544 815 566">IndoorAreas</p> <p data-bbox="694 656 759 678">Ulinks</p> <p data-bbox="694 723 751 745">Doors</p> <p data-bbox="694 813 799 835">Landmarks</p> <p data-bbox="694 913 820 936">IndoorFloors</p>	<p data-bbox="858 409 1289 465">IndoorRoom1 is an indoor_room. IndoorRoom2 is an indoor_room.</p> <p data-bbox="858 477 1501 533">IndoorArea1 is an area. the parent of the IndoorArea1 is IndoorRoom2.</p> <p data-bbox="858 544 1501 600">IndoorArea2 is an area. the parent of the IndoorArea2 is IndoorRoom2.</p> <p data-bbox="858 611 1501 667">IndoorArea0 is an area. the parent of the IndoorArea0 is IndoorRoom1.</p> <p data-bbox="858 678 1449 734">north of IndoorArea2 is south of IndoorArea1.</p> <p data-bbox="858 745 1449 801">Door1 is a door. "Door Room 1 to Room 2". Door1 is north of IndoorArea3. Door1 is south of IndoorArea2.</p> <p data-bbox="858 813 1517 891">Landmark1 is a landmark. printed name of the Landmark1 is "Landmark 1". the Landmark1 is in IndoorRoom1.</p> <p data-bbox="858 902 1305 969">IndoorFloor1 is a region. IndoorRoom1 is in IndoorFloor1. IndoorRoom2 is in IndoorFloor1.</p>

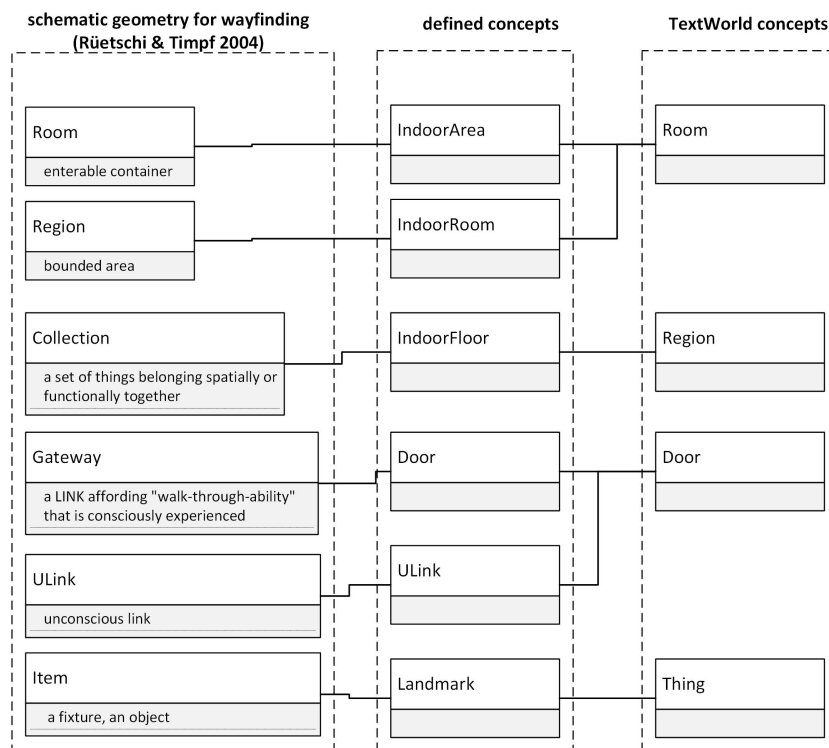


Figure 4. UML of defined concepts and their relation with schematic geometry and text-based games concepts.

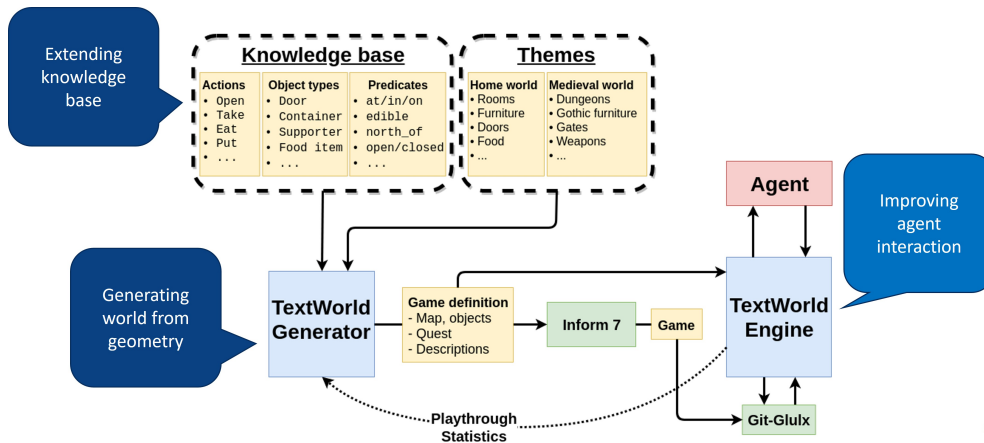


Figure 5. Extended structure of TextWorld, adopted from (Côté et al., 2018)

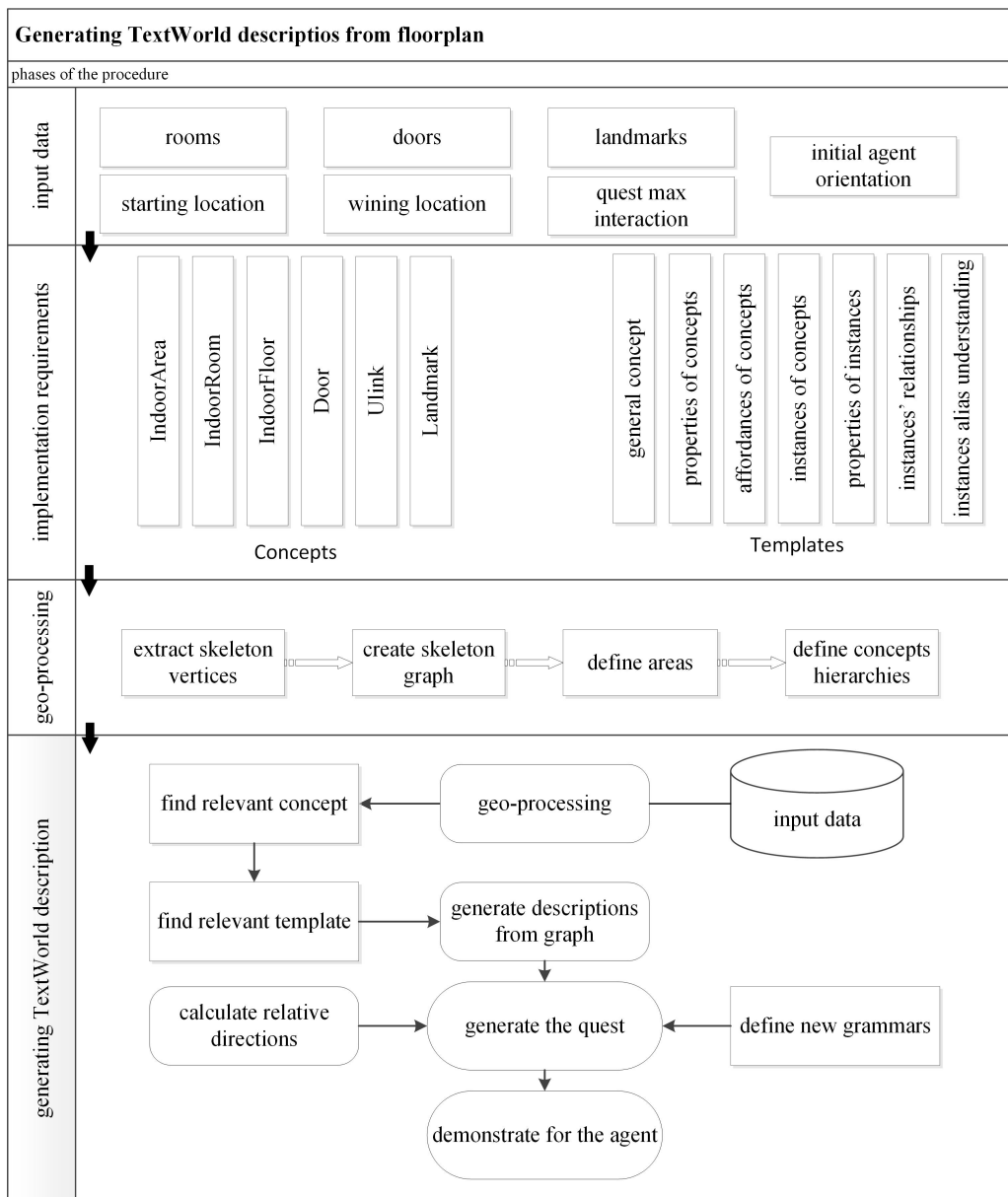


Figure 6. Proposed workflow for automatic text-based games description generation from floorplan

touches the boundary in at least two points. Therefore the circle fitting method is used to compute MAT, and is one of the steps in the computation of the Voronoi diagram. Lee (2004) proposed a complementary algorithm called Straight MAT or SMAT. SMAT solves distortion problems in MAT, including T-shape, L-shape, and X-shape displacements, by simplifying Voronoi diagrams of planar straight-line graphs.

The space decomposition used in this study relies on Lee's algorithm for SMAT calculation implemented in the scikit-geometry Python library⁶. A skeleton graph derived using the SMAT algorithm is shown in Figure 3f for an example floorplan. According to the conceptual design introduced in Section 3, the skeleton nodes represent the *IndoorAreas*. TextWorld environments are symbolic representations and do not include geometric information. Hence, instead of calculating the boundaries of the *IndoorAreas*, only their connections and adjacency are captured here.

4.2 Generating Text-based Games' Environments from Skeleton Graphs

We define dynamic and static templates to frame the automatic generation of text-based games. These templates use the skeleton graph generated based on the floorplan as input and create concepts, instances, relationships, and actions. The outputs are descriptions understandable to Inform7 and TextWorld compilers. Concepts defined in 3, concept properties, affordances, and their default values have been set using a predefined set of templates (Table 3).

The text generation process must enable automatic definition and identification of *Ulink* connections between the atomic *IndoorAreas*, so that the agent can continue exploring in the same room without opening a door. *Door* is the connection between two neighboring *IndoorAreas* located in different rooms. Algorithm 1 summarizes the process of establishing valid connections between neighbouring *IndoorAreas*.

Landmarks are added to the TextWorld descriptions using predefined templates and located at the nearest skeleton node. If the agent's corresponding *IndoorArea* is the same as that of a *Landmark*, the agent can interact with the *Landmark*. However, there can be *Landmarks* that do not afford interaction while they afford visibility to an agent located in a distant *IndoorArea* (i.e., *IndoorArea* that is not the parent of the *Landmark*). These can be identified via the property of the visibility domain of landmarks. The visibility domain can include multiple *IndoorAreas* while the intractability domain can only include the same *IndoorArea* in which the landmark is located.

⁶www.github.com/scikit-geometry

Algorithm 1 Defining containment and connection relationships between *IndoorAreas*

Require: georeferenced floorplan with geometries of rooms' polygons and doors' lines
IndoorAreas = {*IA*₁, *IA*₂, ..., *IA*_{*n*}} ←
 Skeleton(*floorplan*) {extracting *IndoorAreas*}
for *IA*_{*i*} and *IA*_{*j*} in *IndoorAreas* when *i* ≠ *j* **do**
 MinMoves(*IA*_{*i*}, *IA*_{*j*}) ← minimum number of moves required to reach *IA*_{*j*} from *IA*_{*i*}
 if MinMoves(*IA*_{*i*}, *IA*_{*j*}) == 1 **then**
 if ParentRoom(*IA*_{*i*}) == ParentRoom(*IA*_{*j*}) **then**
 define *Ulink* between *IA*_{*i*} and *IA*_{*j*}
 else
 define *Door* between *IA*_{*i*} and *IA*_{*j*}
 end if
 end if
end for

4.3 Extending Agent Interactions

Interactions in TextWorld are supported by grammar relating to new concepts, properties, and actions. Only instructions with known mapping to actions, including operands, are supported. Thus, for direction concepts, it is necessary to identify the actions, i.e., the impact on the agent's state the action will have (e.g., turning), as well as the terms used for issuing these commands (the grammar).

For calculating the agent's orientation, We extended TextWorld to keep track of the history of the agent's positions. Tracking the agent's orientation enables to use of relative movement actions and translates them to cardinal or even the more nuanced eight inter-cardinal (aka ordinal) directions. We define relative direction *RD* as the deviation between the direction in which the agent is currently oriented (looking) and the desired movement direction. Specifically, *RD* is calculated based on the agent's orientation *AO* and the direction of the next room *D_N* (Equation 1). Thus, a direction is expressed internally in TextWorld as an integer number 0-7, starting from North, counterclockwise (Figure 7).

$$RD = (AO - D_N) \bmod 8 \quad (1)$$

Constant tracking of *AO* is handled by the TextWorld engine (that, as shown in Figure 5, by obtaining information from both the agent and the designed game, generates a description for possible actions in each state for the agent) externally from the agent and enables interactions through egocentric navigation commands included in the grammar of actions. The implementation of egocentric navigation commands using the TextWorld engine corresponds to the TextWorld design since, currently, the agents cannot track their trajectories. For example, if the agent is currently looking North (N, *AO* = 0) and communicates with the environment by the `go sharp left` command, the agent will enter the room Southwest (SW) of the current

Table 3. Templates and examples for the automatic generation of concepts, instances, relationships, and actions in text-based games

Definition	Template	Example
general concept	{concept} is a kind of {type}.	area is a kind of room.
properties of concepts	{concept} has a {type} called {variable}.	area has a text called description. area has a indoor_room called parent.
affordances of concepts	{concept} can be {ability}.	area can be enterable.
instances of concepts	{instance} is a {concept}. "{description}".	a0r0 is a area. "An area (0) in [parent]".
properties of instances	the {property} of {instance} is {value}.	the parent of the a0r0 is r0.
connectivity relationship between instances	{relation} of {area1} is {reverse_relation} of {area2}.	north of a0r1 is south of a1r1.
alias understanding	Understand "{reference}" as {variable}.	Understand "Area 0 in Room 0" as a0r0.

room. Details for all possible agent orientations and egocentric navigation commands are illustrated in Figure 8, including the two turn qualifiers, *slight* and *sharp*.

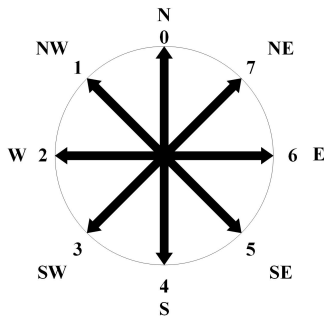


Figure 7. Assigned numbers to cardinal and ordinal directions: N=North, W=West, E=East, S=South, SE=Southeast, SW=Southwest, NE=Northeast, NW=Northwest

The complementary interaction enhancement of the actions proposed here is the definition of extensions to the grammar supporting the actions related to directions. For example, when new actions such as *go slight right* are defined for navigation, the environment must include the corresponding instructions, such as *veer right* or *turn slight right*. To introduce a new grammar component to the TextWorld knowledge base, we use the template *alias understanding* from Table 3. The interaction for the example above would be *Understand "veer right" as going slight right*.

5 Demonstration

We have implemented the proposed workflow to a simple grid floorplan as well as a real-world indoor floor plan (Figure 9). The floorplan, templates, and concepts are integrated in a TextWorld generator program using a Python conversion tool, importing GeoJSON geometries. This GeoJSON represents the floorplan through three separate files, containing simple 2D geometries that capture rooms as closed polygons, doors as points along the edge where two polygons touch, and landmarks as points within these polygons. This is a structure equivalent to the IMDF format (which can capture further, more complex information). The conversion from IMDF or IFC to our model is not covered in this paper.

In this demonstration, the floorplan data of a real-world indoor environment are represented through 10 polygon geometries representing rooms of the environment, 13 point geometries of doors, and 6 point geometries of landmarks. The *ID* of features are assigned randomly while the features' *description* attributes are assigned based on their corresponding role in the environment. Landmarks include ordinary objects and signs showing the agent guiding text. For example, a sign in *Room 8* in Figure 9 guides the agent with a *Exit is in this room!* description. Then, the quest is defined with parameters incl. starting location, destination (winning location), and the maximum number of allowed interactions.

Users can navigate the rooms and areas using the extended TextWorld commands, including egocentric agent orientation. As shown in Table 4, the descriptions of the current state of the agent and the movement options to the immediate next states are provided to the player and agent. Descriptions may contain quest instructions that suggest the agent choose particular actions. For example, when

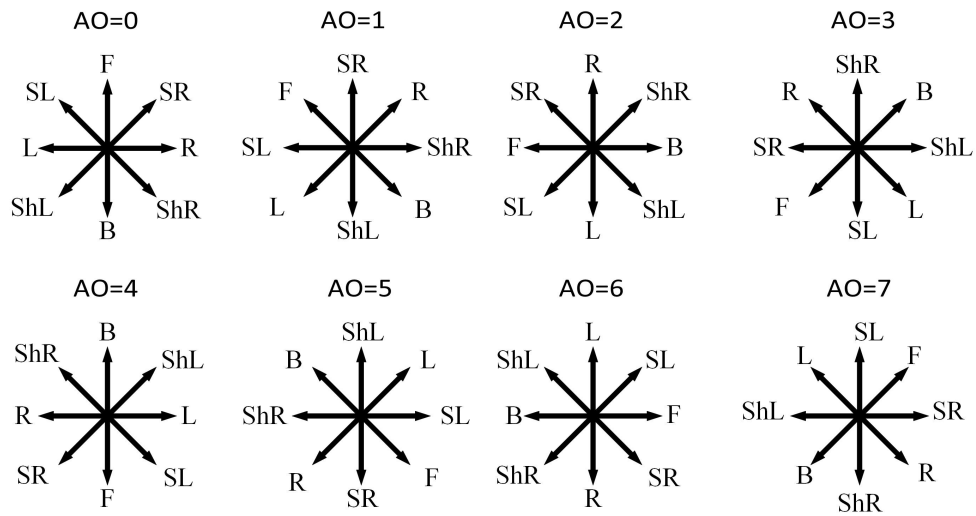


Figure 8. Conversion of allocentric reference frame into egocentric using agent orientation *AO* and next room direction, *AO* - Agent Orientation, *F* - Front, *B* - Back, *R* - Right, *L* - Left, *Sh* - Sharp, *S* - Slight

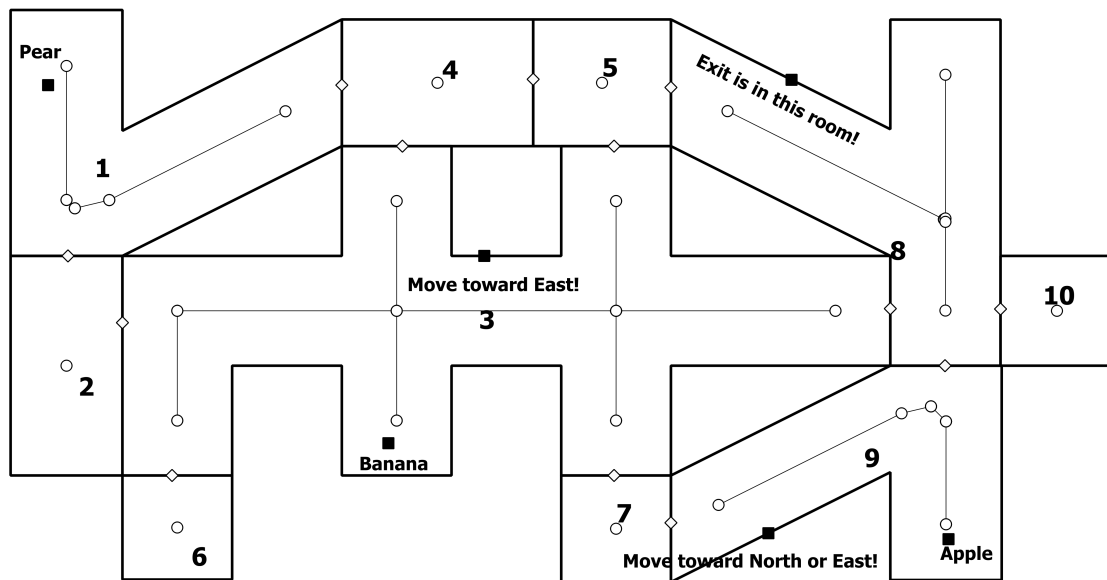


Figure 9. Sample floorplan of the indoor space implemented in TextWorld based on extended concepts and templates. The numbers are the labels of the rooms. Derived from MAT skeleton, nodes are *IndoorAreas*. Green landmarks are navigational signage that guides the agent to find the destination. The agent cannot interact with signage landmarks but can pick up other landmarks, such as Apple.

the agent visits *Room 3* in Table 4, TextWorld will guide the agent by a `Move toward the east!` description that is derived from a sign landmark.

Instruction may be about the immediate action or final goal. For example, in Table 4, lines that explain possible actions and their immediate results, such as `You can continue in the Room 3 by going north (on the left) for moving within a room or (first opening Door 12) You can enter the Room 9 by going south (on the right) for moving between the rooms. However, Move toward the east! or Exit is in this room!` are about the final goal of the designed quest.

In the implemented experiment, the agent starts in *Room 1, IndoorArea1*, which is shown by the point close to *Pear* landmark in Figure 9. The desired destination for the agent is *Room 8*. The agent can explore and receive information about the consequences of immediate actions (e.g., `enter the r9 by going south`) or the desired destination (e.g., `Exit is in this room!`). Agents can choose navigation actions based on the understanding capabilities of the text-based game environment. The examples shown in Table 4 contain allocentric and egocentric navigation instructions. This provides more options for the agent to choose actions in each state (*IndoorArea*).

Agents can interact with landmarks in several ways, such as by opening a box with a `open the wooden box` command, adding a keychain to the inventory with a `take the red key` command, and wearing cloth by `wear the jacket on the table` command. As shown in Table 4, some landmarks can be visible but are too distant to be manipulated by the agent at some *IndoorAreas* – these are so-called *intractable* landmarks. To interact with them, agents must navigate to the same *IndoorArea* of distant landmarks.

It is possible to extend the aliases of the actions in TextWorld so that the agent can communicate with TextWorld by using different commands, i.e., different commands will apply to the same actions. We can use *alias understanding* templates introduced in Table 3 to extend concept aliases (such as `understand "Area 1 in Room 2" as alr2`) that is applicable for translating human-understandable descriptions to TextWorld variables. Moreover, *alias understanding* templates can also be applied to actions (such as `Understand the command "access" as "open"`). The TextWorld aliasing feature helps to introduce synonyms, map commands entered by the agent to actions, or recognize landmarks by their properties (such as `Understand the yellow landmark as bannana`).

The presented work is a novel approach for generating text-based games from real-world indoor environments. By leveraging GeoJSON geometries and extending the TextWorld platform, the proposed workflow enables the generation of games with multiple rooms, doors, and landmarks. The demonstration of the proposed approach on real-world indoor floorplans highlights the potential for its application in domains such as route instruction following. The use of aliases to extend the understanding of concepts and actions further enhances the adaptability of the generated games, allowing for more natural and intuitive interactions. To wrap it up, the proposed workflow and its implementation hold promise for the development of interactive text-based games from real-world environments, opening up new possibilities for text-based agent navigation applications.

6 Data and Software Availability

The Geojson data of indoor environments and codes used in this paper are available from the URL: https://github.com/tomko-lab/Geometry_to_TextWorld. We have used `scikit-geometry`, `scipy.spatial`, `shapely`, and `geojson` Python libraries for geometric calculations and the TextWorld library for description generation.

7 Conclusions and Future Work

In this paper, we have introduced the capabilities of text-based game environments for spatial research, particularly for simulating real-world indoor spaces and interactions in these spaces. After reviewing the challenges of capturing spatial complexities in text-based games, we defined a small set of new abstract concepts based on wayfinding image schemata. We then describe how spatial data from geometric floorplans can be mapped into abstract concepts in text-based games.

We further introduce egocentric agent navigation in TextWorld using relative direction commands. A new grammar is added to the TextWorld engine based on extended navigation concepts to enrich agent interaction modalities.

We demonstrate how this approach enables to import and navigate an indoor space floorplan in TextWorld based on an automatic import procedure. We demonstrate this capability on both real-world and made-up floorplans.

In this paper we did not address the impact of different skeletonization methods like MAT and SMAT on floorplan modeling. Additionally, to make the process fully automated, the input must be in a well-known format, reminiscent of IMDF. In future work, we will address the ability to fully automatically import valid IMDF environments.

In future work, we will focus on the agent-environment interaction aspect of text-based games, as a target environ-

Table 4. Sample agent navigation in TextWorld using the extended egocentric grammar.

```
> go forward

Room 3
An area (2) in r3

Move toward the East!

You can continue in Room 3 by going north (on the left)
You can continue in Room 3 by going south (on the right)
You can continue in Room 3 by going east (at the front)
You can continue in Room 3 by going west (at the back)
Landmark 5 is visible from here, but too far! You can move in this room to examine or
access it
5 minutes have passed, decide well in your future actions, you have limited time (10
minutes)
```

ment for assessing route instruction following capabilities based on deterministic (rule-based) and learning agents navigating in both simulated and real-world environments. There, we will also explore the Inform7 concept of *Person* to model agent affordances. Text-based games provide a flexible and extensible base for novel approaches to spatial language understanding, instruction generation, instruction following, and complex space simulation (e.g., integration of indoor and outdoor spaces).

References

- Adhikari, A., Yuan, X., Côté, M.-A., Zelinka, M., Rondeau, M.-A., Laroche, R., Poupart, P., Tang, J., Trischler, A., and Hamilton, W.: Learning dynamic belief graphs to generalize on text-based games, *Advances in Neural Information Processing Systems*, 33, 3045–3057, 2020.
- Afyouni, I., Ray, C., and Claramunt, C.: Spatial models for context-aware indoor navigation systems: A survey, *Journal of Spatial Information Science*, 1, 85–123, 2012.
- Aikin, J.: The inform 7 handbook, www.musicwords.net/if, 2009.
- Ammanabrolu, P. and Hausknecht, M.: Graph Constrained Reinforcement Learning for Natural Language Action Spaces, in: *International Conference on Learning Representations*, 2020.
- Côté, M.-A., Kádár, A., Yuan, X., Kybartas, B., Barnes, T., Fine, E., Moore, J., Hausknecht, M., Asri, L. E., Adada, M., et al.: Textworld: A learning environment for text-based games, in: *Workshop on Computer Games*, pp. 41–75, Springer, 2018.
- Frank, A. U.: Pragmatic information content—how to measure the information in a route, *Foundations of Geographic Information Science*, p. 47, 2003.
- Frank, A. U. and Raubal, M.: Formal specification of image schemata—a step towards interoperability in geographic information systems, *Spatial Cognition and Computation*, 1, 67–101, 1999.
- Hausknecht, M., Ammanabrolu, P., Côté, M.-A., and Yuan, X.: Interactive fiction games: A colossal adventure, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 7903–7910, 2020.
- Hedblom, M. M., Kutz, O., Peñaloza, R., and Guizzardi, G.: Image schema combinations and complex events, *KI-Künstliche Intelligenz*, 33, 279–291, 2019.
- Jansen, P. A.: A Systematic Survey of Text Worlds as Embodied Natural Language Environments, *arXiv preprint arXiv:2107.04132*, 2021.
- Johnson, M.: *The body in the mind: The bodily basis of meaning, imagination, and reason*, University of Chicago press, 2013.
- Karas, I. R., Batuk, F., Akay, A. E., and Baz, I.: Automatically extracting 3D models and network analysis for indoors, in: *Innovations in 3D Geo Information Systems*, pp. 395–404, Springer, 2006.
- Karimi, H. A.: *Indoor wayfinding and navigation*, CRC Press, 2015.
- Koirikivi, R.: *The architecture and evolution of computer game engines*, Pelimootorit. jultika. oulu. fi, 2015.
- Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Her-rasti, A., Gordon, D., Zhu, Y., Gupta, A., and Farhadi, A.: AI2-THOR: An Interactive 3D Environment for Visual AI, *arXiv preprint arXiv:1712.05474*, 2017.
- Kuhn, W.: An image-schematic account of spatial categories, in: *International Conference on Spatial Information Theory*, pp. 152–168, Springer, 2007.
- Ledoux, H. and Gold, C. M.: Simultaneous storage of primal and dual three-dimensional subdivisions, *Computers, Environment and Urban Systems*, 31, 393–408, 2007.
- Lee, D.-T.: Medial axis transformation of a planar shape, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4, 363–369, 1982.
- Lee, J.: A spatial access-oriented implementation of a 3-D GIS topological data model for urban entities, *GeoInformatica*, 8, 237–264, 2004.
- Lee, J. and Kwan, M.-P.: A combinatorial data model for representing topological relations among 3D geographical features in micro-spatial environments, *International Journal of Geographical Information Science*, 19, 1039–1056, 2005.
- Lynch, K.: *The image of the city*, MIT press, 1964.

- Marshall, S., Gil, J., Kropf, K., Tomko, M., and Figueiredo, L.: Street network studies: from networks to models and their representations, *Networks and Spatial Economics*, pp. 735–749, <https://doi.org/10.1007/s11067-018-9427-9>, 2018.
- Martens, C.: Ceptre: A language for modeling generative interactive systems, in: *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.
- Mortari, F., Clementini, E., Zlatanova, S., and Liu, L.: An indoor navigation model and its network extraction, *Applied Geomatics*, 11, 413–427, 2019.
- Nelson, G.: Natural language, semantic analysis, and interactive fiction, *IF Theory Reader*, 141, 99–104, 2006.
- Nelson, G.: Afterword: Five years later, *IF Theory Reader*, p. 189, 2011.
- Nelson, G. et al.: The z-machine standards document, Version, 1, 22, 1997.
- Paillard, J.: Motor and representational framing of space, *Brain and Space*, pp. 163–182, 1991.
- Raubal, M., Egenhofer, M. J., Pfoser, D., and Tryfona, N.: Structuring space with image schemata: Wayfinding in airports as a case study, in: *International Conference on Spatial Information Theory*, pp. 85–102, Springer, 1997.
- Roth, J., Hashimshony, R., and Wachman, A.: Turning a graph into a rectangular floor plan, *Building and Environment*, 17, 163–173, 1982.
- Rüetschi, U.-J.: Wayfinding in scene space: Modelling transfers in public transport, Ph.D. thesis, University of Zurich, 2007.
- Rüetschi, U.-J. and Timpf, S.: Schematic geometry of public transport spaces for wayfinding, *Geoinformation und Mobilität*, 40, 191–203, 2004.
- Russo, D., Zlatanova, S., and Clementini, E.: Route directions generation using visible landmarks, in: *Proceedings of the sixth ACM SIGSPATIAL international workshop on indoor spatial awareness*, pp. 1–8, 2014.
- Tomko, M. and Winter, S.: Describing the functional spatial structure of urban environments, *Computers, Environment and Urban Systems*, 41, 177–187, 2013.
- Tuli, M., Li, A. C., Vaezipoor, P., Klassen, T. Q., Sanner, S., and McIlraith, S. A.: Instruction Following in Text-Based Games, in: *The Third Wordplay: When Language Meets Games Workshop*, 2022.
- Yang, L. and Worboys, M.: Generation of navigation graphs for indoor space, *International Journal of Geographical Information Science*, 29, 1737–1756, 2015.
- Zhou, Z., Weibel, R., Richter, K.-F., and Huang, H.: HiVG: A hierarchical indoor visibility-based graph for navigation guidance in multi-storey buildings, *Computers, Environment and Urban Systems*, 93, 101 751, 2022.