Approaching Interactions in Agent-Based Modelling with an Affordance Perspective

Franziska Klügl[1](✉) and Sabine Timpf[2]

[1] School of Natural Science and Technology,
Örebro University, SE 70182 Örebro, Sweden
franziska.klugl@oru.se
[2] Institute for Geography, Augsburg University, 86135 Augsburg, Germany
sabine.timpf@geo.uni-augsburg.de

Abstract. Over the last years, the affordance concept has attracted more and more attention in agent-based simulation. Due to its grounding in cognitive science, we assume that it may help a modeller to capture possible interactions in the modelling phase as it can be used to clearly state under which circumstances an agent might execute a particular action with a particular environmental entity.

In this discussion paper we clarify the concept of affordance and introduce a light-weight formalization of the notions in a way appropriate for agent-based simulation modelling. We debate its suitability for capturing interaction compared to other approaches.

1 Introduction

A critical part of building an agent-based model is related to interactions between agents, as well as between agents and other objects in their environment. There is an inherent gap between formulating agents, their properties, individual goals and/or behaviour at the micro level and the overall intended outcome observable at a macro level. When running the simulation, the simulated agents – put together and into an environment –, eventually *generate* this aggregated outcome. Interaction hereby forms the element of the model that connects micro- and macro level. Yet, one cannot easily foresee who will actually interact with whom in the running simulation. Diverse methodologies for developing agent-based simulation models propose different solutions to produce some form of predictability of interactions, defining a systematic approach to formulations in the model.

In this contribution we aim at clarifying the concept of an affordance so that it becomes a helpful notion for general agent-based simulation model development. We suggest a formalization that – if embedded into an appropriate development methodology – can support a more reliable model development by

This paper has already been published in: © Springer International Publishing AG 2017. G. Sukthankar and J. A. Rodriguez-Aguilar (Eds.): AAMAS 2017 Best Papers, LNAI 10642, pp. 222-238, 2017. https://doi.org/10.1007/978-3-319-71682-4_14.

© Springer International Publishing AG, part of Springer Nature 2018
A. El Fallah-Seghrouchni et al. (Eds.): EMAS 2017, LNAI 10738, pp. 21–37, 2018.
https://doi.org/10.1007/978-3-319-91899-0_2

explicitly representing potential interactions. Affordances have been seen as useful in so diverse areas such as Human-Computer Interaction and Virtual Reality, Robotics or spatially explicit agent-based simulation. Our goal is to develop a concept that supports a modeller in capturing interactions, not in a way to be able to automatically reason about them before running a simulation, but in a way to make the modeller aware of the circumstances in which interactions happen or not. Interactions that occur during a simulation run need to be fully explainable. Analysis of interactions that actually happened during the simulation, shall supporting understanding and thus quality control of the simulation model. We argue that affordances – due to their grounding in cognitive science theory – form a natural basis for guiding a modeller.

In the following we first set the scene by discussing how interactions are handled when developing agent-based simulation models, this is followed by a discussion of related work on affordances in agent-based simulation. We then introduce our particular interpretation of the original affordance notion, define affordances for use during simulation runtime and affordance schemata to be specified during model definition. We illustrate how those notions can be applied in a small example. The contribution ends with a discussion of challenges not yet addressed and our future planned work.

2 Formulating Interactions

There are different perspectives that a modeller may consider when developing with an agent-based simulation model. Depending on the particular methodology applied, the set of perspectives is different. Yet, there is a core set containing first, a model of the agents, and second, a model of the simulated environment in which the agents are embedded. The third perspective aims at capturing the interactions between the those elements of the model. A fourth perspective deals with the simulation infrastructure containing information about scheduling updates, time model, et cetera.

The perspective of a single agent is well understood - using techniques and meta-models elaborated in diverse agent architectures such as rule-based systems or BDI agents. The environmental perspective received much attention over the last years, sometimes mixed with the infrastructural elements especially when handling the environments' update from the actions of the agents that should happen in parallel. Yet, the interaction perspective in a general sense appears to be neglected.

The Merriam-Webster dictionary shortly characterizes "interaction" as "mutual or reciprocal action or influence". In addition it distinguishes between two forms of interaction: (1) Interaction as communication and (2) interaction as mutual effect. The first form is often adopted in Agent-Oriented Software Development when specification of agent interaction is reduced to specification of protocols for exchanges of structured messages. The second form is more current. Considering interactions is basically a first step to connect the micro-level behaviour of agents to observations at the system or macro-level.

2.1 Interaction in AOSE

It is not surprising that formulating and dealing with interactions is at the heart of developing and analysing multi-agent systems. Basically from the beginning, researchers analysed conditions and circumstances for interaction of all kinds of agents – simple reactive to intelligent agents. Ferber [8] systematically analysed interaction situations.

Organization models were spotlighted as a mean to structure societies of agents. Hereby, the number of potential interaction partners is restricted to agents within the group or to agents having adopted particular roles. The basic idea behind those endeavours is to make system-level behaviour more predictable. Organizational notions hereby allow determining with whom to interact, while what actually happens during interaction is formulated in a protocols. Many meta-models were proposed for organizational models (such as [14] or [9]).

AUML [4] became the de facto standard for representing agent communication protocols as it provided more flexibility and a higher abstraction level than plain UML sequence diagrams at that time. Meanwhile, the corresponding UML2 diagrams offer similar features [5]. At a higher abstraction level interactions and relations between agents can also be represented using UML Use Case diagrams [7].

2.2 Interaction in Agent-Based Simulation

When specifying a particular behaviour with which an agent interacts or interferes with another entity, it essential to understand in which context the interaction will actually happen during runtime. This reads strange as one may assume that only what is given during modelling, is actually happening during simulation. However, this is just the case in models in which interaction situations are fully given - e.g. the above mentioned pre-defined organizations exactly provide such fully determined place. Yet this is not the case in general. In a simulation with a kind of stigmergic interaction, an agent modifies an environmental entity, another agent perceives the result and reacts to it. Interaction here consists of action and perception in a decoupled way. Who actually reacts to the modification is unclear, when the modeller determines the agent behaviour and potential interaction. Stigmergic interaction may be an extreme example, but similar situations happen in all cases in which the agent behaviour definition contains elements that are determined during runtime – when the agent interacts in a particular situations with the entities that are actually there. This makes it so difficult to handle interaction in agent-based simulation modelling.

Definition and simulation of interactions consequently forms a major source of errors eventually leading to extensive debugging and analysing. Thus, it is a highly critical element of a modelling methodology to get the interactions right as early as possible. Their proper specification, documentation and analysis is essential.

Over the years, several approaches have been published that suggest ways of explicitly handling interactions when creating a model. Like in the AOSE case, particular organization-level models have been proposed, such as using an institutional perspective as in the MAIA methodology [12]. Also integrating a model of social networks, such as discussed in [2] provides a structure who interacts with whom.

As in AOSE, UML Sequence diagrams that enable to formulate an interaction as a sequence of messages, can be used to specify interaction in agent-based simulation models [6]. For a higher abstraction level UML Use Case diagrams may be used. But, the flexible element may concern the interaction partner.

A basic framework for supporting the modelling of interaction is presented with the IODA approach [25]. In their methodology they propose to define interactions explicitly in a way separated for the actual agent models. This is also done when using explicit models of organizations, yet IODA is special as it directly couples interaction to agent action. The central element of IODA is a table that label how agents of one family interact with others. The label connected to a program or script as well as some form of condition. Hereby, Kubera et al. [24] also argue that everything can be an agent; so basically all actions can be phrased as interaction. IODA is particularly apt for reactive agents. It does not cover selection of interaction partners – all agents of a particular type within a specified distance may interact.

In this contribution, we want to explore if the concept of affordances helps to capture possible interactions in the modelling phase. Affordances also could be used to select interaction partners during a running simulation. Before we elaborate on our thoughts, we give an overview on what affordances actually are supposed to be as well as how they are currently used in agent-based simulation.

3 Notions and Usages of Affordances

3.1 The Concept of Affordance

The notion of affordance is at the core of ecological psychology, brought forward by Gibson [13]. Gibson defined affordances as action potentials provided by the environment: “The affordances of the environment are what it offers the animal, what it provides or furnishes, whether for good or ill”. For example, a bench affords sitting to a human. The potential action of ‘sitting’ depends on properties of the bench, properties of the human, and on the current activity the human is engaged in. Gibson put special emphasis on this reciprocity between animal and environment, insisting that affordances are neither objective nor subjective. Thus, Stoffregen [32] defined affordances as “properties of the animal-environment system [...] that do not inhere in either the environment or the animal”.

In the context of cognitive engineering Norman [27] determines the usability of environmental objects for a human carrying out a specific task by considering not only the affordances but the “perceivable affordances” of objects and the ease of perception for humans. Norman is dedicated the designing objects in such a way that their affordances become immediately perceivable by a person engaged in some task. Transferred to the context of modelling interactions a modeller needs to anticipate the affordances that will be needed within a specific action context and that the agent should be able to perceive.

Affordances in robotics reasoning [3] are used for enabling robots to handle unexpected situations. The moment an object is recognized as for example a mug, the robot can retrieve what actions it affords from an object-affordance database. Based on this the robot may adapt its plan to water plants using the mug instead of another container which is not available. This approach requires an extensive database which first needs to be assembled. Raubal and Moratz [30] developed a functional model for affordance based agent, aiming at enabling robots to perceive action-relevant properties of the environment. This research clarifies the notions but stays at an abstract level of formalization. Although they don't name it"affordances" but services, [11] present an idea that is related to both the robotics idea of affordances. They use an action planner to configure a learning scenario an educational game. Appropriate objects providing the services that are needed in the scenario are integrated into the scenario. This is not a simulation application per se, but has some relation.

In Geographic Information Science the affordance concept has been used extensively in order to model and understand human environmental perception and cognition. Jordan et al. [20] created an affordance-based model of place, discovering that the agent, the environment and the task of the agent need to be modelled in order to be able to determine affordances of places. Raubal [29] based his model of wayfinding in airports on an extended concept of affordances, including social and emotional aspects, thus enabling agents to interpret the meaning of environmental entities relevant to the task at hand. Jonietz and Timpf [17,18] interpret affordances as a higher-order property of an agent-environment system, which is determined by agent- and environment-related properties termed capabilities and dispositions at a lower level. As in the previous modelling approaches, affordances are interpreted as properties that may be modelled and not as something that emerges from the interaction between agent and environment. However, the affordance concept emphasizes the central role of action potentials and ties the afforded action and the respective environmental entities in a pragmatic sense [15].

3.2 Affordances in Agent-Based Simulation Modelling

During the last years the concept of affordances has become popular in agent-based simulation. Affordances were basically used to enable a modeller to formulate some element in the simulated environment that the agents could use for deciding about where to go next or what to do next.

There are a number of models that aim at reproducing how a human reasons about its environment for achieving more realism. These models are highly motivated by cognitive science. The basic assumption is that following hypotheses how humans really think, the model can achieve a higher degree of structural validity. Examples for those models are [28–30] or [17]. A formalisation focussing on affordances as an emergent property based on a detailed model of spatially explicit environment as well as actions and relations in that environment can be found in [1].

Other works interpret the notion of affordances more freely: Joo et al. [19] propose affordance-based Finite State Automata. They use affordance-effect pairs to structure the transitions between states of a simulated human. In an evacuation scenario, an agent follows a given route to the exit, but checks every step whether necessary affordances are fulfilled, using affordances to evaluate different local options.

Kapadia et al. [21] use "affordance fields" for representing the suitability of possible actions in a simulation of pedestrian steering and path-planning behaviour. An affordance is hereby a potential steering action. The affordance field is calculated from a combination of multiple fields filled with different kinds of perception data. The agent selects the action with the best value in the affordance field. A particular interesting approach is suggested by Ksontini et al. [23]. They use affordances in traffic simulation denoting virtual lanes as an occupyable space. Agents reason about what behaviour is enabled by the environmental situation. The affordances offered by the environment are explicitly represented by those virtual objects that offer driving on them. [22] labelled environmental entities with "affordances" such as "provides medication" as counterparts of agent needs enabling the agents to flexibly search for interaction partners or destinations.

In these approaches, affordances are used as more as rules, for representing constraints or for identifying options. They serve as a tool for flexibly connecting an agent to interaction partners. There is no intention to advance the research in cognitive science.

3.3 Our Concept of Affordances

Affordances capture an emerging potential for interaction between an agent in a particular mind set intending to carry out a particular action and an environmental entity or ensemble of entities that the intended action involves. The entities need to have specific dispositions that can match up with the capabilities of the agent.

We use "affordance" as a kind of technical term capturing something that would be not be capturable otherwise. We do no claim to formalize the psychological, cognitive-science view on how humans actually reason about affordances. Our focus is on helping the modeller understand and think about interactions between agent and environment. Affordance shall make the potential for interaction between an agent and its environment explicit. So, we let the affordance stand per se for a potential interaction independent of how an agent selects its actions during simulation runtime. One can see it as a "shortcut" for representing what the agent perceives as relevant for selecting an entity as an interaction partner, without explicitly listing relevant features. In Gibson's original affordance idea there is no space for explicit selection between different affordances - the potential for action is directly linked to action in a Boolean fashion.

4 From Affordances to Interaction

Our aim is to deal with interactions in an explicit and flexible way using affordances. Therefore, we need to differentiate between an *affordance* emerging for a simulated agent while it"moves" through its environment, and between a representation that is defined by the modeller as some kind of declarative pattern from which the perception is generated. In principle, we assume that there is something like an explicit, declarative model that the modeller creates which is then interpreted or compiled for actually executing it during simulation runtime when the agent actually "lives". We call the run-time representations "affordance", and the modelling-time representation "affordance schema".

For running the actual Agent-Based Simulation, there must be some process to generate affordances. Theoretically, there is no emergence involved when a simulated agent perceives simulated affordances, as everything is defined by the modeller. Yet, from the point of view of the agent, an affordances may be in deed unexpected. For a modeller an affordance cannot "emerge" surprisingly.

4.1 Affordance and Affordance Schema

We define an affordance as a relation between a potential action and an environmental configuration. So, theoretically, it is neither a part of teh environmet, nor a part of the agent, but connects both. The affordance becomes noticeable by an agent a at a particular time point t during simulation. $pAff_{a,t}$ are all affordances that the agent a can perceive at time t:

$$\langle a, act, x\rangle \in pAff_{a,t} \tag{1}$$

Such an affordance denotes the possibility of establishing a relation between an agent a and an entity x with respect to action act. x may serve as an interaction partner, if the given action is executed. With the perception of the affordance, the action becomes possible. Both, a and x are in a particular state at the time point t. We apply an extended view on "state" that goes beyond pure representation of kind-of metabolic values, but also contains activity, motivations and goals, beliefs, etc. We do not make assumptions on how this state looks like in a particular model. We also need to assume that the agent a has an explicit set of distinct, potential actions from which it selects one to perform in its environment.

As given in the previous section, an affordance links a potential action to an environmental *constellation*. Per se, such a constellation is not just a single entity in a particular state, but contains context. For example, a bench affords sitting-down just if the area to sit on is sufficiently stable (state of the bench entity). Selection is influenced by the context of the entity - whether it is below a tree casting shadow on it during sunny, too hot times or under a roof that protects it from rain on a rainy day. We assume that all information that qualifies an entity for offering a particular potential action is represented in its state; information that makes it more or less qualified in comparison to other entities affording the same potential action is determined by its context. Preferences or degrees of qualification are not considered in the classical affordance concept. In [16] this classical view is extended by elaborating gradation of affordances. Nevertheless, the selection of affordances depends on the particular way the agent reasons about affordances which should be independent from the actual affordance. An agent might follow the first affordance relation that it encounters or a random one or might evaluate different options for determining which one to prefer.

This formalization is different from [31] who see an affordance as an acquired relation between a combination of an environmental object with behaviour and an effect of this behaviour. The idea of acquiring knowledge about an affordance illustrates their robotics perspective.

Thus, we image an affordance as an explicit object (as a kind of data structure) during simulation runtime about which the agent reasons with respect to carrying it out or not. Thus, there is a need for an higher-level data structure or schema that enables to create such runtime affordance objects. Such a schema must be more than a class in the object-oriented sense from which affordance instances can be created. For being useful in modelling per se, the schema needs to contain more contextually relevant information and conditions under which the affordance actually “emerges”. We define such an *affordance schema* in the following way:

$$\langle AType, act, EType, hContext, sContext \rangle \tag{2}$$

Such an *affordance schema* can be seen as a “pattern” that can be used to generate or determine affordances present in the agents environment[1]. An affordance schema is specified during modelling, but does not necessarily exist as an explicit data structure during simulation runtime. When a modeller specifies such affordance schemata, she explicitly writes down under which circumstances an interaction might happen between an agent of type $AType$ performing action act with an object of type $EType$. The action in the affordance can be a more specific and parametrized version of the action given in the affordance schema. The actual action representation may depend on the applied agent architecture. The fourth and fifth elements $hContext$ and $sContext$ capture the circumstances under which an affordance $\langle a, act, x \rangle$ can be really created for a being a kind Of $AType$ and x being a kind Of $EType$, offering the action. The difference between $hContext$ and $sContext$ is that the former contains hard conditions that enable the affordance - focussing on object and agent properties directly; the latter contains weaker conditions or even just criteria that make a particular constellation more favourable than others.

For example, in a park simulation (such as [33]), during a hot day, an agent a enters a park that is equipped with a currently broken bench $b1$ in the shadow, a clean bench $b2$ in the sun and a nice looking stone $st1$ under shady trees. The agent a entering the park is tired and searches for a place

[1] Our idea of an affordance schema is on a higher abstraction level than what W. Kuhn called “Image Schema” in [26]. He describes an environmental constellation using spatial categories and connects them to a process that they afford.

to sit down (action *sitDown*). Thus, a scans the environment using the affordance schema for *sitDown* and generates the following affordance objects: $\langle a, sitDown, b2\rangle$ and $\langle a, sitDown, st1\rangle$. It does not generate an affordance for $b1$ because the bench is not apt for sitting on it due to the broken surface. Depending on some form of ontology capturing environmental entities such as benches or stones as entities with flat surfaces, the modeller has defined the following affordance schema as a pattern to describe potential interactions: $\langle Visitor, sitDown, ObjectWithFlatSurface, hConditions, sConditions\rangle$. Agent a is of type $Visitor$ and requires for the action *sitDown* an entity of type $EntityWithFlatSurface$ which is the superclass of benches, stones, etc. However, not every one of those entities affords the action for a $Visitor$ agent. This is represented in $hConditions$ which define under which circumstances the environmental entity e affords the action *sitDown*: {*stable(surface(e)), height(surface(e))* < *110 cm), ...*}. The set of soft conditions may contain for example $\{inShadow(e)\}$. The conditions may also refer to other objects present in the vicinity of e affecting whether and how well the entity e actually can afford the given action.

It is important to stress that our definition of affordance and affordance schema does not contain a description of the effect of the action it refers to. The description as in the example just contained a label *sitDown*. What this means. In simulation, we are dealing with an environment for the agents' behaviour that is part of the model - that means fully defined by the modeller. With the environment the effect of actions is usually fully defined, even if a modeller follows the conceptually cleaner distinction between agent action and environmental reaction as described by [10].

Another essential aspect distinguishing the specification of affordances in agent-based simulation versus robotics (and also agent-oriented software development) is that actions in simulation may be defined at arbitrary levels of abstraction – adapted to the abstraction level of the environment. For example the action *sitDown* may not have a lower level correspondence when the agent executes it. There might be no going towards, arching joints, lowering backs, or whatever low-level commands are necessary to execute such an action. Abstraction levels might differ a lot between models describing the same phenomenon. This is also the reason why we would not expect to be able to create a set of "standard" affordances that can be used across many simulation applications.

Enabling a distinction between different environmental objects so that the agent may prefer one to another is NOT part of the original affordance idea. An affordance connects an environmental object to a potential action of the agent. How the agent reasons is not part of the affordance. Thus, conditions are intentionally only present on the affordance schema, i.e., the modelling level. The runtime affordance depends on the simulated agents' point of view within the simulated environment. But somehow during simulation, there must be a process generating the affordances that the agent then can select, etc. Thus, we need to discuss processes of how the affordance schemata generate affordances and determine the agents' actual behaviour and interaction.

4.2 Processes Around Affordances

In theory, an affordance *emerges* as a potential action for an actor with a particular motivation (goal, desire ...). In a simulation, it needs to be determined either by the simulated agent itself or by some higher level process which may not be manifested as an actor in the simulation. In Fig. 1 an abstract view is visualized of how different elements of such a process can be connected.

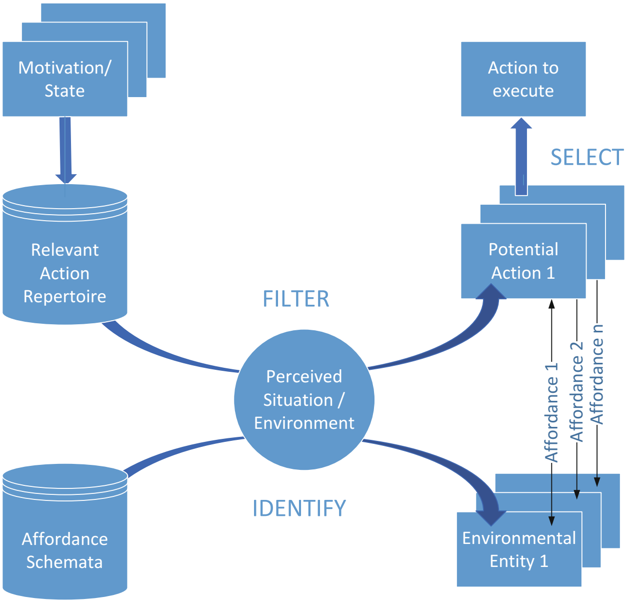

Fig. 1. Overview over processes related to affordance generation and usage.

Following Fig. 1 we need to elaborate partial processes relevant for creating interactive agent behaviour from explicitly defined affordances:

- Mechanism that connects agent goals to actions that are apt to achieve the agents' goal or satisfy its motivation. This process element is responsible for a pre-selection of actions from an action repertoire capturing what the agent is able to do in general. The selected actions need to be connected to the agents' motivational concepts and perceptions/beliefs in a classical way: the agent shall not select actions that it believes not to work in a particular environment, etc.
- Potential actions are filtered based on the environment checking whether the prerequisites for the actions are fulfilled or not. This is done by doing some kind of "pattern matching" of affordance schemata to perceived environment. This connects potential actions to environmental entities by generating (identifying) affordance relations.

- Having established a set of realizable actions with potential interaction partners, the agent can use the affordances connecting actions and entities to evaluate which of the combinations are the preferable ones. Such a preference relation between affordances (based on an evaluation of the context information given on the modelling level) is then used to select the action that is executed.

5 Illustrative Example: A Supermarket

Consider the following situation and process: The agent A has collected a number of items in a supermarket and moves towards the cash points to pay. The agent has sufficient money (in cash or on/with card). There exist two manned cash points as well as one self-payment counter machine: {*CashierRight, CashierMiddle, AutoCashier*}. Of the two manned cash points, only $CashierRight$ is actually busy. *CashierMiddle* misses the cashier agent. Each of the working cash points has a queue of agents waiting for their turn: In front of $CashierRight$ there are 4 persons queuing up, in front of $AutoCashier$ only another person is waiting for the current person to finish. So it is highly probably that A will be served earlier when queueing up at the electronic cash point. A strongly prefers to interact with humans, yet is under time pressure.

5.1 Description of Interaction with Affordances

When A approaches the cash point area with the intention of doing the action $Pay2Leave$, A perceives the three cashiers and immediate sees that only two are available for the intended action.

$$\langle A, Pay2Leave, CashierRight, Cond_{CashierRight,now}\rangle$$
$$\langle A, Pay2Leave, AutoCashier, Cond_{AutoCashier,now}\rangle$$

with $Cond_{CashierRight,now}$ = {*queue(CashierRight, 4), female(CashierRight), young(CashierRight), friendly(CashierRight)*} describing the configuration of the particular cashpoint at time *now*. The configuration of $AutoCashier$ is $Cond_{AutoCashier,now} = (queue(AutoCashier, 1))$.

There is no affordance for $CashierMiddle$ as it is actually not working due to the missing cashier. Both affordances have particular properties that describe the current configuration the agent evaluates for making a decision for one of the interaction partner $CashierRight$ or $AutoCashier$. The agent needs to evaluate whether it prefers to wait for the interaction with a nice human cashier or wants to go for the faster automated way. The $Pay2Leave$ action may have a particular implementation for each of the interaction partners specified as a communication protocol as given below.

While this describes a simulation run-time situation, the modeller defines affordances schemata to specify the interaction. In this example case, the relevant affordance schema may look like that:

$$\langle SHOPPER, Pay2Leave, CASHpOINT, Prereq, PrefCriteria\rangle$$

The affordance schema contains the following elements: first, a combination of agent type $SHOPPER$ and a particular action/activity that the agent wants to do: *Pay2Leave*. Hereby, $Pay2Leave \in ActionRepertoire(SHOPPER)$, that means the action must be part of the default – as defined on the class level – action repertoire of any shopper agent. A as an instance of a $SHOPPER$ has this action in its action repertoire. $CASHPOINT$ is an abstract class from which the classes of $MANNED{-}CASHPOINT$ AND $AUTOMATED{-}CASHPOINT$ are inheriting. Both types can afford the *Pay2Leave* action of the shopping agent. Yet, additional conditions must be fulfilled. These conditions and criteria depend on the concrete type of $CASHPOINT$ (see Table 1).

Table 1. Prerequisites and conditions in cashier scenario

AType	Conditions	Preference criteria
$MANNED - CASHPOINT$	manned(C)	Queue, Friendliness ...
$AUTOMATED - CASHPOINT$	functioning(C), available(A,card)	Queue

For achieving a fully functional model clearly a lot of elements are missing. We just focus on a small number of potential interactions. Additional interactions could be between A and diverse products that A wants to buy. Hereby each product affords to be taken and put into the cart. Before we continue discussing our approach, we have a look how the corresponding formulations would look like when using ways of specification as introduced in Sect. 2.

5.2 Description of Interaction with IODA or MAIA

In the following we intend to give a general impression of two rather extreme alternatives to formulate interactions: IODA [25] aiming more at simulation of emergent phenomena and MAIA [12] following an explicit organization-oriented approach. We do neither give full models, nor the does our description describe exactly the same part of the model. Thus, a lot of context is missing which would be necessary to precisely apply these two methodologies for developing agent-based simulations.

The central element of designing this scenario with IODA [25] is to define the interaction matrix as shown in Table 2.

The table specifies that elements of one agent "family" interact with an element of another. For example, a Customer agent may initiate an interaction with a Cashier agent, if the distance between them is lower than 3 units. "Pay&Pack" is hereby a label for a sequence of actions describing the actions of the involved entities during the interaction. The model specifies what happens during an interaction, under which circumstances the interaction is triggered and what type of agents are involved. What is actually done is represented as a sequence of actions executed by the contributing agents. How the actual interaction is

Table 2. Raw Interaction Matrix in the supermarket scenario following IODA [25]. In following the steps of the overall methodology, interactions would be detailed and selection process is specified, etc.

Source	Target			
	Shelf	Customer	Employee	Cashier
Customer	TakeGoods (d = 0)	WaitBehind (d = 2)	AskForHelp (d = 2)	Pay&Pack (d = 3)
Employee	ReFill (d = 0)			
Cashier		RequestPayment (d = 3)		

selected is determined by the actual agent architecture. Initially, Kubera et al. assumed reactive agents, that means agents that more or less directly connect perception to action without reasoning about explicit representations of agent goals.

As introduced above, MAIA [12] forms a framework for agent-based simulation based on the formalization of a particular organizational model. It is especially apt for social models.

In a simulation reproducing how humans behave in a supermarket, one may assume two types of agents: Customers as individual actors and the supermarket as a composite actor, bringing together all its employees that temporally take over a particular role, such as `ReFiller` or `Cashier`. Agents may have particular attributes, such as contents of the shopping chart or entries on the shopping list. The roles have an associated objective, such as acquire and pay all items on the shopping list. There may be dependencies between roles based on dependencies between objectives - captured also in institutional settings. When adopting a role, an agent also gets capabilities that basically correspond to possible activities or actions that the agent with that role is able/permitted to perform. For the specification of interactions the set of rules and conventions that govern agent behaviour to be specified by institutional statements is particularly interesting. There are different types of those statements for describing which behaviour can be expected by an agent and what happens if the agent does not fulfil the expectations: not following a `rule` results in sanctions, a `norm` is behaviour without sanctions if not followed. The weakest notion of an institutional statement is `shared strategy`. In Table 3 we give a few examples of institutional statements of a customer actor in the supermarket scenario.

These elements set up the constitutional structure of the model. In addition, the modeller needs to specify the physical context (environmental model) and the operational environment, which describes how an agent influences the overall system state. A simulation has an *action arena* which contains so called *action situations*. The latter basically describes some kind of plan structure organizing atomic actions in an institutional context for an agent exhibiting a particular role. Interactions between different agents takes place within an entity actions.

Table 3. Institutional Statements defining the expectations on how customer agents behave

Type of statement	Statement
Rule	A customer always has to pay the goods before leaving
Norm	A customer has to wait in line behind earlier customers
Shared strategy	Customers start to pack directly after the cashier accounted for a good
...	...

So, actually what happens during interaction is hidden quite deeply in the overall model specification.

In Sect. 2 we mentioned that UML can be used to formulate interactions. How this could look like at a rather high abstraction level is shown in Fig. 2.

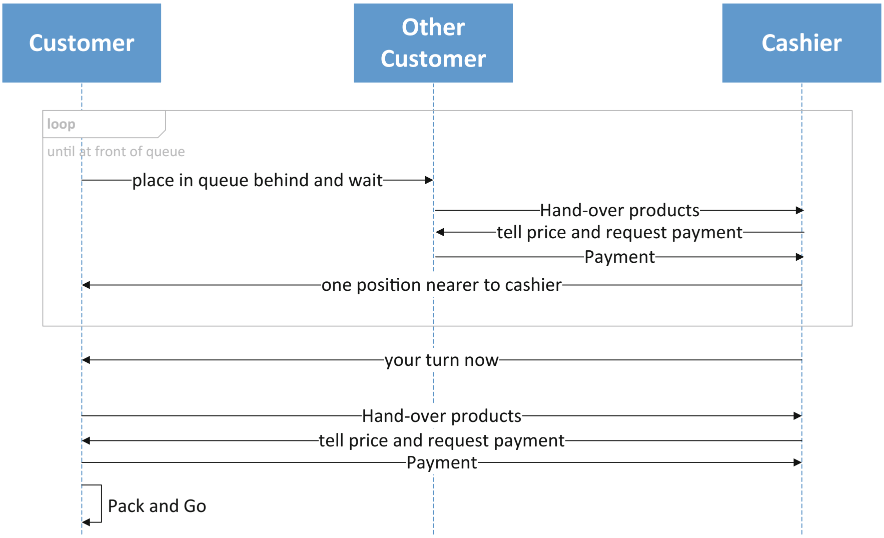

Fig. 2. Protocol-like definition of interactions between waiting customers and the cashier handling one after the other.

One can see that the different frameworks and approaches actually focus on different problems. Our affordance/affordance schema concepts actually concentrate on the selection of the interaction partner in a more flexible, yet less predictable way as in more organization-oriented approaches. One may interpret it as more specific and apt for agents that actually reason about their next action than in the IODA methodododology.

6 Discussion and Conclusion

In this contribution we clarified the notion of affordances and introduced affordance schemata showing that such a distinction is necessary when distinguishing between what happens during simulation runtime and what a modeller explicitly formulates. We put those concepts into an interaction modelling context. The questions remain whether these concepts can be really useful, what to do such that they become useful and how to evaluate their usefulness? The current stage of our research is quite preliminary, as we first wanted to clearly agree on what we actually model when specifying affordances. The current contribution thus cannot be more than a discussion paper. For creating a methodology we would need to make assumptions on meta-models formulating a context such that we can formalize every detail necessary to fully support the complete modelling and simulation process. Based on such a meta-model we could then create tools that directly support modelling - and as we explicitly approach interactions hopefully support model analysis in an improved way. However, we are not sure whether yet another methodology provides a good idea. What we actually want to achieve is to propose a suitable language that supports a modeller when formulating interactions. It should help the modeller to stay aware of when, if and under which circumstances interactions happen and which agents with which particular features participate in the interaction.

References

1. Afoutni, Z., Courdier, R., Guerrin, F.: A multiagent system to model human action based on the concept of affordance. In: 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH), 8 p. (2014)
2. Amblard, F., Bouadjio-Boulic, A., Gutiérrez, C.S., Gaudou, B.: Which models are used in social simulation to generate social networks? A review of 17 years of publications in JASS. In: 2015 Winter Simulation Conference (WSC), pp. 4021–4032 (2015)
3. Awaad, I., Kretschmar, G., Hertzberg, J.: Finding ways to get the job done: an affordance-based approach. In: 24th International Conference on Automated Planning and Scheduling (ICAPS 2014), Portsmouth, USA, June 2014, pp. 499–503 (2014)
4. Bauer, B., Müller, J.P., Odell, J.: Agent UML: a formalism for specifying multiagent software systems. In: Ciancarini, P., Wooldridge, M.J. (eds.) AOSE 2000. LNCS, vol. 1957, pp. 91–103. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44564-1_6
5. Bauer, B., Odell, J.: Uml 2.0 and agents: how to build agent-based systems with the new UML standard. Eng. Appl. Artif. Intell. **18**(2), 141–157 (2005)
6. Bersini, H.: UML for ABM. J. Artif. Soc. Soc. Simul. **15**(1), 9 (2012)
7. Chella, A., Cossentino, M., Faso, U.L.: Applying UML use case diagrams to agents. In: Proceedings of the AI*IA 2000 Conference, pp. 13–15 (2000)
8. Ferber, J.: Multi-Agent System: An Introduction to Distributed Artificial Intelligence. Addison Wesley Longman, Boston (1999)

9. Ferber, J., Gutknecht, O.: A meta-model for the analysis and design of organizations in multi-agent systems. In: Proceedings of the 3rd International Conference on Multi Agent Systems (ICMAS1998), Washington, DC, USA, pp. 128–135. IEEE (1998)
10. Ferber, J., Müller, J.-P.: Influences and reactions: a model of situated multiagent systems. In: Proceedings of the ICMAS'96. AAAI Press (1996)
11. Ferdinandus, G.R., Peeters, M., van den Bosch, K., Meyer, J.-J.C.: Automated scenario generation - coupling planning techniques with smart objects. In: Proceedings of the 5th International Conference on Computer Supported Education, Aachen, Germany, pp. 76–81 (2013)
12. Ghorbani, A., Bots, P., Dignum, V., Dijkema, G.: MAIA: a framework for developing agent-based social simulations. J. Artif. Soc. Soc. Simul. **16**(2), 9 (2013)
13. Gibson, J.J.: The Ecological Approach to Visual Perception. Houghton Mifflin, Boston (1979)
14. Hübner, J., Sichman, J.F., Boissier, B.: Developing organised multi-agent systems using the MOISE+ model: programming issues at the system and agent levels. Int. J. Agent-Oriented Soft. Eng. **1**(3/4), 370–395 (2007)
15. Janowicz, K., Scheider, S., Pehle, T., Hart, G.: Geospatial semantics and linked spatiotemporal data - past, present, and future. Semant. Web **3**(4), 321–332 (2012)
16. Jonietz, D.: From space to place - a computational model of functional place. University of Augsburg, Geographical Information Science, Thesis (2016)
17. Jonietz, D., Timpf, S.: An affordance-based simulation framework for assessing spatial suitability. In: Tenbrink, T., Stell, J., Galton, A., Wood, Z. (eds.) COSIT 2013. LNCS, vol. 8116, pp. 169–184. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-01790-7_10
18. Jonietz, D., Timpf, S.: On the relevance of Gibson's affordance concept for geographical information science. Cogn. Process. Int. Q. Cogn. Sci. **16**(1(Suppl.)), 265–269 (2015)
19. Joo, J., Kim, N., Wysk, R.A., Rothrock, L., Son, Y.-J., Oh, Y.-G., Lee, S.: Agent-based simulation of affordance-based human behaviors in emergency evacuation. Simul. Model. Pract. Theor. **13**, 99–115 (2013)
20. Jordan, T., Raubal, M., Gartrell, B., Egenhöfer, M.J.: An affordance-based model of place in GIS. In: Poiker, T., Chrisman, N. (eds.) Proceedings of the 8th International Symposium on Spatial Data Handling, Vancouver, CA, pp. 98–109 (1998)
21. Kapadia, M., Singh, S., Hewlett, W., Faloutsos, P.: Egocentric affordance fields in pedestrian steering. In: Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games (I3D '09), pp. 15–223, New York, NY, USA. ACM (2009)
22. Klügl, F.: Using the affordance concept for model design in agent-based simulation. Ann. Math. Artif. Intell. **78**, 21–44 (2016)
23. Ksontini, F., Mandiau, R., Guessoum, Z., Espié, S.: Affordance-based agent model for traffic simulation. J. Auton. Agents Multiagent Syst. **29**(5), 821–849 (2015)
24. Kubera, Y., Mathieu, P., Picault, S.: Everything can be agent! In: van der Hoek, W., Kaminka, G., Lespérance, Y., Luck, M., Sen, S. (eds.) Proceedings of 9th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), 2010, Toronto, Canada, p. 1547f (2010)
25. Kubera, Y., Mathieu, P., Picault, S.: IODA: an interaction-oriented approach for multi-agent based simulations. Auton. Agent. Multi-Agent Syst. **23**(3), 303–343 (2011)
26. Kuhn, W.: An image-schematic account of spatial categories. In: Winter, S., Duckham, M., Kulik, L., Kuipers, B. (eds.) COSIT 2007. LNCS, vol. 4736, pp. 152–168. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74788-8_10

27. Norman, D.A.: The Invisible Computer. MIT Press, Cambridge (1999)
28. Paris, S., Donikian, S.: Activity-driven populace: a cognitive approach to crowd simulation. IEEE Comput. Graph. Appl. **29**(4), 34–43 (2009)
29. Raubal, M.: Ontology and epistemology for agent-based wayfinding simulation. Int. J. Geogr. Inf. Sci. **15**, 653–665 (2001)
30. Raubal, M., Moratz, R.: A functional model for affordance-based agents. In: Rome, E., Hertzberg, J., Dorffner, G. (eds.) Towards Affordance-Based Robot Control. LNCS (LNAI), vol. 4760, pp. 91–105. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-77915-5_7
31. Şahin, E., Çakmak, M., Doğar, M.R., Uğur, E., Üçoluk, G.: To afford or not to afford: a new formalism of affordances towards affordance-based robot control. Adapt. Behav. **15**(4), 447–472 (2007)
32. Stoffregen, T.: Affordances as properties of the animal environment system. Ecol. Psychol. **15**(2), 115–134 (2003)
33. Timpf, S.: Simulating place selection in urban public parks. In: International Workshop on Social Space and Geographic Space, SGS 2007, Melbourne (2007)