Dynamic Context-Sensitive Deliberation*

Maarten Jensen[1], Lois Vanhee[1], and Frank Dignum[1]

Umeå University, Umeå, Sweden

Abstract. Truly realistic models for policy making require multiple aspects of life, realistic social behavior and the ability to simulate millions of agents. Current state of the art Agent-based models only achieve two of these requirements. Models that prioritize realistic social behavior are not easily scalable because the complex deliberation takes into account all information available at each time step for each agent. Our framework uses context to considerably narrow down the information that has to be considered. A key property of the framework is that it can dynamically slide between fast deliberation and complex deliberation. Context is expanded based on necessity. We introduce the elements of the framework, describe the architecture and show a proof-of-concept implementation. We give first steps towards validation using this implementation.

Keywords: Context deliberation · Scalability · Realism · Social agents

1 Introduction

Models for policy making require multiple aspects of life, realistic social behavior and scalability by being able to simulate up to millions of agents. State of the art Agent-based models contain multiple aspects of life, however struggle incorporating realism and scalability in the same model. We propose a Dynamic Context-Sensitive Deliberation (DCSD) framework that dynamically slides between fast deliberation (scalable) and complex deliberation (retains realism). Context is explored based on necessity rather than determined beforehand. The framework's potential is shown using a proof-of-concept implementation.

During the COVID-19 pandemic many simulation models were created [18]. Most models were aimed to provide insights on the potential effects of policies. To realistically model the pandemic these models needed to incorporated many aspects of life. For example contagiousness, people, public transport, social life, health care, etc. Probabilistic agent models such as the Oxford model [12] were able to incorporate many aspects of life and simulate millions of agents. The advantage of these models is scalability, however the behavior of these agents lacks realism. An agent spending one day in quarantine would have the same probability leaving quarantine as an agent spending one year in quarantine. Adding additional probabilities to simulate adaptive behavior is possible. However these additional probabilities would be hard to estimate, as data on future behavior

* Supported by WASP - Wallenberg AI, Autonomous Systems and Software Program

of people is not available. In contrast there are social theory based models. The Agent-based Social Simulation of the Coronavirus Crisis (ASSOCC) model [6] uses a complex need-based deliberation model that considers more aspects of life [14]. In this model staying in quarantine for a prolonged amount affects the needs of the agents. The longer the agents stay in quarantine the lower some of the needs become making the agents more likely to break quarantine. This realistic behavior comes at a cost as considering all information slowed down deliberation significantly. This limited the number of agents to about 2000 and made expansion of the model no longer practically possible. The complex deliberation is the bottleneck, making the model realistic though not scalable. Balancing complexity and scalability requires a different approach.

Our proposal is a deliberation system that is simple in principle, but uses complexity by necessity. This is inspired by work by Kahnemann [16], who suggests humans use quick and fast thinking most of the times, however when necessary use more complex decision making drawing in more information. Kahnemann suggests two distinct systems, i.e. fast and slow. We suggest a dynamic system that gradually slides from fast to more complex deliberation based on necessity. Determining how and when to increase complexity is dependent on context. Context needs to explicitly be taken into account in the framework.

The most naive approach of taking context explicitly into account is determining the complete context before hand. While practical in a simple agent model. This approach is not practical in a complex model such as ASSOCC [14]. This model has millions of states, each a different context. Tying each of these contexts to a specific deliberation will be practically impossible. The model by Edmonds [7] circumvents this problem since it uses a learning rule to generate the context. Even though this system could be efficient in its deliberation, the context determination becomes the bottleneck. Another interesting model is the Consumat model by Jager [13]. This model is able to change its type of deliberation based on the situation. However the criteria to change deliberation is not context dependent but rather based on a utility value.

In this paper we propose a Dynamic Context-Sensitive Deliberation framework. The framework deliberates at a more abstract level, the meta-level, to be able to consider all types of contextual information. The meta-level is defined by a *tuple* consisting of activities, goals, plans and actions. The framework dynamically manipulates the tuple drawing in information from the simulation. This process is continued until an action has been found. The framework is partially implemented in a simulation that serves as a proof-of-concept. This proof-of-concept shows how the framework can potentially achieve scalability without loosing realism for simulations that model daily life for policy makers.

2 Background

Context for software systems has been studied before. For example Zimmerman [21] gives a categorization of context. Context is split into five categories: time, location, activity, individuality and relations. This work is based upon an earlier definition by Dey [5], that is *'Context is any information that can be used to characterize the situation of an entity'*. This definition is meant for context of a system interacting with a user. As the definition indicates any information can be relevant. For example in a house, objects such as: chairs, mats, tables, people, and mugs; but also every property of those objects: color, weight, shape, and durability. This type of context is basically infinite. The context for agent deliberation within a simulation is however different. Rather than context in a system, it is context of an agent within a system (the simulation). For example a house could be implemented as an object that contains a location, can hold a number of agents and stores food. Meaning those objects are the only contextual information that can be considered from the house. The context within the simulation is therefore limited by the implementation. All the information is readily available.

While simulation context is discrete and easily accessible, this does not mean the problem of context determination is solved. In a simulation with that many daily life aspects as ASSOCC the relevant information still has to be selected during the deliberation process. A starting point could be to discern the information needed for different deliberation contexts. If an agent follows a habit, for example a morning routine, relevant information could be: the bed, clothes, shower, and breakfast. While an agent that is deciding what activity to do with friends, will probably consider the preferences of those friends (theory of mind).

2.1 Information Relevance in Decision Contexts

Earlier work suggests which information can be relevant in different deliberation contexts. This work is based on the Contextual Action Framework for Computational Agents or in short CAFCA [8]. CAFCA provides nine deliberation contexts, structured according to two dimensions, i.e. the reasoning and the sociality dimension. Figure 1 adopted from Jensen [15], uses CAFCA and determines per decision context the type of information that could be relevant.

Figure 1 shows distinct cells of information that could be relevant. Deliberating on this information can be done using many different theories and frameworks. A vast amount of sociological, cognitive and agent literature could be used to inspire deliberation. The survey by Balke [1] distinguishes many frameworks for deliberation in social simulation. Many frameworks exist that deal with deliberation for agents, for example for planning using BDI [19], norms [4], game theory [2] or values [11]. Explicitly incorporating all these techniques in the framework would not only be practically impossible, it will also make the framework very complex. In addition, information from different cells could be used together to make a decision. The framework needs to operate at a more abstract level to deal with all these different types of information and different deliberation methods. Deliberation about deliberation is needed which from now on will be referred to as meta-deliberation or deliberation at the meta-level. The meta-deliberation and its use within the framework will be explained in the following section.

	Individual	Social	Collective
Habitual	Accessible objects, Accessible people, Actions currently performed Accessible means being accessible to the DB in the current context.	Theory of Mind: G, B, I Actions performed by relevant people Accessible objects, Accessible people, Actions currently performed Relevant people are those who have a similar goal to the DB. There is a minimal theory of mind.	Theory of Group: G, B, I Expected action as team member ToM: G, B, I Actions performed by relevant people The group considered is the group that the DB wants to join. The DB need information to perform actions to belong to the group.
Strategic	Useful objects, useful people, Utility Accessible objects, Accessible people, Actions currently performed The set of objects and people is extended to include also not directly accessible objects for plan making.	ToM: Mental attitudes ToM: G, B, I Actions performed by relevant people, Utility Useful objects, useful people Relevant people are those who can aid or hinder the DB. Mental attitudes referes to the information needed to make an estimation of the actions that other agents will perform.	ToG: Mental attitudes, roles Agents in my group ToM: Mental attitudes, Theory of Group: G, B, I Expected action as team member The mental attitudes and roles are information needed for the DB to make decisions in the group. E.g. status, structure of team, mental models, roles
Normative	Related rules, Related laws, Useful objects, Useful people, Utility Rules and laws that are relevant for the current context	Related social norms People's opinion towards those norms Related rules, Related laws, ToM: Mental attitudes Social norms related to the current context. That may hinder or lead behavior of the DB.	(Moral) values of self, Theory of Mind: values, Theory of Group: values ToG: Mental attitudes, roles Agents in my group Related social norms People's opinion towards those norms Consider values of self, others, group.

Fig. 1. Adopted from [15], shows information relevance per CAFCA cell. Where DB = Deliberating Agent, G = Goals, B = Beliefs, and I = Intentions

3 Dynamic Context-Sensitive Deliberation Framework

As discussed in the background section meta-deliberation is needed for Dynamic Context-Sensitive Deliberation (DCSD). To deliberate on a more abstract level, Meta-deliberation should contain the most basic elements necessary for deliberation. As a starting point the output of the framework is considered, which is an action, as the purpose of the framework is to choose one action for the agent. Actions require plans which are sequences of actions. To create a plan a goal is needed. Goals are related to activities, e.g. if the activity is *playing soccer*, goals could be *winning the game*, *playing together with friends* and *physical activity*. The *activity* is what the agent is involved in. Having activities, goals, plans and actions as elements serve as the basis for this meta-deliberation. Meta-deliberation in the framework is thus represented by a tuple containing those four elements. Figure 2 shows the DCSD framework. Meta-criteria and criteria are used to steer context exploration and deliberation and are explained further below.

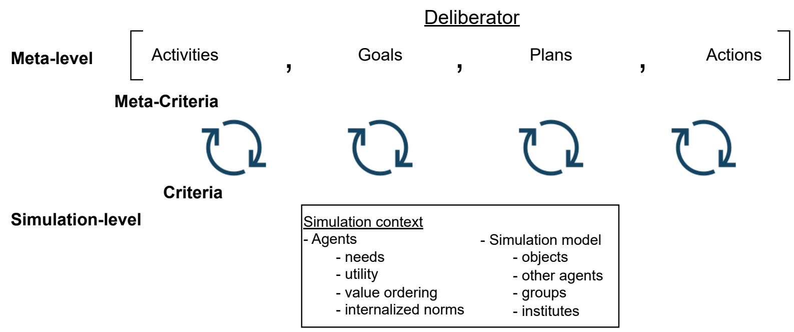

Fig. 2. The Dynamic Context-Sensitive Deliberation cycle.

3.1 Elements of the Tuple

Activities An activity steers what information is relevant for the context. To give an example, when the activity is *working*, a laptop, colleagues, and tasks could be relevant information. While for the activity *grocery shopping* information such as food, stores, and employees could be relevant. Activities serve as starting points. An activity often comes with pre-determined information as is the case in a habit. An activity can be linked with a goal, a plan and an action. For example the habit of grocery shopping can contain the following elements:
$Activity : grocery\_shopping$, $Goal : get\_food$
$Plan : \{drive\_to\_supermarket, buy\_food, drive\_home\}$
$Action : drive\_to\_supermarket$

Goals Goals specify what the agent want to achieve. A goal allows the agent to select or create plans. Selecting a plan can be done with pre-existing plans that fulfill the goal. In some situations the agent has default plans available. Plan creation based on goals can be performed by classic planning described in [3]. A goal makes the context to be consider more specific than an activity.

Plans A plan is a sequence of actions. Plans enable achieving goals that require multiple actions. This enables agents to achieve longer term goals. An example of a $grocery\_shopping$ plan was shown in the activities section. Habits can be represented by plans. A plan does not have to be complete to be useful. There can also be partially filled in plans such as the plan below. These plans can still be useful to the agent as the agent in this case only has to deliberate about which mode of transport to take (e.g. $TRANSPORT = bicycle, car$).
$Plan : \{drive\_to\_supermarket\_by\_TRANSPORT, buy\_food,$
$drive\_home\_by\_TRANSPORT\}$

Actions Agents use actions to manipulate themselves and the simulated world. The goal of the framework is to determine a single action for the agent. The ac-

tion buy_food removes *food* from the store and gives the *food* to the agent. Actions can also have effect on the internal state of the agent. The action eat_food removes *food* from the simulated world and decreases the hunger need in an agent. Some examples of actions are:
$Actions : buy\_food, eat\_food, sleep, work, drive\_car\_home, play\_soccer$

3.2 Meta-Criteria, Criteria and Simulation Context

The framework uses meta-criteria and criteria to manipulate the tuple. Meta-criteria determine which element the deliberator should manipulate. More precisely whether to add (expand) elements or subtract (narrow) elements. The criteria then determines how this manipulation should be performed and which type of information is required from the context.

Meta-Criteria The meta-criteria is defined as being one of the following.
$Meta\_criteria : \{narrow\_activities, narrow\_goals, narrow\_plans, narrow\_actions, expand\_actions, expand\_plans, expand\_goals, expand\_activities\}$

The indicated order should be seen as default, however it can be deviated from in certain contexts. By default, if there are element types that contain multiple elements they will be narrowed first, e.g. if there are multiple activities relevant, the framework will prioritize selecting one activity. If no element type contains more than one element, then by default elements types will be expanded starting with the actions. The mechanism of expanding and narrowing will be explained in the section Manipulating the tuple.

Criteria Selection between activities, goals, plans, actions can be performed by a vast amount of methods. The criteria determines how and based on which type of information the elements in the tuple have to be manipulated. As indicated in the background there is a vast amount of information and deliberation types available. We will not describe all of them but rather give some examples of criteria. Its important to note that criteria generally should be selected based on lowest computational complexity first. In most situations using a default action is preferred over deliberation about norms to choose an action. Also criteria that require only information about the agent's internal state are prioritized over criteria requiring information from the other agents mental state. As a general heuristic one could consider figure 1 by using information from the top-left first, moving gradually to the bottom-right as deliberation continues.

- **Default heuristic:** When there is a default option take it. For example for going to work most people have a default mode of transportation. One could read Gigerenzer [10] for more about heuristics.
- **Typical:** In some contexts some activities, goals, plans or actions are typical in general or for the agent. For example on a Friday evening it could be typical to go to a bar, go to the cinema, watch movies at home.

- **Urgency:** Some activities can be more urgent than others. E.g. sometimes an important meeting at work may make a person skip breakfast. This could be based on which need is more important at that moment.
- **Utility:** Utility can be a criteria for choosing between actions [9]. It can be determined individually but also using game theory [2] or team reasoning [20]. The aspiration and take-the-best heuristics can be used.
- **Preference:** It is possible to make a preference ordering. There can be default preference but it can also stem from for example values [11], but also rules and norms [4]. The aspiration and take-the-best heuristics [10] can be used.

Simulation Context As mentioned in the background section the simulation context is not determined by the framework, but rather by the implemented simulation. Basically any information in the simulation can be considered part of the simulation context. Both physical such as the agents, places, affordances, but also social aspects such as social networks, norms, agent internal state. In principle all of this information is readily available as it is is formalized and implemented. The framework can draw in any of this information to manipulate the tuples.

3.3 Manipulating the Tuple

The goal of the framework is to find a single action for the agent. The framework achieves this by adjusting the elements in the tuples using information from the context. Figure 3 gives an example of both expanding and narrowing.

1) In this specific example the deliberator has the activity: *Leisure* and the goal: *Hang out with friends*. 2) Since there are no plans or actions available the framework will select the meta-criteria: *expand_plans* by default. As criteria: *Typical plans by goal* is selected, which can be a low computational cost method to find plans when a goal is known. In this specific situation the agent has two plans that are typical and can satisfy the goal: *Hang out with friends*. Those are *Go to the pub* and *Go to the cinema*. 3) Since there are now two plans available the deliberator chooses the meta-criteria: *narrow_plan*. The goal is to *hang out with friends*, to succeed the agent needs to be with its friends. To incorporate the preferences of the friends in the decision making *game theory* is selected as criteria. Solving with game theory, there is a preference over *Go to the pub* (two plusses) compared to *Go to the cinema* (one plus). *Go to the pub* is selected as the preferred plan, thus giving the agent one action ending the deliberation cycle.

4 A Proof-of-Concept Implementation

Based on the framework we created an implementation. The implementation[1] is written in Python using Mesa [17]. This implementation contains parts of

[1] https://github.com/maartenjensen/context-sensitive-deliberation, commit number: 03a0191

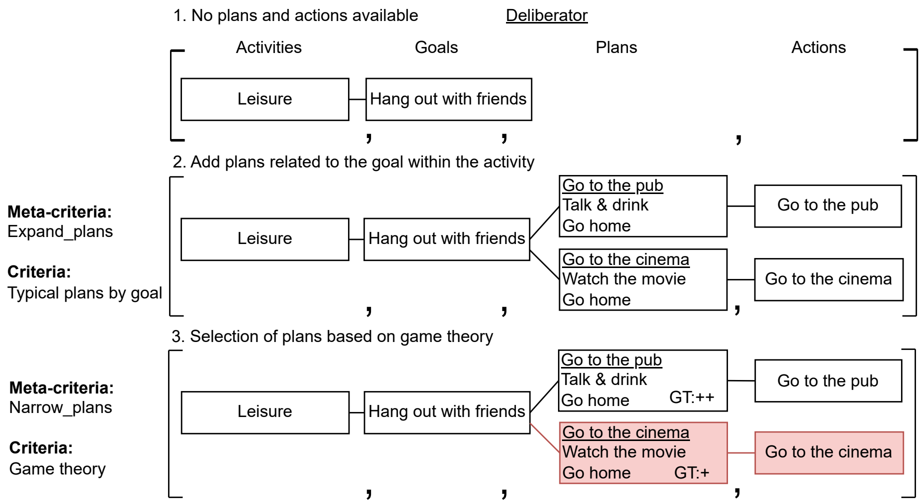

Fig. 3. Example of manipulating the tuple. The plans are expanded based on the available goal and then narrowed based on game theory.

the proposed framework. The goal of this implementation is to show DCSD has potential to decrease computational complexity while retaining realism.

The model simulates daily life in a village containing three agents, houses, a shop and a working place. The time scale of the model is two hours per tick, twelve ticks a day. Both working days and weekends are modeled. All the elements of the framework are shown below. These agents each have slight tweaks to get them to deliberate slightly different. Agent 1: is low on food and has to buy food. Agent 2: has low default food but has other types of food. Agent 3: at the soccer match does not have a habit for determining how to play. We simulated three days, from Thursday (02:00) to Sunday (00:00). The list below indicates all relevant deliberation elements.

Activities	{Eat, Sleep, Work, Buy food, Leisure, Soccer}
Goals	{Eat food, Social activity with friends, Become pro soccer player}
Plans	-
Actions	{Sleep, Eat beef, Eat chicken, Eat tofu, Work, Buy food, Relax, Soccer goalie, Soccer team player, Soccer serious player}
Criteria	{Urgency, Utility, Game theory}

4.1 The Simulation's Deliberation Model

The following steps describe the implemented deliberation model. This is a simplified model where deliberation follows this order.

1. Expand activities and related elements based on time. For example on a working day a time between 7 and 18 makes the working activity relevant.

2. If more than one activities: filter activities based on urgency (need based). Needs are linked to activities. Sleep to sleep need, eat to hunger need, buy food to food safety need, soccer is preset at being the highest urgency since its planned.
3. If not more than one action: Expand actions, actions are expanded based on the goal. Goal eat_food gives actions: $\{eat\_chicken, eat\_tofu, eat\_beef\}$.
4. If more than one action (and not soccer since its social): calculate the utility, in this implementation each agent contains preset utility values for food and soccer actions.
5. Expand goals, goals are expanded since there was no appropriate action. This is only the case with soccer.
6. Narrow down actions, the actions are narrowed down. This is done arbitrarily rather than with game theory. Implementing actual game theory is not relevant for showing the functionality of the framework and could therefore be abstracted.

4.2 Deliberation Cost

Since this is a small scale simulation with many abstractions, the computational time for deliberating does not accurately reflect the time it will take in a large scale simulation. Therefore we implemented an arbitrary deliberation cost (DC) that represents computational complexity. Each of the following steps has a deliberation cost of one. While the last step game theory, has a deliberation cost of three. As game theory could require more deliberation and more information from the context than the other methods.

4.3 The Results

Figure 4 shows deliberation cost over time. The figure shows deliberation cost for the three agents, the average and an intuitive full scale deliberation cost which is added. The deliberation cost is usually one or two. One happens when only one typical activity is available and thus an action can directly be derived. With DC of two usually multiple activities are available so a selection based on urgency is made. The *full scale DC* represents a hypothetical model that deliberates using all the information. The DC for this model is set at five, just below the highest DC of an agent. Since the most expensive computation with the DCSD framework will be slightly more costly than for a full scale model, because the DCSD model has some additional cost due to the meta-deliberation. It is however easily visible that the average DC has a lower DC than the Intuitive full scale DC. Performing a simple integral calculation of the surface below the lines leads to the Average DC using about 30% of the deliberation cost compared to the Intuitive full scale DC. The key take away here is however not that the deliberation cost will always be 30% lower using DCSD. Rather that the more complex the model is, the larger this difference in deliberation cost will be. There are a couple of key properties represented in the framework, which will be described below.

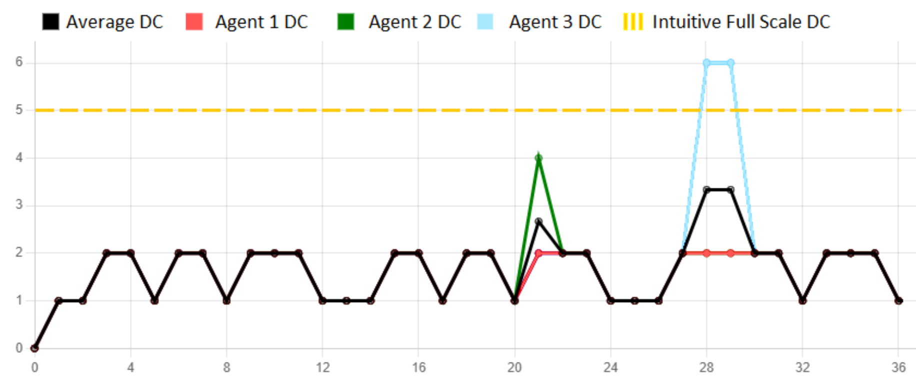

Fig. 4. DC = Deliberation Cost (DC), indicated over time (1 tick = 2 hours)

Following a Habit (Buying Food): At time tick 10, i.e. 20:00, nothing special can be seen. However at this time Agent 1 decided to buy food. This did not increase deliberation cost as more complex deliberation was not necessary. All that had to be done was expanding activities and subsequently filter activities based on urgency. This example shows flexibility of the framework without extra deliberation cost.

Creating a Habit (Selecting Food): Agent 2 has the same deliberation cost until tick 21 (18:00). This is exactly the time the agents have dinner. The agent has a peak in deliberation cost since the default food (Tofu) is not available. It will consider what other food options are available, and subsequently select the food option with the highest utility (Beef). It also changed its default food to beef as it does not have tofu, but does have beef at home. In this example we can see a more complex deliberation can change into a habit. This decreases deliberation cost through the use of learning habits.

Complex Deliberation (Soccer Match): The biggest peak happens for agent 3 at time tick 28 and 29, respectively Saturday 8:00 and 10:00. The agent considers what actions to take during the soccer activity. This involves multiple steps: activity expansion, activity narrowing, goal expansion, and abstracted game theory. Through goal expansion the agent adopts the goal 'social activity with friends'. This leads the agent to a game theoretical approach, it selects becoming a goalie. During this decision, deliberation cost for the agent is the highest. Although it only influences the average deliberation cost slightly. Therefore the simulation would still run efficiently even if some agents use more complex deliberation methods every now and then.

5 Evaluation of the Framework

The results of the implementation show that Dynamic Context-Sensitive Deliberation has potential. Figure 4 shows that in general, average deliberation cost stays relatively low. This is because the agents only need to deliberate on simple decisions in those moments. E.g. sometimes a single action is already given by only considering time and location. While in slightly more advanced cases an action can be selected based on which need is more urgent. These types of deliberation has a low computational complexity, especially compared to a model that would take into account all information at the same time. Having these simple deliberation most of the time decreases the computational complexity over all and makes the model scalable.

The agents in the simulation show complex behavior when necessary. They can draw in all the information needed to make a decision, resulting in the same capabilities as a model that uses all information. At any moment if the situation requires the agent can use all information. This makes the framework potentially as realistic as a framework that takes into account all information all the time. It might even be more realistic as humans also make decisions based on partial information [16], thus making it potentially more like human deliberation.

We expect low computational complexity due to gradual complexity by need. We expect 80% very fast deliberation, e.g. a default action is used. It will not be the case that in the other 20% of cases the most complex deliberation is used. Rather this 20% is split into for example 15% relatively fast deliberation, 4% more complex deliberation and only in 1% of the cases all information is considered. This pattern can continue onward. This principle potentially retains realism while also allowing for scalability. Note that the mentioned numbers are only very rough approximations to illustrate the point.

6 Discussion and Conclusion

The Dynamic Context-Sensitive Deliberation approach is relevant for scalable and realistic models for policy making. However it has to be said that the computational complexity gain will be low for smaller scale models. When creating only small scale models, we do not claim that the DCSD approach will be less computationally complex than other approaches. Still the expressiveness of the framework might be interesting for small scale models. The framework can serve as a coat hanger on which complex deliberation process can easily be attached.

The earlier discussed ASSOCC model did reach its maximum capacity. Switching the need-based deliberation model to a DCSD model does not solve all the problems. A simulation is of course also limited due to visualization and other necessary calculations. E.g. calculations such as need updating after performing an action. However still we expect a significant improvement nonetheless when switching to DCSD, making the model more scalable while retaining realism.

The provided implementation serves as a proof-of-concept. To properly validate the framework it needs to be implemented in a large scale simulation and tested for retaining realism while giving decreased computational complexity. In the future we will re-implement ASSOCC with DCSD. This version will be compared with the original ASSOCC model.

Due to its complexity by necessity principle DCSD can both be scalable as well as realistic. The evaluation shows that the presented framework has potential for scalable and realistic models for policy making. In future work we will implement the framework in the ASSOCC framework. A comparison between the original ASSOCC framework and a Context-Sensitive variant of the ASSOCC framework will validate the Dynamic Context-Sensitive Deliberation framework.

References

1. Balke, T., Gilbert, N.: How do agents make decisions? a survey. Journal of Artificial Societies and Social Simulation (2014)
2. Binmore, K.: Game theory: a very short introduction. OUP Oxford (2007)
3. Bratman, M.: Intention, plans, and practical reason (1987)
4. Castelfranchi, C., Dignum, F., Jonker, C., Treur, J.: Deliberative normative agents: Principles and architecture. vol. 1757, pp. 364–378. Springer (1999)
5. Dey, A.K.: Understanding and using context. Personal and ubiquitous computing **5**, 4–7 (2001)
6. Dignum, F.: Foundations of social simulations for crisis situations (2021)
7. Edmonds, B.: The sociality of context (2014)
8. Elsenbroich, C., Verhagen, H.: The simplicity of complex agents: a contextual action framework for computational agents. Mind and Society **15**, 131–143 (6 2016)
9. Fishburn, P.C.: Utility theory. Management science **14**, 335–378 (1968)
10. Gigerenzer, G., Hertwig, R., Pachur, T.: Heuristics: The Foundations of Adaptive Behavior. Oxford University Press (2011)
11. Heidari, S., Jensen, M., Dignum, F.: Simulations with values (2020)
12. Hinch, R., Probert, W., Nurtay, A., Kendall, M., Wymant, C., Hall, M., Lythgoe, K., Cruz, A.B., Zhao, L., Stewart, A., Others: Effective configurations of a digital contact tracing app: a report to nhsx (2020)
13. Jager, W., Janssen, M.: The need for and development of behaviourally realistic agents. vol. 2581, pp. 36–49. Springer (2002)
14. Jensen, M., Vanhée, L., Kammler, C.: Social simulations for crises: From theories to implementation (2021)
15. Jensen, M., Verhagen, H., Vanhée, L., Dignum, F.: Towards efficient context-sensitive deliberation (2022)
16. Kahneman, D.: Thinking, fast and slow. Macmillan (2011)
17. Kazil, J., Masad, D., Crooks, A.: Utilizing python for agent-based modeling: The mesa framework. pp. 308–317 (2020)
18. Lorig, F., Johansson, E., Davidsson, P.: Agent-based social simulation of the covid-19 pandemic: A systematic review. JASSS: Journal of Artificial Societies and Social Simulation **24** (2021)
19. Rao, A.S., Georgeff, M.P., et al.: Bdi agents: from theory to practice. pp. 312–319 (1995)
20. Sugden, R.: The logic of team reasoning. Philosophical explorations pp. 165–181 (2003)
21. Zimmermann, A., Lorenz, A., Oppermann, R.: An operational definition of context. In: International and Interdisciplinary Conference on Modeling and Using Context. pp. 558–571 (2007)