From Blueprint Personas to Epistemic Agents: A Comparative Study of ASP-Based and L-DINF-Based Approaches to Medical Appointment Scheduling

Valentina Pitoni[1,**], Stefania Costantini[1,3], Andrea Monaldini[1,2] and Alina Vozna[1,2]

[1]*Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila,Italy*
[2]*University of Pisa, Largo B. Pontecorvo, Pisa, 56127, Italy*
[3]*Gruppo Nazionale per il Calcolo Scientifico - INdAM, Roma, Italy*

Abstract
This paper investigates the evolution from a medical appointment scheduling framework based on Answer Set Programming (ASP) integrated with Blueprint Personas to a more cognitively rich, agent-based scheduling system employing the L-DINF epistemic logic framework. We illustrate how agent-oriented models incorporating beliefs, intentions, and dynamic reasoning capabilities can effectively enhance or replace the persona-based constraint optimization traditionally used. Key advantages of the L-DINF model, such as improved adaptability, enhanced explainability, and more human-like decision-making, are emphasized. Furthermore, a structured translation methodology from static personas into dynamic epistemic agents is proposed, accompanied by a modular logical architecture supporting real-time, responsive scheduling.

Keywords
Multi Agent Systems, Modal Logic, Epistemic Logic, Answer Set Programming, Blueprint Personas

1. Introduction

The scheduling of medical appointments remains a significant challenge due to the inherent complexity of balancing limited resources, patient urgency, and individualized preferences. Traditional manual scheduling methods and simple heuristic-based solutions [1, 2] often do not accommodate this complexity, resulting in inefficiencies, prolonged waiting times, and compromised care quality.

In an earlier paper recently accepted, we introduced a scheduling framework based on Answer Set Programming (ASP), a declarative, logic-based paradigm that is well suited for constraint satisfaction and combinatorial optimization [3, 4, 5, 6, 7], enriched with Blueprint Personas [8]. These personas encode structured representations of patients, capturing socio-clinical characteristics, preferences, and accessibility constraints, thereby enabling a form of patient-aware scheduling.

Although ASP with Blueprint Personas proved effective for static scheduling problems, its limitations become evident in dynamic settings. Personas are, by design, static abstractions and do not support real-time reasoning, belief updates, or proactive behavior. As a result, they cannot easily accommodate changing availability, evolving preferences, or unforeseen disruptions in clinical operations.

To address these limitations, we explore the integration of L-DINF, a logic-based framework for modeling intelligent agents with epistemic capabilities such as beliefs, intentions, preferences, and contextual reasoning over actions and environmental changes [9, 10, 11, 12]. Rather than replacing ASP, we propose an integration with the L-DINF framework to enable proactive behavior, intention review, belief-driven planning, and agent coordination, features crucial for modern, responsive healthcare systems.

CILC 2025: 40th Italian Conference on Computational Logic, June 25–27, 2025, Alghero, Italy
*Corresponding author.
valentina.pitoni@univaq.it (V. Pitoni); stefania.costantini@univaq.it (S. Costantini); andrea.monaldini@student.univaq.it (A. Monaldini); alina.vozna@student.univaq.it (A. Vozna)
0000-0002-4245-4073 (V. Pitoni); 0000-0002-5686-6124 (S. Costantini); 0009-0004-2518-2055 (A. Monaldini); 0009-0009-0179-6948 (A. Vozna)

We show how the cognitive properties of L-DINF can be layered on top of the ASP scheduling backbone, supporting dynamic adaptation without the need to recompute entire schedules. The resulting hybrid architecture retains the ASP's optimization strength while enriching it with real-time reasoning and explainability.

This paper presents a structured methodology for translating Blueprint Personas into epistemic agents, articulates the rationale for the integration of L-DINF into ASP-based systems, and demonstrates how such integration addresses the limitations of purely static models. We argue that this hybrid approach is particularly well-suited to the healthcare domain, where scheduling must be responsive to constant change.

The paper is organized as follows: Section 2 reviews the Personas and L-DINF frameworks; Section 3 elaborates on the motivation to integrate epistemic agents into persona-based scheduling; Section 4 analyzes the feasibility of this integration; Section 5 details the translation of personas into L-DINF agents; Section 6 provides a running example; and Section 7 summarizes our findings and outlines future research directions.

2. Background

2.1. Blueprint Personas

Blueprint Personas, originally introduced in digital health transformation projects, act as structured archetypes representing prototypical patients. They combine clinical information (e.g., chronic conditions), social context (e.g., dependency on caregivers), cognitive attributes, and digital literacy levels [8]. When embedded in ASP models, these personas enable individualized constraint modeling while preserving scalability across patient populations.

In our framework, appointment scheduling is modeled as a constraint satisfaction problem (CSP) with embedded optimization goals. Each appointment must satisfy a set of hard constraints (e.g., resource availability, physical accessibility) while optimizing soft constraints such as patient preferences for time, clinic, and physician.

It is important to note that the examples presented in this section are purely illustrative. Personas in our system should be understood as abstract ontological templates, conceptual structures that define variables and relationships related to the patient. These templates are instantiated using real patient data, EHR (Electronic Health Records), and system-level parameters, which populate the factual layer of the ASP model.

Each persona defines a structured combination of clinical status, socio-environmental context, and digital capabilities. This layer of abstraction allows the system to reason over complex, human-centered scheduling needs without hard-coding per-patient logic. Specifically, a patient persona may include attributes such as: personal identity and geographical location, physical or mobility conditions (e.g., disability status), clinic preferences or accessibility needs, sensory sensitivities (e.g., noise or light sensitivity), preferences over physician specialization and experience, preferred time windows for appointments and distance or travel time to clinics. These high-level profiles are encoded as ASP facts, forming the basis for reasoning:

```
patient(p1, "Mario", "Rossi", "L'Aquila").
disabled(p1).
preference(p1, c3).
sensory_preference(p1, "noise").
doctor_preference(p1, "GP", "chronic_diseases", 10).
appointment_preference(p1, c3, 1850, 2000).
distance(p1, c3, 15).
```

Listing 1: Patient Profile with Preferences and Constraints

Clinician personas are modeled in the same way. They include attributes such as medical expertise, experience, and operational limitations, enabling the system to consider staffing constraints and match appropriate providers to patient needs, in fact each visit is identified by: name, cost and classification indicating its chronicity (0 = non-chronic, 1 = chronic):

```
doctor(m1, "Marco", "Bianchi", 52, "L'Aquila", "GP").
doctor_experience(m1, "GP", 25).
doctor_experience(m1, "chronic_diseases", 5).

visit_type(v1, "Cardiology", "Heart Attack", 0, 0, 0).
visit_cost(v1, 1000).
required_sessions(v1, 2).
session_interval(v1, 14, 28).
```

Listing 2: Doctor Profile and Medical Expertise

This ontology-based modeling allows the system to infer constraints and utility values for scheduling in a way that is both medically sound and personalized. It also enables scenario-based validation, where synthetic patients are simulated to test how the system handles edge cases or vulnerable populations.

In the ASP encoding, *inference rules* transform base facts into utility values and feasibility checks, guiding the solver toward optimal, patient-centered outcomes. Preferences are modeled as soft rules, contributing weighted terms to the objective function.

For instance, a patient's preference for a particular doctor type is captured by a scoring rule:

```
appointment_preference_effect(Patient, Time, Clinic, 1) :-
patient(Patient, _, _, _),
clinic(Clinic, _),
availability(Clinic, _, _, Time),
appointment_preference(Patient, _, Start, End),
X = (((Time \ 86400) * 3600) * 100) + (((Time \ 3600) / 60) / 3) * 5,
X <= End, X >= Start.
```

Listing 3: Effect of Patient Preference on Doctor Assignment

Other utility-generating rules model preferences related to clinic selection, time windows, and environmental sensitivity. Collectively, these soft constraints steer the optimization process toward maximizing user satisfaction and care appropriateness.

In parallel, *hard constraints* define the space of valid solutions by enforcing rules grounded in clinical, operational, and ethical requirements. These constraints ensure the feasibility of assignments.

For example, the following constraint ensures that patients are scheduled for the exact number of sessions required for a particular treatment:

```
Sessions { appointment(Patient, Clinic, Doctor, Visit, Time) :
availability(Clinic, Doctor, Visit, Time) } Sessions :-
need(Patient, Visit, _),
required_sessions(Visit, Sessions).
```

Listing 4: Choice Rule for Appointment Allocation Based on Patient Needs

Additional constraints—such as time slot exclusivity, priority handling, clinic capacity, and service delivery modes—are implemented to ensure system realism. For full access to the rules, we provide our codebase online[1].

2.2. Logical Framework: L-DINF

The logical framework *L-DINF*, that we illustrate in this subsection, allows the modeling of group dynamics of cooperative agents. Consequently, one can model agents that can form groups and support each other in performing collective mental actions [13, 14]. Moreover, agents can consider preferences about performing one action instead of another [15]. The logical framework also encompasses the

[1]https://github.com/DawidPado/An-ASP-based-Solution-to-the-Medical-Appointment-Scheduling-Problem/tree/main

possibility for agents to have *roles* within their group of agents. Roles determine which actions each agent is enabled by its group to perform [16]. A mental action is considered executable if at least one agent of the group can perform the action, with the group's approval and on behalf of the group. An agent can join or leave a group whenever it wants (and, consequently, the role of an agent may change as it joins another group).

The agents of a group can share their beliefs, so that any agent can access beliefs of other agents. This ability opens up the possibility of modeling aspects of *"Theory of Mind"* [17]. For instance, an agent can maintain a version (possibly outdated) of the mental state of other agents and perform inferences about such knowledge.

It models agents not just as data structures but as intelligent entities capable of forming beliefs, revising intentions, reasoning over equivalent actions, and adapting to environmental changes. This makes L-DINF particularly well-suited for contexts like healthcare, where schedules must often respond to evolving patient needs and real-world disruptions.

Below we illustrate the syntax and semantics of L-DINF, moreover a formal axiomatic system exists for the logic's core, and it is proven to be strongly complete, but that does not guarantee computational tractability, reasoning in such a rich system is PSPACE-hard; for more detail refer to [9, 10, 11, 12]. *L-DINF* is a logic composed of a static component and a dynamic component. The first, called *L-INF*, is a logic of explicit beliefs and background knowledge. The second component extends the static one with dynamic operators that express the consequences of agents' mental actions.

2.2.1. Syntax

A comprehensive exposition of the logical framework, encompassing its truth conditions, and axiomatic structure, is provided in the referenced publication: https://ceur-ws.org/Vol-3428/paper10.pdf.

Let $Atm = \{p, q, ...\}$ be a countable set of atomic propositions. The set Atm_A is the set of the physical actions that agents can perform, including "active sensing" actions (e.g., *"let's check whether it rains"*, *"let's measure the temperature"*, etc.). Let Agt be a set of agents and Grp the set of groups of agents.

The language of *L-DINF*, denoted by $\mathscr{L}_{L\text{-}DINF}$, is defined by the following grammar:

$$
\begin{array}{lll}
\varphi, \psi & ::= & p \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathbf{B}_i\varphi \mid \mathbf{K}_i\varphi \mid \\
& & do_G^P(\phi_A) \mid do_G(\phi_A) \mid can\_do_G(\phi_A) \mid \\
& & intend_G(\phi_A) \mid exec_G(\alpha) \mid pref\_do_i(\phi_A, d) \mid pref\_do_G(i, \phi_A) \mid \\
& & [G : \alpha]\,\varphi \mid Cl(\phi_A, \phi'_A) \mid fCl_i(\phi_A) \\
\alpha & ::= & +\varphi \mid \vdash(\varphi,\psi) \mid \cap(\varphi,\psi) \mid \downarrow(\varphi,\psi) \mid \dashv(\varphi,\psi)
\end{array}
$$

where p ranges over Atm, $\phi_A, \phi'_A \in Atm_A$, $i \in Agt$, $d \in \mathbb{N}$, and $G \in Grp$. Other Boolean operators are defined from $\neg$ and $\wedge$ in the standard manner.[2] The language of *mental actions* of type α is denoted by $\mathscr{L}_{\mathsf{ACT}}$. The static part *L-INF* of *L-DINF*, includes only those formulas not having sub-formulas of the form $[G : \alpha]\,\varphi$.

Let us briefly describe the intended informal meaning of basic formulas of *L-INF*. As mentioned, we are interested in modelling the reasoning of agents acting cooperatively. We consider the set of agents as partitioned in groups: each agent $i \in Agt$ always belongs to a unique group in Grp. We assume that all agents initially belong to an initial group. Any agent i, at any time, can perform a (physical) action $join_A(i, j)$, for $j \in Agt$, in order to change her group and join j's group. The special case in which $i = j$ denotes the action that allows agent i to leave her current group and form the new singleton group $\{i\}$.

The formula $intend_i(\phi_A)$ indicates the intention of agent i to perform the physical action ϕ_A, in the sense of the BDI agent model [18]. Formulas of this form can be part of agent's knowledge base from the beginning or it can be derived later. In this paper we do not cope with the formalization of BDI, for which the reader may refer, e.g., to [19]. Hence, we will deal with intentions rather informally, also assuming that $intend_G(\phi_A)$ holds whenever all agents of group G intend to perform ϕ_A.

[2] For simplicity, whenever $G = \{i\}$ we will write i as subscript in place of $\{i\}$. So, for instance, we often write $exec_i(\phi_A)$ instead of $exec_{\{i\}}(\phi_A)$ and similarly for other constructs.

The formula $do_i(\phi_A)$ indicates the *actual execution* of action ϕ_A by agent, automatically recorded by the new belief $do_i^P(\phi_A)$ (postfix "*P*" standing for "past" action). Note that, we do not provide an axiomatization for *do* (and similarly for do_G, that indicates the actual execution of ϕ_A by the group of agents *G*). In fact, we assume that in any concrete implementation of the logical framework, do_i and do_G are realized by means of a *semantic attachment* [20], that is, a procedure which connects an agent with its external environment in a way that is unknown at the logical level. The axiomatization only concerns the relationship between doing and being enabled to do.

The expressions $can\_do_i(\phi_A)$ and $pref\_do_i(\phi_A, d)$ are closely related to $do_i(\phi_A)$. In particular, $can\_do_i(\phi_A)$ must be seen as an enabling condition, indicating that the agent *i* is enabled to perform the action ϕ_A, while $pref\_do_i(\phi_A, d)$ indicates the level *d* of preference/willingness of agent *i* to perform ϕ_A.

The formula $pref\_do_G(i, \phi_A)$ indicates that agent *i* exhibits the *maximum level* of preference on performing action ϕ_A within all group members. Notice that, if a group of agents intends to perform an action ϕ_A, this will entail that the entire group intends to do ϕ_A, that will be enabled to be actually executed only if at least one agent $i \in G$ can do it, i.e., it can derive $can\_do_i(\phi_A)$.

The formula $Cl(\phi_A, \phi'_A)$ denoted the equivalence of the two physical actions ϕ_A and ϕ'_A. Intuitively, this means that in the specific practical context at hand, the two actions have "something in common", i.e., for instance, they use similar resources, perform in a similar way, can be used by an agent to obtain equivalent results, etc. Notice that the predicate *Cl* induces a partition of Atm_A in a collection of equivalence classes.

Agents modeled through *L-DINF* deal with two kind of memories, namely, a *working memory* used to represent beliefs, i.e., facts and formulas acquired via perceptions during an agent's operation, and a *long-term memory* used to model agent's background knowledge. Such knowledge is assumed to satisfy *omniscience* principles, such as: closure under conjunction and known implication, closure under logical consequence, and introspection.

Background knowledge of an agent *i* is specified by means of the modal operator $\mathbf{K}_i$, which is actually the usual S5 modal operator often used to model knowledge. The fact that background knowledge is closed under logical consequence is justified because we conceive it as a kind of stable and reliable *knowledge base.* The modal operator $\mathbf{B}_i$, instead, is used to represent the beliefs of agents *i* kept in *i*'s working memory. The contents of the working memory is determined by the mental actions *i* has executed. We assume the background knowledge to include: facts/formulas known by the agent from the beginning, and facts the agent subsequently decided to store in its long-term memory (via a decision-making mechanism not covered here) after processing them in its working memory. We therefore assume that background knowledge is irrevocable, in the sense of being stable over time.

Whenever an agent wants to perform a physical action ϕ'_A, it can exploit the equivalence described by the facts of the form $Cl(\phi_A, \phi'_A)$ to execute a most convenient action ϕ_A (in terms of resources requires, preferences, etc.) drawn from the equivalence class of ϕ'_A. The formula $fCl_i(\phi_A)$ indicates that ϕ_A is the more convenient action among those in the set $\{\phi'_A | Cl(\phi_A, \phi'_A)\}$.

The formulas $exec_G(\alpha)$ express executability of mental actions by a group *G* (which is a consequence of the fact that any member of the group is able to perform the action). They have to be read as: "α is a mental action that an agent in *G* can perform".

A formula of the form $[G : \alpha]\, \varphi$, where α must be a mental action, states that "φ holds after action α has been performed by at least one of the agents in *G*, and all agents in *G* have common knowledge about this fact".

Let us now introduce the dynamic component of the framework. Borrowing from [15, 21], we distinguish five types of mental actions α that capture some of the dynamic properties of explicit beliefs and background knowledge. $+\varphi$, $\downarrow(\varphi, \psi)$, $\cap(\varphi,\psi)$, $\dashv(\varphi, \psi)$, and $\vdash(\varphi,\psi)$. These actions characterize the basic operations of belief formation through inference:

- $+\varphi$: learning perceived belief: the mental operation that serves to form a new belief from a perception φ. A perception may become a belief whenever an agent becomes "aware" of the perception and takes it into explicit consideration.

- $\downarrow(\varphi, \psi)$ is the mental action which consists in inferring ψ from φ, where ψ is an atom: an agent, believing that φ is true and having in its long-term memory that φ implies ψ, starts believing that ψ is true.
- $\cap(\varphi,\psi)$ is the mental action which closes the beliefs φ and belief ψ under conjunction. Namely, $\cap(\varphi,\psi)$ characterizes the mental action of deducing $\varphi \wedge \psi$ from φ and ψ.
- $\dashv(\varphi, \psi)$, where φ and ψ are atoms, is the mental action that performs a simple form of "belief revision", i.e., it removes ψ from the belief set, in case φ is believed and, according to the background knowledge, $\neg\psi$ is logical consequence of φ.
- $\vdash(\varphi,\psi)$, where ψ is an atom; by means of this mental action, an agent believing that φ is true (i.e., it is in the working memory) and that φ implies ψ, starts believing that ψ is true. This last action operates exclusively on the working memory without recovering anything from the background knowledge.

2.2.2. Semantics

Many relevant aspects of an agent's behaviour are specified in the definition of *L-INF model*, including what mental and physical actions an agent can perform, what is the cost of an action and what is the budget that the agent has at its disposal, what is the degree of preference of the agent to perform each action, what is the degree of preference of the agent to use a particular resource. This choice has the advantage of keeping the complexity of the logic under control and making these aspects modular. Definitions 2.1 and 2.2 introduce the notion of *L-INF model*, which is then used to introduce semantics of the static fragment *L-INF*. A model M is composed of two parts. A *core* part $\mathscr{C}_M$ and a collection of *packages* $\mathscr{P}_M$. More specifically:

Definition 2.1. *The* core *part* $\mathscr{C}_M$ *of a model* M *is a tuple* $(W, N, \mathscr{R}, V, S)$, *where*

- W *is a set of worlds (or situations);*
- $\mathscr{R} = \{R_i\}_{i \in Agt}$ *is a collection of equivalence relations on* W: $R_i \subseteq W \times W$;
- $N : Agt \times W \longrightarrow 2^{2^W}$ *is a neighborhood function such that, for each* $i \in Agt$, *each* $w, v \in W$, *and each* $X \subseteq W$ *these conditions hold:*
    - $(\mathbf{C1})$ *if* $X \in N(i, w)$ *then* $X \subseteq \{v \in W \mid wR_iv\}$,
    - $(\mathbf{C2})$ *if* wR_iv *then* $N(i, w) = N(i, v)$;
- $V : W \longrightarrow 2^{Atm}$ *is a valuation function;*
- $S : W \longrightarrow 2^{\{do_G(\phi_A), do_i^P(\phi_A) \mid \phi_A \in Atm_A, i \in Agt, G \in Grp\}}$ *is a valuation function for formulas of the forms* $do_G(\phi_A)$ *and* $do_i^P(\phi_A)$.

To simplify the notation, let $R_i(w)$ denote the set $\{v \in W \mid wR_iv\}$, for $w{\in}W$. The set $R_i(w)$ identifies the situations that agent i considers possible at world w. It is the *epistemic state* of agent i at w. In cognitive terms, $R_i(w)$ can be conceived as the set of all situations that agent i *can retrieve* from its long-term memory and reason about. While $R_i(w)$ concerns background knowledge, $N(i, w)$ is the set of all facts that agent i explicitly believes at world w, a fact being identified with a set of worlds. Hence, if $X \in N(i, w)$ then, the agent i has the fact X under the focus of its attention and believes it. We say that $N(i, w)$ is the explicit *belief set* of agent i at world w. Constraint $(\mathbf{C1})$ imposes that agent i can have explicit in its mind only facts which are compatible with its current epistemic state. Moreover, according to constraint $(\mathbf{C2})$, if a world v is compatible with the epistemic state of agent i at world w, then agent i should have the same explicit beliefs at w and v. In other words, if two situations are equivalent as concerns background knowledge, then they cannot be distinguished through the explicit belief set. This aspect of the semantics can be extended in future work to allow agents make plausible assumptions.

The packages of a model can be thought as modular extensions of the core part. Each package is used to specify a specific feature, such as preferences, costs, executability, etc. Ideally, each package, (may) correspond to some syntactic element of the syntax of *L-INF*. The connection between the syntactic elements and the corresponding package will be established by a suitable component of the semantics (so be seen). The following are some possible packages. Note that we are focusing on those of interests for the purposes of this paper. Plainly, the designer of a particular MAS may decide to include only part of the following packages or even to add/model other features (also providing a suitable adaptation of the notion of truth).

Definition 2.2. *Given a core model $\mathscr{C}_M = (W, N, \mathscr{R}, V, S)$, the packages $\mathscr{P}_M$ are:*

EXECUTABILITY FOR MENTAL ACTIONS

- $E : Agt \times W \longrightarrow 2^{\mathscr{L}_{\mathsf{ACT}}}$ *is an executability function of mental actions such that, for each* $i \in Agt$ *and* $w, v \in W$*, it holds that:*

  (D1) *if* wR_iv *then* $E(i, w) = E(i, v)$;

BUDGET AND COSTS FOR MENTAL ACTIONS

- $B_1 : Agt \times W \longrightarrow \mathbb{N}$ *is a budget function such that, for each* $i \in Agt$ *and* $w, v \in W$*, the following holds*

  (E1) *if* wR_iv *then* $B_1(i, w) = B_1(i, v)$;

- $C_1 : Agt \times \mathscr{L}_{\mathsf{ACT}} \times W \longrightarrow \mathbb{N}$ *is a cost function such that, for each* $i \in Agt$, $\alpha \in \mathscr{L}_{\mathsf{ACT}}$, *and* $w, v \in W$*, it holds that:*

  (F1) *if* wR_iv *then* $C_1(i, \alpha, w) = C_1(i, \alpha, v)$;

EXECUTABILITY FOR PHYSICAL ACTIONS

- $A : Agt \times W \longrightarrow 2^{Atm_A}$ *is an executability function for physical actions such that, for each* $i \in Agt$ *and* $w, v \in W$*, it holds that:*

  (G1) *if* wR_iv *then* $A(i, w) = A(i, v)$;

BUDGET AND COSTS FOR PHYSICAL ACTIONS

- $B_2 : Agt \times W \longrightarrow Amounts$ *is a budget function for physical action, such that, for each* $i \in Agt$*, and* $w, v \in W$*, it holds that:*

  (E2) *if* wR_iv *then* $B_2(i, w) = B_2(i, v)$;

- $C_2 : Agt \times Atm_A \times W \longrightarrow Amounts$ *is a cost function for physical action, such that, for each* $i \in Agt$, $\phi_A \in Atm_A$*, and* $w, v \in W$*, it holds that:*

  (F2) *if* wR_iv *then* $C_2(i, \phi_A, w) = C_2(i, \phi_A, v)$;

AGENTS' ROLES

- $H : Agt \times W \longrightarrow 2^{Atm_A}$ *is an enabling function for physical actions such that, for each* $i \in Agt$ *and* $w, v \in W$*, it holds that:*

  (G2) *if* wR_iv *then* $H(i, w) = H(i, v)$;

PREFERENCES ON PHYSICAL ACTIONS

- $P : Agt \times W \times Atm_A \longrightarrow \mathbb{N}$ *is a preference function for physical actions* ϕ_A *such that, for each* $i \in Agt$ *and* $w, v \in W$*, it holds that:*

  (H1) *if* wR_iv *then* $P(i, w, \phi_A) = P(i, v, \phi_A)$;

  For each i *and* w*, the function* P *induces a preference order* $\preceq_{i,w}$ *on* Atm_A*, such that* $\phi_A \preceq_{i,w} \phi'_A$ *iff* $P(i, w, \phi_A) \leq P(i, w, \phi'_A)$.

EQUIVALENCE OF PHYSICAL ACTIONS

- $Q : Atm_A \times W \longrightarrow 2^{Atm_A}$ *is a function describing a partition of* Atm_A *in equivalence classes (i.e.,* Q *associates each physical action with its equivalence class), such that for each* $i \in Agt$ *and* $w, v \in W$*, it holds that:*

  (I1) *if* wR_iv *then* $Q(\phi_A, w) = Q(\phi_A, v)$;

- $F : Agt \times W \times \mathscr{L}_{\mathrm{ACT}} \longrightarrow \mathscr{L}_{\mathrm{ACT}}$ *is a selector function for physical actions that, given i and w, selects one physical action* $F(i, w, \phi_A)$ *from the equivalence class of* ϕ_A*. Namely, it holds that* $F(i, w, \phi_A) \in Q(\phi_A, w) \wedge \forall \phi'_A \in Q(\phi_A, w)\, \phi'_A \preceq_{i,w} \phi_A$*. For each* $i \in Agt$ *and* $w, v \in W$*, it holds that:*

  (I2) *if* wR_iv *then* $F(i, w, \phi_A) = F(i, v, \phi_A)$*.*

Let us briefly describe the intended features shaped by the packages introduced by Def. 2.2. Notice that the concrete implementation, in a real MAS, the specification of some packages might depend on other packages (for example, in what follows we will describe a possible implementation of *F* that relies on the function *P*).

For an agent *i*, $E(i, w)$ is the set of mental actions that *i* can execute at world *w*. To execute a mental action, *i* has to pay the cost $C_1(i, \alpha, w)$. $B_1(i, w)$ is the budget that *i* has (in *w*) to perform mental actions. As mentioned, concerning physical actions, we are interested in modeling situations where performing an action may require multiple resources. Hence, the cost $C_2(i, \phi_A, w)$ of an action ϕ_A (for agent *i* in world *w*) is a tuple in *Amounts*, while the available budget is described by $B_2(i, w)$. For an agent *i*, the set of physical actions it can execute at *w* is $A(i, w)$. Equivalence between physical actions is determined by function *Q*. That is, $Q(\phi_A, w)$ is the set of physical actions that are equivalent to ϕ_A in *w*. Roles of agents (that, as we will see, affects the capability of agents in a group to execute actions) is described through *H*. Namely, $H(i, w)$ is the set of physical actions that agent *i* is enabled by its group to perform (recall that, at each time instant, an agent belongs to a single group). Agent's preference on execution of physical actions is determined by the function *P*. For an agent *i*, and a physical action ϕ_A, the value of $P(i, w, \phi_A)$ should be intended as a *degree of willingness* of agent *i* to execute ϕ_A at world *w*. Analogously to property **(C2)** imposed in Def. 2.1, the constrain **(D1)** imposes that agent *i* always knows which mental actions it can perform and those it cannot, but if two situations/worlds are equivalent as concerns background knowledge, then they cannot be distinguished through the executability of actions. Similar "indistinguishability' requirements are imposed for each package by conditions **(E1)**, **(F1)**, **(F2)**, **(G1)**, **(H1)**, **(G2)**, **(E2)**, **(I1)**, and **(I2)**.

Let us give some hints on how the functions *P* and *F* might be actually implemented in a concrete MAS. As concerns *P*, we assume defined (e.g., by the MAS designer) a preference relation among (equivalent) actions, for any agent *i*. In practice, this relation might be obtained by exploiting some specific reasoning module. Some possibilities in this sense are described in [22, 23]. Similarly, as for all packages, a specific module in the MAS implementation may be devoted to realize the selector function *F*. Here we outline a simple option in defining *F*, relying on the availability of functions B_2, C_2, and *Q*. Given an agent *i*, a world *w*, and an action ϕ_A, let $\mathscr{A} = \{\phi'_A | \phi'_A \in Q(\phi_A, w) \wedge C_2(i, \phi'_A, w) \le B_2(i, w)\}$ and $\mathscr{A}' \subseteq \mathscr{A}$ such that for each $\phi'_A \in \mathscr{A}'$ the value $sum(C_2(i, \phi'_A, w))$ is minimal among the elements of $\mathscr{A}$. Finally, select the preferred element to be the $\preceq_{i,w}$-maximal element of $\mathscr{A}'$ (i.e., the action ϕ''_A with the larger value of $P(i, w, \phi''_A)$. In case of multiple options, any deterministic criterion can be applied).

3. Motivation for the Integration

The decision to integrate Blueprint Personas with the L-DINF framework stems from both theoretical and practical considerations. While Blueprint Personas provide a powerful abstraction for representing patient profiles within ASP-based scheduling systems, their utility is limited by their inherently static and declarative nature. Once instantiated, these personas remain fixed, unable to evolve or adapt in response to contextual changes. ASP solvers, though highly efficient in producing globally optimized schedules, operate in a centralized and one-shot manner. As a result, they struggle to accommodate the frequent and often unpredictable disruptions that characterize real-world healthcare environments.

The introduction of the L-DINF framework is not intended to replace ASP, but rather to augment it. L-DINF conceptualizes patients (and other agents) as cognitively capable entities endowed with beliefs, intentions, and the ability to engage in ongoing deliberation. These agents are capable of dynamically perceiving environmental changes, such as a clinic becoming unavailable, and updating their internal states accordingly. This agent-based reasoning adds a layer of adaptability and proactivity that complements ASP's static optimization, enabling systems to respond in real time without requiring full recomputation. So, the ASP program may be scheduled for periodic execution to ensure overall optimization, whereas the L-DINF framework can be utilized to manage run-time changes.

More specifically, L-DINF enriches the scheduling process through the following capabilities:

- **Dynamic Adaptation**: Agents revise beliefs and intentions on the fly, enabling local adjustments without re-running the entire ASP program.
- **Enhanced Explainability**: Agent decisions are grounded in explicit reasoning chains involving beliefs, preferences, and inferred intentions.
- **Personalized Scheduling**: Cognitive profiles support individualized, goal-directed planning beyond rule-based parameter matching.
- **Social Coordination**: Through formal group dynamics, agents can coordinate, delegate, and share resources, capabilities that are difficult to express declaratively in ASP alone.

While the integration of L-DINF introduces challenges, including increased logical complexity and computational cost in large-scale deployments, its advantages in flexibility, responsiveness, and transparency make it particularly valuable in dynamic settings. Importantly, in scenarios where scheduling constraints are stable and well-defined, ASP remains the most efficient and suitable choice.

Thus, our proposal is not to abandon ASP, but to extend it. The incorporation of L-DINF offers a synergistic enhancement, leveraging ASP's optimization strengths while overcoming its limitations in adaptivity and reasoning. This hybrid approach enables more resilient, explainable, and patient-centered scheduling solutions that are better aligned with the complex demands of modern healthcare systems.

4. Why the Substitution from Blueprint Personas to L-DINF Agents Is Possible

In this section, we argue that the substitution of Blueprint Personas with L-DINF agents in appointment scheduling systems is not only feasible but also conceptually coherent. This is due to the fact that the fundamental components of a persona, such as preferences, constraints, and goals, can be structurally and semantically mapped into the epistemic constructs provided by the L-DINF framework. However, while some elements map easily, others require non-trivial adaptation due to the shift from static declarative models to autonomous cognitive agents.

4.1. Components That Map Easily

- **Declarative Attributes → Beliefs (B_i)**
  Blueprint Personas include static descriptors such as location, disability status, or sensory preferences. These are analogous to explicit beliefs in L-DINF. Since these attributes are not meant to evolve during execution but form the basis for reasoning, they can be directly encoded as agent beliefs. This mapping is straightforward because the semantics in both models are declarative.
  Example:

```
ASP: disabled(p1).
L-DINF: B_i(disabled(p1)). i is an agent who manages the reservations
```

- **Preferences → Preference Functions ($pref\_do_i$, $P(i, w, \phi_A)$)**
  Preferences in ASP (e.g., for doctors, time slots, or clinics) are modeled as weighted rules or soft constraints. In L-DINF, preferences are formalized using $pref\_do_i$ and scored by a function $P(i, w, \phi_A)$ that quantifies the desirable action of a particular agent i in the world w. This allows agents to rank alternatives in a principled way, much like ASP optimizers—but grounded in agent beliefs. *Example:*

```
ASP: appointment_preference(p1, c3, 1850, 2000).
L-DINF: B_p1(appointment_time_preference(c3, 1850, 2000)).
        pref_do_p1(slot(c3, t1), 8).
```

- **Constraints → Feasibility Rules (*can_do_i*)**
Constraints in ASP, such as distance limits or accessibility conditions, are often specified as hard constraints. In L-DINF, these are interpreted as feasibility conditions that determine whether an agent can perform a given action. This mapping is logical and local: it preserves the original semantics while embedding it in agent-specific reasoning. *Example:*

```
can_do_p1(slot(c3, t1)) <--accessible(c3) and distance(c3)<20.
```

4.2. Components that Require Adaptation

- **Soft Optimization → Local Intentional Reasoning**: In ASP, soft constraints are globally optimized via solvers like Clingo, producing a solution that minimizes a cost function. In L-DINF instead, agents must reason locally over their preferences and constraints to select the most suitable action. This requires restructuring the optimization logic into modular, distributed reasoning procedures.
- **Static Personas → Dynamic Agents**: Blueprint Personas are static input structures. Once declared, they do not change during runtime. In L-DINF, agents can update beliefs based on environmental perception and modify their intentions accordingly. This demands the modeling of inference rules that govern how belief updates propagate through the agent's decision process.
- **One-shot Decision Making → Ongoing Deliberation**: ASP computes a single solution per run. L-DINF, by contrast, supports ongoing deliberation and dynamic replanning. This means that agents can abandon intentions if conditions change or generate new ones in response to updated knowledge. This shift requires modeling the agent's decision lifecycle, including `intend`, `drop`, and `replace` operations.
- **No Social Context → Group Reasoning and Negotiation**: Personas in ASP are individual and isolated. L-DINF allows agents to join groups, share beliefs, and coordinate actions. Implementing this requires defining group membership rules, shared knowledge bases, and collaborative preference negotiation mechanisms. While powerful, this introduces a layer of complexity not present in ASP.

Persona Element	L-DINF Equivalent	Effort to Adapt
Static attributes (e.g., location)	Beliefs (B_i)	Easy
Preferences (doctor, time)	$pref\_do_i$, $P(i, w, \phi_A)$	Easy
Access/distance constraints	Feasibility rules (can_do_i)	Easy
Soft constraints	Local reasoning and selection	Moderate
Behavioral adaptation	Intention updates, inference	High
Multi-agent behavior	Group dynamics ($joinA$, K_G)	High

L-DINF provides a more expressive and adaptive framework that retains the strengths of the original personas while extending their functionality into autonomous, intelligent scheduling agents. Synthesizing we can think of using for translation of all the modules this pseudo code:

```
forall patient P:
    B_P(prefers_clinic = C) <-- preference(P, C)
    pref_do_P(slot(C, T), D) <-- appointment_preference(P, C, S, E) and T in [S, E]
    B_P(sensory_sensitive(S)) <-- sensory_preference(P, S)
    B_P(doctor_preference(T, S, Y)) <-- doctor_preference(P, T, S, Y)
    B_P(distance_to_clinic(C, D)) <-- distance(P, C, D)
```

5. General Schema for Translating Blueprint Personas into L-DINF Agents

The following schema outlines how elements from a Blueprint Persona defined in ASP can be translated into formal components of the L-DINF framework. Each transformation associates declarative or preference-based information from the persona with a corresponding logical construct in the agent's epistemic model.

Blueprint Persona (ASP)	**L-DINF Representation**
`patient(P, …)`	agent identity
`disabled(P)`	$B_i(disabled(P))$ i is an agent who manages the reservations
`preference(P, C)`	$B_i(prefers\_clinic = C)$
`appointment_preference(P, C, S, E)`	$B_i(appointment\_time\_preference(C,S,E))$
`sensory_preference(P, ``noise'')`	$B_i(sensory\_noise\_sensitive)$
`doctor_preference(P, T, S, Y)`	$B_i(doctor\_preference(T,S,Y))$
`distance(P, C, D)`	$B_i(distance\_to\_clinic(C,D))$
`need(P, V, N)`	$intend_i(n\_sessions(V)) \wedge constraint(N)$
`availability(C, T)`	$can\_do_i(slot(C,T))$
`alternative(C1, T1, C2, T2)`	$Cl(slot(C1,T1), slot(C2,T2))$

6. Use Case: Mario, a proactive patient Agent

We consider a patient named **Mario**, who suffers from a chronic condition and is sensitive to noise. Initially, Mario prefers morning appointments at `Clinic C1`, which is located near his home. His medical profile and preferences are encoded as a traditional ASP-style persona in section 2.1. In transitioning to an L-DINF-based representation, Mario is modeled as a cognitive agent endowed with epistemic capabilities, specifically, beliefs, preferences, and intentions,which allow for autonomous and context-aware reasoning. The initial mental state of the agent is represented through the following beliefs:

```
B_mario(prefers_clinic = c1).
B_mario(appointment_time_preference(c1, 0800, 1000)).
B_mario(sensory_noise_sensitive).
B_mario(doctor_preference("GP", "chronic_diseases", 10)).
B_mario(distance_to_clinic(c1, 12)).
```

Preferred time slots are expressed using preference functions:

```
B_mario(pref_do_mario(slot(c1, 0830), 9)).
```

Mario is able to attend this slot if the clinic is accessible and within his mobility budget:

```
B_mario(can_do_mario(slot(c1, 0830)) <-- accessible(c1) and distance(c1)< 20).
```

A key advantage of the L-DINF framework emerges when environmental changes are introduced. Suppose Mario perceives that `Clinic C1` has become inaccessible, expressed as a perceptual update: $+\neg accessible(c1)$. Based on an epistemic inference rule, he is able to derive a revised belief about his action feasibility: $\vdash(\neg accessible(c1), \neg can\_do\_mario(slot(c1,0830)))$. This triggers a goal revision process. Recognizing the infeasibility of his original plan, Mario searches for alternative, equivalent actions: he knows that $slot(c1,0830)$ and $slot(c2,0930)$ are equivalent action and his degree of willingness for the second action is 8 ($P(mario,w,slot(c2,0930)) = 8.$); we also know that he prefers the second action so as to form the fact $fCl\_mario(slot(c2,0930))$. This is expressed as follows:

```
B_mario(Cl(slot(c1, 0830), slot(c2, 0930))).
B_mario(fCl_mario(slot(c2, 0930))).
B_mario(intend_mario(slot(c2, 0930))).
```

Initially, Mario belongs to a singleton group: $G1 = \{mario\}$ and $K_{G1} = \emptyset$. Anna, another patient agent, is part of a group scheduled for Clinic C2: $G2 = \{anna\}$ and $K_{G2} = \{accessible(c2), group\_slot(c2, 0930), doctor\_available(c2)\}$. Mario joins Anna's group using the group action primitive $do\_mario(joinA(mario, anna))$ and after this operation $G2 = \{anna, mario\}$. Mario now has access to the shared group knowledge base, and he can revise his beliefs and confirm his new intention:

```
B_mario(accessible(c2)).
B_mario(can_do_mario(slot(c2, 0930))).
B_mario(do_mario(slot(c2, 0930))).
```

Mario has autonomously revised his beliefs, intentions, and group membership in response to environmental changes. He selects an equivalent action, joins a relevant group, and executes a feasible alternative. This proves the flexibility and expressiveness of the L-DINF agent model.

7. Conclusions

In this work, we have explored the integration of the L-DINF epistemic logic framework into medical appointment scheduling systems originally based on ASP and Blueprint Personas. Rather than replacing ASP, our goal is to enhance its declarative and optimization-oriented strengths with the capacity of L-DINF for cognitive reasoning, agent-level deliberation, and adaptive behavior. This integration addresses the limitations of static persona models by enabling agents to revise beliefs, reformulate intentions, and coordinate with others in response to real-time contextual changes. By embedding cognitive constructs such as beliefs, preferences, and intentions into scheduling agents, the L-DINF framework introduces significant advantages in terms of dynamic adaptability, personalized decision making, and transparent reasoning. Agents modeled in L-DINF are not only responsive to environmental disruptions but also capable of proactively participating in collaborative planning through shared group knowledge and structured delegation mechanisms.

We have proposed a structured translation methodology that transforms Blueprint Personas into epistemic agent models; while this extension introduces additional modeling and computational complexity, particularly in large-scale deployments, the resulting increase in system intelligence, flexibility, and robustness demonstrates its value in realistic and dynamic healthcare environments. We have also shown that the ability to reason dynamically and deliberate continuously allows L-DINF agents to operate effectively within uncertain and evolving clinical scenarios. Future work will involve implementing this hybrid ASP + L-DINF architecture in real-world systems, validating its performance through empirical studies, and demonstrating its practical feasibility across diverse scheduling contexts.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools to generate content. They have used tools to correct minor mistakes.

References

[1] D. Gupta, B. Denton, Appointment scheduling in health care: Challenges and opportunities, IIE transactions 40 (2008) 800–819.

[2] M. H. Alrefaei, A. Diabat, Modelling and optimization of outpatient appointment scheduling, RAIRO-Operations Research-Recherche Opérationnelle 49 (2015) 435–450.

[3] C. Baral, Knowledge representation, reasoning and declarative problem solving, Cambridge university press, 2003.
[4] V. Lifschitz, B. Porter, F. Van Harmelen, Handbook of knowledge representation, Elsevier, 2008.
[5] S. Costantini, A. Formisano, Answer set programming with resources, Journal of Logic and Computation 20 (2010) 533–571.
[6] A. Dyoub, S. Costantini, G. De Gasperis, Answer set programming and agents, The Knowledge Engineering Review 33 (2018) e19.
[7] V. Lifschitz, Answer set programming, volume 3, Springer Cham, 2019.
[8] A. Monaldini, A. Vozna, S. Costantini, Blueprint personas in digital health transformation (2024).
[9] S. Costantini, A. Formisano, V. Pitoni, Timed memory in resource-bounded agents, in: C. Ghidini, B. Magnini, A. Passerini, P. Traverso (Eds.), AI*IA 2018 - Advances in Artificial Intelligence - XVIIth International Conference of the Italian Association for Artificial Intelligence, 2018, Proceedings, volume 11298 of *LNCS*, Springer, 2018, pp. 15–29.
[10] V. Pitoni, Memory management with explicit time in resource-bounded agents, in: S. A. McIlraith, K. Q. Weinberger (Eds.), Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), AAAI Press, 2018, pp. 8133–8134.
[11] S. Costantini, A. Formisano, V. Pitoni, An epistemic logic for multi-agent systems with budget and costs, in: W. Faber, G. Friedrich, M. Gebser, M. Morak (Eds.), Logics in Artificial Intelligence - 17th European Conference, JELIA 2021, Proceedings, volume 12678 of *LNCS*, Springer, 2021, pp. 101–115.
[12] S. Costantini, A. Formisano, V. Pitoni, Preference management in epistemic logic L-DINF, in: A. Dovier, A. Formisano (Eds.), Proceedings of the 38th Italian Conference on Computational Logic, Udine, Italy, June 21-23, 2023, volume 3428 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3428/paper10.pdf.
[13] S. Costantini, V. Pitoni, Towards a logic of "inferable" for self-aware transparent logical agents, in: C. Musto, D. Magazzeni, S. Ruggieri, G. Semeraro (Eds.), Proc. of the Italian Workshop on Explainable AI co-located with 19th International Conference of AI*IA, 2020, volume 2742 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 68–79.
[14] S. Costantini, A. Formisano, V. Pitoni, Cooperation among groups of agents in the epistemic logic L-DINF, in: G. Governatori, A. Turhan (Eds.), Proc. of RuleML+RR'22, volume 13752 of *LNCS*, Springer, 2022, pp. 280–295.
[15] S. Costantini, A. Formisano, V. Pitoni, An epistemic logic for modular development of multi-agent systems, in: N. Alechina, M. Baldoni, B. Logan (Eds.), Proc. of EMAS'21, Revised Selected papers, volume 13190 of *LNCS*, Springer, 2021, pp. 72–91.
[16] S. Costantini, A. Formisano, V. Pitoni, Modelling agents roles in the epistemic logic L-DINF, in: O. Arieli, G. Casini, L. Giordano (Eds.), NMR'22, volume 3197 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 70–79.
[17] A. I. Goldman, Theory of mind, in: E. Margolis, R. Samuels, S. P. Stich (Eds.), The Oxford Handbook of Philosophy of Cognitive Science, volume 1, Oxford University Press, 2012, pp. 402–424.
[18] A. S. Rao, M. Georgeff, Modeling rational agents within a BDI architecture, in: Proc. of the Second Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'91), Morgan Kaufmann, 1991, pp. 473–484.
[19] H. V. Ditmarsch, J. Y. Halpern, W. V. D. Hoek, B. Kooi, Handbook of Epistemic Logic, College Publications, 2015. Editors.
[20] R. W. Weyhrauch, Prolegomena to a theory of mechanized formal reasoning, Artif. Intell. 13 (1980) 133–170.
[21] P. Balbiani, D. F. Duque, E. Lorini, A logical theory of belief dynamics for resource-bounded agents, in: Proc. of AAMAS'16, ACM, 2016, pp. 644–652.
[22] S. Costantini, A. Formisano, Modeling preferences and conditional preferences on resource consumption and production in ASP, J. Algorithms 64 (2009) 3–15.

[23] S. Costantini, A. Formisano, Augmenting weight constraints with complex preferences, in: Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, AAAI Press, USA, 2011.