KONT: Computing Tradeoffs in Normative Multiagent Systems

Özgür Kafalı, Nirav Ajmeri, and Munindar P. Singh
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
Email: {rkafali,najmeri,singh}@ncsu.edu

Abstract

We propose KONT, a formal framework for comparing normative multiagent systems (nMASs) by computing tradeoffs among liveness (something good happens) and safety (nothing bad happens). Safety-focused nMASs restrict agents' actions to avoid undesired enactments. However, such restrictions hinder liveness, particularly in situations such as medical emergencies. We formalize tradeoffs using norms, and develop an approach for understanding to what extent an nMAS promotes liveness or safety. We propose patterns to guide the design of an nMAS with respect to liveness and safety, and prove their correctness. We further quantify liveness and safety using heuristic metrics for an emergency healthcare application. We show that the results of the application corroborate our theoretical development.

1 Introduction

Normative multiagent systems (nMASs) consist of norms that regulate how autonomous agents interact with each other as well as control how agents access nonautonomous components. An important problem in such settings is to characterize how well an nMAS addresses *liveness* and *safety*. Traditionally, liveness means that something good will eventually happen on every enactment and safety means that nothing bad ever happens on any enactment. When the parties are autonomous, we can never ensure traditional liveness or safety. Therefore, we adopt these working definitions: Liveness means that something good will eventually be done on every enactment unless someone is held to account for some infraction. Safety means that nothing bad ever happens on any enactment without someone being held to account for some infraction. An example liveness requirement is to provide guaranteed service for users: physicians are always be able to access patients' electronic health records (EHRs). Otherwise, network administrators are accountable for any downtime. An example safety requirement is preserving privacy: patients' protected health information (PHI) is not disclosed. Otherwise, the hospital administration is accountable for any disclosure.

Tradeoffs often arise between liveness and safety in healthcare. For example, we may choose whether to preserve privacy or act upon private data, and decide that violating privacy to save a patient's life (liveness) is better than preserving the patient's privacy (safety). Moreover, tradeoffs might occur among liveness (or safety) properties when limited resources prevent agents from fulfilling multiple tasks.

Modern information systems can partially implement such tradeoffs via dynamic access control policies between agents and software (Marinovic, Dulay, and Sloman 2014), e.g., relaxing privacy decisions in emergencies by allowing physicians to access patients' EHRs without consent. However, they cannot handle agent autonomy, e.g., how should a physician consult a colleague regarding a patient's PHI to protect the patient's privacy? Therefore, comparing nMASs based on how well they satisfy liveness and safety requirements in various operating modes (e.g., regular practice and medical emergency) is crucial. Norms have been widely studied (Kafalı, Ajmeri, and Singh 2016; Sergot 2013; Singh 2013), in particular, for compliance verification, monitoring, and revision (Alechina, Dastani, and Logan 2013; Aştefănoaei et al. 2009; Chesani et al. 2013; Criado and Such 2016). However, little attention has focused on comparing norms to understand tradeoffs between them.

Accordingly, we propose KONT (stands for computing normative tradeoffs). KONT presents an approach for comparing nMASs. We first develop a strength relation to compare norms using their formally stated properties, and determine which norm would provide greater liveness or safety. For example, a norm that prohibits physicians from disclosing patients' PHI to outsiders or sharing it with colleagues is stronger than a norm that only prohibits physicians from disclosing patients' PHI to outsiders. That is, the latter can be replaced with the former to provide greater safety. We then generalize the strength relation to sets of norms, and compare nMASs with respect to liveness and safety. Following the above example, KONT would determine whether an nMAS with the stronger prohibition is safer than an nMAS with the weaker one if the other norms are unchanged.

Our contributions are as follows: First, we formalize tradeoffs between liveness and safety using nMAS elements (Sections 2 and 3). Second, we develop an approach for comparing nMASs (Section 4) and propose patterns to guide the design of an nMAS (Section 5). Third, we propose heuristic metrics for measuring to what extent an nMAS provides liveness and safety in an emergency healthcare application (Section 6).

Table 1: KONT syntax.

Spec	$\longrightarrow$	Spec $\cup$ Spec | Capability | Norm
Capability	$\longrightarrow$	cap(AG, Expr)
Norm	$\longrightarrow$	n(AG, AG, Expr, Expr)
n	$\longrightarrow$	$a \mid c \mid p$
Expr	$\longrightarrow$	ϕ | $\neg$ Expr | Expr $\wedge$ Expr

2 nMAS Specifications

An nMAS specification is generated by the grammar given in Table 1, where AG represents an agent identifier, Expr is a logical expression, and ϕ is an atomic proposition. We ignore temporal effects by treating each proposition as capturing that a relevant condition has been made true (and remains true subsequently). For example, authenticate means the physician has been authenticated, consent means the patient has given consent. An nMAS specification is a set of capability and norm definitions. Capability cap(AG, Expr) means that AG can bring about Expr.

Definition 1 describes a *norm* following Singh's (2013) model. Here, X and Y are agents, ant and con are logical expressions, and n is a placeholder for the norm type: authorization (a), practical commitment (c), and prohibition (p).

Definition 1. *A norm n(X, Y, ant, con) represents a social relationship between its subject (X) and object (Y) regarding its consequent (con) when its antecedent (ant) holds.*

We model *conditional*, *detached*, *satisfied*, and *violated* norm states. A conditional norm is detached when its antecedent holds. For brevity, we explain the violation conditions for different norm types via examples.

- a(PHY, HOS, consent, EHR $\vee$ operate): a physician PHY is authorized by the hospital HOS to access a patient's EHR as well as to operate upon the patient when the patient's consent is obtained. Here, the object (HOS) is accountable to the subject (PHY). If the physician cannot access the patient's EHR or operate upon the patient when the authorization is detached (i.e., authenticate is true), then the authorization is violated. Authorization as defined is not like a permission in traditional deontic logic: a permission can never be violated but an authorization can be. In effect, an authorization acts as a prohibition on the authorizing party making it accountable to ensure the authorized party is not blocked from the consequent if the antecedent holds (Von Wright 1999).
- c(PHY, HOS, emergency, operate): a physician PHY is practically committed to the hospital HOS to operating upon patients in an emergency. The physician is accountable to the hospital for this commitment. If the physician fails to operate upon patients, the commitment is violated.
- p(PHY, HOS, true, disclose): a physician PHY is prohibited by the hospital HOS from disclosing a patient's PHI to others (disclose). The physician is accountable to the hospital for this prohibition. This prohibition is unconditional because its antecedent is true. If the patient's PHI is disclosed, the prohibition is violated.

3 Understanding Tradeoffs

Liveness and safety often compete with each other. Examples 1 and 2 describe US Health Insurance Portability and Accountability Act (HIPAA) scenarios pertaining to regular and emergency medical practice (HHS 2014).

Example 1. Hospitals are bound by law to keep their patients' PHI secure. Accordingly, Hospital A requires its physicians to authenticate with the hospital's servers (*safety*) before accessing patients' EHRs (*liveness*), and prohibits its physicians from operating upon patients (*liveness*) or disclosing their PHI without consent (*safety*).

Example 1 describes the medical practice adopted by a hospital to comply with the law. Authentication controls access to EHRs. A prohibition provides accountability for physicians regarding disclosure of patients' PHI. Note that these safety regulations do not necessarily affect liveness under regular medical practice, but may do so in medical emergencies due to limited resources.

Example 2. There is a public emergency near Hospital A, and several unconscious patients need to be operated upon immediately. We emphasize two important aspects: Hospital A does not have the required number of physicians on staff to attend to the emergency situation, and there is no opportunity for obtaining consent from the patients to share their PHI with family members.

Example 2 describes a tradeoff between liveness and safety. Following the guidelines of the American College of Emergency Physicians (ACEP) for disasters (ACEP 2013), Hospital A may assign temporary credentials to outside physicians to cope with the load (increasing liveness). Moreover, Hospital A may waive the consent requirement for contacting family members following HIPAA emergency clauses (HHS 2014). However, allowing outside physicians to access patients' EHRs is a safety concern (i.e., not having a control property to restrict access to EHRs).

Figure 1 depicts such tradeoffs between liveness and safety schematically, treating each as a single dimension. Each point represents an nMAS specification. The dashed curve corresponds to specifications that are optimal in that an increase in one dimension would decrease the other. Let us review each case. Note that KONT supports first-order predicates, i.e., agents and other relevant parameters can be specified (see Table 1). For the rest of this paper, we use atomic propositions for brevity to describe norm examples.

Suboptimal describes a nonoptimal specification. Consider authorization a(PHY, HOS, authenticate $\wedge$ expert, operate). Expert physicians who are authenticated can operate upon patients. This specification provides liveness and safety for regular practice. However, liveness is decreased in emergencies due to the restrictions on physicians, e.g., authentication is not possible for outside physicians.

Loss describes a transition to a specification in which one dimension is decreased without affecting the other (e.g., from Suboptimal to Diminished). Consider a(PHY, HOS, true, disclose) in addition to the above authorization. Liveness is not affected, but safety is decreased since physicians are allowed to disclose patients' PHI.

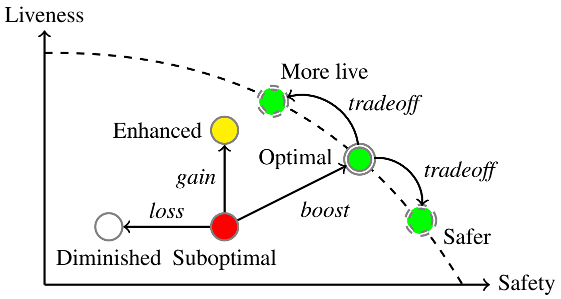

Figure 1: Design space for nMASs, schematically. Regular circles represent nonoptimal specifications. Double circles on the dashed curve represent optimal specifications, for which dashed circles represent distinct tradeoffs between safety and liveness.

Gain describes a transition to a specification in which one dimension is increased without affecting the other (e.g., from Suboptimal to Enhanced). Consider $a(\textsc{phy}, \textsc{hos}, \mathsf{state\_authenticate} \wedge \mathsf{expert}, \mathsf{operate})$ instead of the authorization in Suboptimal. Physicians from other hospitals may perform surgical procedures ($\mathsf{state\_authenticate}$ represents state-wide authentication). Liveness is increased by inviting outside physicians to help. Safety is not reduced since the new authorization too requires physicians to be experts.

Boost describes a transition to a specification in which both dimensions are increased (e.g., from Suboptimal to Optimal). Consider norms $a(\textsc{phy}, \textsc{hos}, \mathsf{credentials}, \mathsf{operate})$ and $p(\textsc{phy}, \textsc{hos}, \neg\mathsf{credentials}, \mathsf{EHR})$. Authentication is replaced by a central registry for physicians. In an emergency, any available physician (not limited to a specific state) can be verified for expertise using credentials, thereby increasing liveness. Safety is increased due to the prohibition for restricting access to EHRs without credentials.

Tradeoff describes a transition between optimal specifications, where only one dimension can be increased. Consider authorization $a(\textsc{phy}, \textsc{hos}, \mathsf{credentials} \vee \mathsf{international}, \mathsf{operate})$. Physicians from international hospitals can operate upon patients, which increases liveness. However, safety is decreased due to difficulty in holding international physicians accountable because their credentials might not be valid in a foreign hospital.

4 Comparing nMAS Specifications

We describe the development of KONT for formalizing the tradeoffs described in Section 3. Φ is the set of domain propositions. $\mathcal{C}$ is the set of agent capabilities. A normative specification, N, is the social component of an nMAS specification. N is the union of the sets of authorizations (A), practical commitments (C), and prohibitions (P). Definition 2 ensures that the normative specification is consistent, i.e., the agent is not both authorized and prohibited or committed and prohibited regarding the same proposition. $\vdash$ represents logical consequence.

Definition 2. *N is consistent iff (i) $p(\textsc{ag}, \textsc{y}, \mathit{ant}_1, \mathit{con}) \notin N$ when $a(\textsc{ag}, \textsc{x}, \mathit{ant}_2, \mathit{con}) \in N$, $\mathit{ant}_2 \vdash \mathit{ant}_1$, and (ii) $p(\textsc{ag}, \textsc{y}, \mathit{ant}_3, \mathit{con}) \notin N$ when $c(\textsc{ag}, \textsc{x}, \mathit{ant}_4, \mathit{con}) \in N$, $\mathit{ant}_4 \vdash \mathit{ant}_3$.*

4.1 Norm Strength

For pairwise comparison of norms and formal understanding of which norms can replace others, we adopt Chopra and Singh's (2009) strength relation for commitments and extend it to all norm types.

Definition 3. *Let n_i and n_j be two norms of the same type and with the same subject and object. Then, n_i is stronger than n_j, denoted $n_i \gg n_j$, iff*

- *n_i is detached whenever n_j is detached; and*
- *n_j is satisfied whenever n_i is satisfied.*

Proposition 1. *$\gg$ is a partial order.*

Proof sketch. $\gg$ is reflexive, antisymmetric, and transitive. Reflexivity and transitivity follow trivially from logical consequence. For antisymmetry, we need to show that $n_i = n_j$ if $n_i \gg n_j$ and $n_j \gg n_i$. When two norms are stronger than each other, they have the same subject and object; their antecedents entail each other; and their consequents entail each other. Therefore, the norms are identical. □

Table 2 shows examples of comparing norms with $\gg$. We omit agents and norm type whenever they are not relevant. That is, we write a norm as $n(\mathsf{ant}, \mathsf{con})$.

Table 2: Examples of norm strength.

$a(\mathsf{authenticate} \vee \mathsf{consent}, \mathsf{EHR})$	$\gg$	$a(\mathsf{consent}, \mathsf{EHR})$
$c(\mathsf{true}, \mathsf{operate} \wedge \mathsf{clinic})$	$\gg$	$c(\mathsf{emergency}, \mathsf{operate})$
$p(\mathsf{true}, \mathsf{consult} \vee \mathsf{disclose})$	$\gg$	$p(\neg\mathsf{emergency}, \mathsf{disclose})$

Next, we present reasoning postulates regarding norms. Postulate 1 covers cases where multiple norms of the same type with the same consequent can be combined into a single norm. Postulates 2–4 cover cases of multiple norms with the same antecedent and different consequents. Note that authorizations and prohibitions are stronger when their consequents are more general but commitments are stronger when their consequents are more specific.

Postulate 1. *$n(r \vee s, u) \in N$ iff $n(r, u) \in N$ and $n(s, u) \in N$*

Postulate 2. *$a(r, u \vee v) \in N$ iff $a(r, u) \in N$ and $a(r, v) \in N$*

Postulate 3. *$c(r, u \wedge v) \in N$ iff $c(r, u) \in N$ and $c(r, v) \in N$*

Postulate 4. *$p(r, u \vee v) \in N$ iff $p(r, u) \in N$ and $p(r, v) \in N$*

Lemma 1. *Normative specifications are closed under norm strength, i.e., if $n_i \in N$ and $n_i \gg n_j$ then $n_j \in N$.*

Example 3. Let $a_1 = a(\textsc{phy}, \textsc{hos}, \mathsf{authenticate} \vee \mathsf{emergency}, \mathsf{EHR}) \in \mathrm{A}_1$. Then, $a_2 = a(\textsc{phy}, \textsc{hos}, \mathsf{authenticate}, \mathsf{EHR}) \in \mathrm{A}_1$ as $a_1 \gg a_2$. Similarly, $a_3 = a(\textsc{phy}, \textsc{hos}, \mathsf{authenticate} \wedge \mathsf{consent}, \mathsf{EHR}) \in \mathrm{A}_1$, $a_4 = a(\textsc{phy}, \textsc{hos}, \mathsf{authenticate}, \mathsf{EHR} \wedge \mathsf{operate}) \in \mathrm{A}_1$ as $a_2 \gg a_3$ and $a_2 \gg a_4$.

Definition 4. *Norm n_i is maximal in N_i, denoted $n_i \in \max(N_i)$, iff $\nexists\, n_j \in N_i$: $n_j \neq n_i$ and $n_j \gg n_i$.*

Definition 4 defines maximal norms in a given normative specification. Consider the authorizations in Example 3 and let A_1 be $\{a_1, a_2, a_3, a_4\}$. Then, $a_1 \in \max(A_1)$.

4.2 nMAS Requirements

We represent requirements R = $\mathrm{R}_c \cup \mathrm{R}_d \cup \mathrm{R}_u$, as the union of three distinct sets of atomic propositions. Here, R_c or *control* properties enforce constraints on agents' actions (liveness or safety); R_d or *desired* properties need to be achieved (liveness); and R_u or *undesired* properties need to be avoided (safety). Control properties describe technical considerations, e.g., a valid authentication is required to access EHR. Desired and undesired properties describe technical (e.g., access to EHR) and social (e.g., interactions among physicians) considerations.

4.3 Comparison of nMASs in Operating Modes

An operating mode M describes agents' capabilities for a specific situation (i.e., what actions can be performed in M). For example, in a medical emergency (emg), patients cannot give consent. Similarly, when there is a server failure (srv), physicians cannot authenticate. However, in regular practice (reg), those actions can be performed. Thus, $\mathcal{C}_{reg} = \mathcal{C}_{emg} \cup \mathcal{C}_{srv} \cup \{$cap(PAT, consent), cap(PHY, authenticate)$\}$.

Agent capabilities are essential for describing operating modes, which specify the technical environment and thus provide a foundation for comparing nMASs. To this end, we consider a possible capability as any pairing of an agent and an expression constructed exclusively via conjunction and disjunction from atomic propositions in R_c. That is, a capability does not involve negation, though the syntax nominally allows negation. Further, an agent's capabilities are grounded on the atomic propositions in R_c. That is, cap(AG, r $\wedge$ s) $\in$ M iff cap(AG, r) $\in$ M and cap(AG, s) $\in$ M. And, cap(AG, r $\vee$ s) $\in$ M iff cap(AG, r) $\in$ M or cap(AG, s) $\in$ M. In addition, a mode is any subset of capabilities provided that if cap(AG, ϕ) $\in$ M and $\phi \vdash \phi'$, then cap(AG, ϕ') $\in$ M. That is, capabilities in modes are closed under strength. In essence, for each agent AG, the set of possible capabilities corresponds to the set of subsets of R_c.

We restrict a normative specification to include only stylized authorizations. In intuitive terms, the subject (authorized party) is an agent whose actions we are concerned with, the object (authorizing party) is an infrastructure provider or facilities operator, and the antecedent is an expression composed entirely of propositions in R_c. In a specific mode, some members of R_c may be directly enabled via agent capabilities; some members via authorizations; others not enabled at all (described next). An authorization may have a non-R_c expression as the consequent. Since specifications are closed under strength, other authorizations may be inferred but the above-mentioned are the only ones directly specified.

We begin describing how to compare nMAS specifications by defining whether a specification enables a proposition. Definition 5 describes enablement, N $\xrightarrow{M} \phi$, which means that N enables proposition ϕ in mode M. There must be an authorization whose consequent is logically entailed by the proposition, and any of the following must hold for the antecedent: the antecedent is true; there is a capability for the antecedent; or the antecedent itself is enabled. Note that there cannot be any prohibitions that interfere with the authorizations due to consistency of normative specifications (Definition 2). Definition 5 does not guarantee satisfaction of the proposition, but verifies whether an agent can achieve it given the agent's capabilities, the controls placed by the technical environment, and the authorizations specified in the social environment.

Definition 5. *$N \xrightarrow{M}$ r iff $\exists$ $a($AG, X, ant, con$) \in N$: r $\vdash$ con and either ant=true or cap(AG, ant) $\in M$ or $N \xrightarrow{M}$ ant.*

Lemma 2. *If $N \xrightarrow{M}$ r and r $\vdash$ s, then $N \xrightarrow{M}$ s.*

Example 4. Let N_1 be $\{a($PHY, HOS, consent, EHR$)$, $a($PAT, HOS, true, consent$)\}$. Then, $\mathrm{N}_1 \xrightarrow{reg}$ EHR, $\mathrm{N}_1 \not\xrightarrow{emg}$ EHR.

Next, we describe three relations for pairwise comparison of normative specifications. We begin with flexibility: AsFlexibleAs(N_i, N_j, M, R) means that N_i is at least as flexible as N_j in mode M with respect to the requirements. We write it as $\mathrm{N}_i \trianglerighteq^M \mathrm{N}_j$ when the requirements are fixed. Definition 6 describes how KONT concludes $\mathrm{N}_i \trianglerighteq^M \mathrm{N}_j$: When N_j enables a proposition r, then N_i must enable r as well. That is, when N_j allows something to happen, then N_i must allow the same thing to happen.

Definition 6. *$N_i \trianglerighteq^M N_j$ iff $\forall$ r: if $N_j \xrightarrow{M}$ r, then $N_i \xrightarrow{M}$ r.*

Proposition 2. *$\trianglerighteq$ is reflexive and transitive.*

We write $\mathrm{N}_i \triangleright^M \mathrm{N}_j$ (N_i is more flexible than N_j in mode M) iff $\mathrm{N}_i \trianglerighteq^M \mathrm{N}_j$, $\mathrm{N}_j \ntrianglerighteq^M \mathrm{N}_i$.

Example 5. Consider two desired properties (R_d=$\{$EHR, operate$\}$), and two specifications: N_2 = $\{a($PHY, HOS, true, EHR$)\}$; N_3 = $\{a($PHY, HOS, authenticate, EHR$)$, $a($PHY, PAT, consent, operate$)\}$.

Let us compare N_2 and N_3 in regular practice mode. N_2 has an unconditional authorization toward accessing patients' EHRs. N_3 has an authorization toward EHR. According to Definition 5, EHR is enabled because there is a capability for the antecedent of the authorization (authenticate). Moreover, N_3 enables operating upon patients since there is an authorization toward operate with a capability for its antecedent (consent). Thus, $\mathrm{N}_3 \trianglerighteq^{reg} \mathrm{N}_2$. However, $\mathrm{N}_2 \ntrianglerighteq^{reg} \mathrm{N}_3$. Therefore, $\mathrm{N}_3 \triangleright^{reg} \mathrm{N}_2$.

We continue with liveness: AsLiveAs(N_i, N_j, M, R) means that N_i is at least as live as N_j in mode M with respect to the requirements. We write it as $\mathrm{N}_i \succcurlyeq^M \mathrm{N}_j$ when the requirements are fixed. Definition 7 describes how KONT concludes $\mathrm{N}_i \succcurlyeq^M \mathrm{N}_j$: When there is a commitment c_j in N_j whose consequent entails a desired property r_d and whose antecedent is enabled, c_j must be in N_i as well. Note that there cannot be any prohibitions that interfere with c_j due to consistency of normative specifications (Definition 2).

Definition 7. *$N_i \succcurlyeq^M N_j$ iff $\forall$ $r_d \in R_d$: $\forall$ c_j = $c($X, Y, ant, con$) \in N_j$: if con $\vdash r_d$ and $N_j \xrightarrow{M}$ ant, then $c_j \in N_i$ and $N_i \xrightarrow{M}$ ant.*

Proposition 3. *$\succcurlyeq$ is reflexive and transitive.*

We write $\mathrm{N}_i \succ^M \mathrm{N}_j$ (N_i is more live than N_j in mode M) iff $\mathrm{N}_i \succcurlyeq^M \mathrm{N}_j$, $\mathrm{N}_j \not\succcurlyeq^M \mathrm{N}_i$.

Example 6. Consider a desired property about prescribing drugs to patients (R_d={prescription}), and two specifications: N_4 = {a(PHY, HOS, true, authenticate), c(PHY, PAT, authenticate, prescription)}; N_5 = {a(PHY, HOS, true, authenticate), c(PHY, PAT, true, prescription $\wedge$ treat)}.

Let us compare N_4 and N_5 in regular practice mode. N_4 has a physician's commitment to prescribe drugs. According to Definition 7, the antecedent of the commitment must be enabled. $N_4 \xrightarrow{reg}$ authenticate since there is an unconditional authorization toward authentication of physicians. Now, N_5 has a stronger commitment than the one in N_4, which means that the same commitment in N_4 is in N_5 as well due to closure under norm strength (Lemma 1). Thus, $N_5 \succcurlyeq^{reg} N_4$. However, $N_4 \not\succcurlyeq^{reg} N_5$. Therefore, $N_5 \succ^{reg} N_4$.

We continue with safety: AsSafeAs(N_i, N_j, M, R) means that N_i is at least as safe as N_j in mode M with respect to the requirements. We write it as $N_i \circ\!\succcurlyeq^{M} N_j$ when the requirements are fixed. Definition 8 describes how KONT concludes $N_i \circ\!\succcurlyeq^{M} N_j$: When there is a prohibition p_j in N_j whose consequent entails an undesired property r_u and whose antecedent is enabled, p_j must be in N_i as well.

Definition 8. *$N_i \circ\!\succcurlyeq^{M} N_j$ iff $\forall\, r_u \in R_u$: $\forall\, p_j = p$(X, Y, ant, con) $\in N_j$: if con $\vdash r_u$ and $N_j \xrightarrow{M}$ ant, then $p_j \in N_i$ and $N_i \xrightarrow{M}$ ant.*

Proposition 4. *$\circ\!\succcurlyeq$ is reflexive and transitive.*

We write $N_i \circ\!\succ^{M} N_j$ (N_i is safer than N_j in mode M) iff $N_i \circ\!\succcurlyeq^{M} N_j$, $N_j \circ\!\not\succcurlyeq^{M} N_i$.

Example 7. Consider an undesired property about disclosing patients' PHI (R_u={disclose}), and two specifications: N_6 = {p(PHY, HOS, true, disclose)}; N_7 = {p(PHY, HOS, true, consult $\vee$ disclose)}.

Let us compare N_6 and N_7. The consequent of the prohibition in N_6 entails disclosure of PHI, for which N_7 has a stronger prohibition. Thus, $N_7 \circ\!\succcurlyeq^{M} N_6$ according to Definition 8. However, $N_6 \circ\!\not\succcurlyeq^{M} N_7$. Therefore, $N_7 \circ\!\succ^{M} N_6$.

Notice that, as defined above, a normative specification may contain authorizations that are irrelevant with respect to a specific mode, in that the normative specification may authorize outcomes based on antecedents that are not within the capabilities of any agent in that mode. For example, in a mode where the patient lacks the capability to give consent, the authorization a(PHY, HOS, consent, EHR) is superfluous. Specifications that contain such superfluous authorizations can be different from each other even if they enable the same propositions in some mode. Therefore, we factor out modes by comparing specifications over all possible modes.

Accordingly, we define $N_i \trianglerighteq N_j$ to mean for all M, $N_i \trianglerighteq^{M} N_j$; $N_i \succcurlyeq N_j$ to mean for all M, $N_i \succcurlyeq^{M} N_j$; and $N_i \circ\!\succcurlyeq N_j$ to mean for all M, $N_i \circ\!\succcurlyeq^{M} N_j$.

Let us revisit Example 7. $N_7 \circ\!\succcurlyeq N_6$. That is, N_7 is as safe as N_6 in all modes because both prohibitions are unconditional (their antecedents are enabled in all modes).

5 nMAS Design via Patterns

We propose normative patterns (Kafalı, Ajmeri, and Singh 2016; Singh, Chopra, and Desai 2009) to guide nMAS design. The patterns transform a given normative specification into one that satisfies a liveness or safety requirement.

5.1 Construction Patterns

The following design patterns create norms based on the liveness and safety requirements.

Gateway creates authorization a(X, Y, r_c, r_d) with respect to control property r_c and desired property r_d.

Achievement creates commitment c(X, Y, r_c, r_d) with respect to control property r_c and desired property r_d.

Guard creates prohibition p(X, Y, r_c, r_u) with respect to control property r_c and undesired property r_u.

5.2 Tradeoff Patterns

The following refinement patterns implement tradeoffs between specifications. That is, applying a refinement pattern increases one dimension (liveness or safety), but might decrease the other.

Progress transforms N_i into N_j with respect to a desired property r_d. N_j is constructed from N_i as follows:

$$\frac{c_i = c(\text{X}, \text{Y}, \mathsf{ant}_i, \mathsf{con}_i) \in \max(\mathrm{N}_i), \mathsf{con}_i \vdash r_d, \mathrm{N}_i \xrightarrow{M} \mathsf{ant}_i}{\begin{array}{c}\mathrm{N}_j = \mathrm{N}_i \cup \{c_j = c(\text{X}, \text{Y}, \mathsf{ant}_j, \mathsf{con}_j)\}, \\ c_j \gg c_i, \text{where } c_i \not\gg c_j, \mathrm{N}_j \xrightarrow{M} \mathsf{ant}_j\end{array}}$$

Let us revisit Example 6. Applying the *Progress* pattern adds c(PHY, PAT, true, prescription $\wedge$ treat) in N_5, which is a strictly stronger commitment than c(PHY, PAT, authenticate, prescription) in N_4. Therefore, $N_5 \succ^{M} N_4$.

Fortify transforms N_i into N_j with respect to an undesired property r_u. N_j is constructed from N_i as follows:

$$\frac{p_i = p(\text{X}, \text{Y}, \mathsf{ant}_i, \mathsf{con}_i) \in \max(\mathrm{N}_i), \mathsf{con}_i \vdash r_u, \mathrm{N}_i \xrightarrow{M} \mathsf{ant}_i}{\begin{array}{c}\mathrm{N}_j = \mathrm{N}_i \cup \{p_j = p(\text{X}, \text{Y}, \mathsf{ant}_j, \mathsf{con}_j)\}, \\ p_j \gg p_i, \text{where } p_i \not\gg p_j, \mathrm{N}_j \xrightarrow{M} \mathsf{ant}_j\end{array}}$$

Let us revisit Example 7. Applying the *Fortify* pattern adds p(PHY, HOS, true, consult $\vee$ disclose) in N_7, which is a strictly stronger prohibition than p(PHY, HOS, true, disclose) in N_6. Therefore, $N_7 \circ\!\succ^{M} N_6$.

5.3 Combining the Patterns

KONT uses the above patterns in sequence to guide the creation of a specification of an nMAS with respect to liveness and safety requirements. For example, assume we have two desired properties R_d={treat, operate} and two control properties R_c={emergency, consent}. The *Achievement* pattern creates a commitment c(PHY, HOS, emergency, operate), and the *Progress* pattern adds a stronger commitment c(PHY, HOS, emergency $\vee$ consent, treat $\wedge$ operate) to the specification to make the specification more live.

6 Emergency Healthcare Application

We adopt ECLiPSe, a constraint logic programming framework (Apt and Wallace 2007), to perform experiments on an emergency healthcare setting. ECLiPSe offers a conceptual modeling language that extends Prolog and provides constraint solver libraries for solving integer constraints. We provide a prototype implementation, KONTES[1] (stands for KONT experimental setting), for evaluating to what extent nMAS specifications provide liveness and safety.

We generate integer costs for surgical procedures, and propose heuristic metrics to measure liveness and safety for two nMASs: N_S corresponds to a Suboptimal specification and N_E corresponds to an Enhanced specification as shown in Figure 1 (see prototype implementation for more details on these two specifications).

Liveness score measures the distance of a given nMAS specification from a perfectly live (i.e., ideal) specification, which represents a (possibly fictitious) specification with no constraints on any surgical procedure. We compute liveness as follows:

$$\text{Liveness score} = \frac{\text{supported procedures}}{\text{all procedures}}$$

Safety score measures the distance of a given nMAS specification from a perfectly safe (possibly fictitious) specification. Here, we choose the perfectly safe specification to be one where outside physicians are not allowed to perform any surgical procedure. Then, the safety score measures how many bad alternatives are avoided. We compute safety as follows:

$$\text{Safety score} = 1 - \frac{\text{procedures by outside physicians}}{\text{supported procedures}}$$

Note that the liveness and safety scores are not necessarily complements of each other, i.e., they do not add up to one.

Table 3 shows the results for two specifications, N_S and N_E, in three modes. We use the heuristic scores for empirical evaluation of our theoretical development, e.g., if $\mathrm{N}_i \succcurlyeq^M \mathrm{N}_j$, then N_i's liveness score would be higher than N_j's score in mode M. Whereas a perfect score (1.00) indicates an ideal specification for a given metric, the same specification may have a lower score for a different metric. We present one such metric for liveness and safety. $\mathrm{N}_S \circ\!\succcurlyeq \mathrm{N}_E$ in all modes since N_S puts more security restrictions on physicians. More specifically, it does not allow outside physicians to perform any surgical procedures, thus leading to a perfect safety score. In regular practice, $\mathrm{N}_E \circ\!\succcurlyeq^{reg} \mathrm{N}_S$ because the relaxation conditions of N_E only work in nonregular practice. In regular practice, both specifications support nine out of the 48 procedures supported by a perfectly live specification, leading to a liveness score of 0.19. Thus, $\mathrm{N}_S \succcurlyeq^{reg} \mathrm{N}_E$.

In a medical emergency, the demand for surgical procedures increases. N_S fails to meet this demand, leading to a liveness score of 0.10 with only five alternatives supported. N_E supports 35 alternatives, leading to a liveness score of 0.73. This means that it covers more desired properties (R_d) described in terms of surgical procedures. Therefore, $\mathrm{N}_E \succ^{emg} \mathrm{N}_S$. However, the safety score of N_E for the medical emergency case is low (0.14), since most of the procedures supported are performed by outside physicians who are not authenticated by the hospital. Therefore, $\mathrm{N}_S \circ\!\succ^{emg} \mathrm{N}_E$.

Table 3: Liveness and safety scores for comparing nMASs.

Mode of	**Liveness**		**Safety**	
operation	$\mathbf{N}_S$	$\mathbf{N}_E$	$\mathbf{N}_S$	$\mathbf{N}_E$
Regular practice	0.19	0.19	1.00	1.00
Medical emergency	0.10	0.73	1.00	0.14
Server failure	0.00	0.21	1.00	0.00

When there is a server failure and authentication is not possible, no procedures are supported by N_S, whereas N_E supports ten alternatives, leading to a liveness score of 0.21. Thus, $\mathrm{N}_E \succ^{srv} \mathrm{N}_S$. However, N_E is completely unsafe for the server failure case, since all the procedures supported are performed by outside physicians. Therefore, $\mathrm{N}_S \circ\!\succ^{srv} \mathrm{N}_E$.

7 Discussion

We developed KONT, an approach for comparing nMASs. Our approach adopts a sociotechnical perspective: Regulation via norms preserves liveness by limiting the need for regimentation to curtail autonomy, and safety is achieved through accountability. We proposed normative patterns to design an nMAS with respect to liveness and safety, and provided experiments on an emergency healthcare application using constraint logic programming. We showed that the results of the application corroborate our conceptual development.

Sergot (2013) discusses the correspondence of normative relations among agents with Hohfeldian legal concepts such as duties and rights, and presents semantics using deontic logics. Alechina et al. (2013) focus on conditional norms with deadlines, and extend CTL and ATL with sanctions to reason about the effects of normative update. They measure norm compliance by verifying if specific states are reached before the deadline, and enforce norms via sanctions. We go beyond verification of compliance, and provide pairwise comparison of normative specifications.

Kafalı and Yolum (2016) propose an approach for monitoring an agent's interactions to determine whether the agent is progressing as expected. In particular, they verify whether the agent's expectations (represented by a set of propositions and commitments) are satisfiable by its current state. Governatori (2013) proposes a conceptual abstract framework to model normative requirements, formalizes different types of obligations, and verifies whether a business process is compliant with requirements (set of obligations). The above works are limited to the representation and verification of commitments and obligations. The main shortcoming with verification based approaches is the inability to handle autonomy despite norms. That is, they consider regimented

[1] See complete implementation at https://research.csc.ncsu.edu/mas/code/security/kontes-clp/.

systems in which norm-violating paths are eliminated from execution. We expand on verification, and provide metrics for measuring the liveness and safety of specifications. It would be valuable to explore probabilistic model checking tools besides CLP.

Vasconcelos et al. (2009) propose methods for resolving conflicts among norms. Their resolution method, norm *curtailment*, manipulates the constraints associated with norms, e.g., reduce the scope of a prohibition to avoid conflict with an obligation. We assume no such conflicts occur among authorizations, commitments, and prohibition in a normative specification. Zhang et al. (2016) discuss probabilistic commitments in open environments with uncertainty. Extending KONT with probabilistic commitments is an interesting direction, especially in the case of dialectical commitments. For example, a physician commits to a patient having cancer with 80% certainty, or an outside physician in an emergency situation commits to being an expert on a specific surgical operation without providing any credentials.

Artikis (2009) proposes an infrastructure to specify dynamic protocol specifications for open multiagent systems. Specifications are modeled as metric spaces, and the infrastructure enables agents to specify protocols at design-time and modify protocols at run-time. According to Artikis' metric, protocols are evaluated based on the distance between specifications at two distinct time points. In a sense, our heuristic metrics are similar since we measure the distance from a perfectly live or safe specification. However, Artikis' metric space does not cover liveness or safety requirements.

Sanctions (Nardin et al. 2016) would add another dimension to KONT's normative comparison. Sanctions provide compensation for norm violations (liveness) as well as deterrence against violating norms (safety). Moreover, our heuristic metrics for the healthcare application can be adapted to other domains, e.g., in consumer banking, liveness corresponds to how easily customers can make payments, whereas safety corresponds to absence of false payments. Additional tradeoff dimensions beyond liveness and safety would be useful to investigate.

Acknowledgments

This research is supported by the US Department of Defense under the Science of Security Lablet grant. We thank our lablet colleagues and the anonymous referees for their helpful comments.

References

ACEP. 2013. Guidelines for crisis standards of care during disasters. American College of Emergency Physicians (ACEP). http://goo.gl/HXWRnH.

Alechina, N.; Dastani, M.; and Logan, B. 2013. Reasoning about normative update. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 20–26.

Apt, K. R., and Wallace, M. 2007. *Constraint Logic Programming Using Eclipse*. Cambridge University Press.

Artikis, A. 2009. Dynamic protocols for open agent systems. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, 97–104.

Aştefănoaei, L.; Dastani, M.; Meyer, J. C.; and de Boer, F. S. 2009. On the semantics and verification of normative multiagent systems. *J. UCS* 15(13):2629–2652.

Chesani, F.; Mello, P.; Montali, M.; and Torroni, P. 2013. Representing and monitoring social commitments using the event calculus. *Autonomous Agents and Multi-Agent Systems* 27(1):85–130.

Chopra, A. K., and Singh, M. P. 2009. Multiagent commitment alignment. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, 937–944.

Criado, N., and Such, J. M. 2016. Selective norm monitoring. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, 208–214.

Governatori, G. 2013. Business process compliance: An abstract normative framework. *Information Technology* 55(6):231–238.

HHS. 2014. Bulletin: HIPAA privacy in emergency situations. United States Department of Health & Human Services (HHS). http://www.hhs.gov/ocr/privacy/hipaa/understanding/special/emergency/.

Kafalı, Ö.; Ajmeri, N.; and Singh, M. P. 2016. Revani: Revising and verifying normative specifications for privacy. *IEEE Intelligent Systems* 31(5):8–15.

Kafalı, Ö., and Yolum, P. 2016. PISAGOR: A proactive software agent for monitoring interactions. *Knowledge and Information Systems* 47(1):215–239.

Marinovic, S.; Dulay, N.; and Sloman, M. 2014. Rumpole: An introspective break-glass access control language. *ACM Transactions on Information and System Security* 17(1):2:1–2:31.

Nardin, L. G.; Balke-Visser, T.; Ajmeri, N.; Kalia, A. K.; Sichman, J. S.; and Singh, M. P. 2016. Classifying sanctions and designing a conceptual sanctioning process model for socio-technical systems. *The Knowledge Engineering Review* 31:142–166.

Sergot, M. 2013. Normative positions. In *Handbook of Deontic Logic and Normative Systems*. College Publications. chapter 5, 353–406.

Singh, M. P.; Chopra, A. K.; and Desai, N. 2009. Commitment-based service-oriented architecture. *IEEE Computer* 42(11):72–79.

Singh, M. P. 2013. Norms as a basis for governing sociotechnical systems. *ACM Transactions on Intelligent Systems and Technology* 5(1):21:1–21:23.

Vasconcelos, W. W.; Kollingbaum, M. J.; and Norman, T. J. 2009. Normative conflict resolution in multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 19(2):124–152.

Von Wright, G. H. 1999. Deontic logic: A personal view. *Ratio Juris* 12(1):26–38.

Zhang, Q.; Durfee, E. H.; Singh, S. P.; Chen, A.; and Witwicki, S. J. 2016. Commitment semantics for sequential decision making under reward uncertainty. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 3315–3323.