A Modeling Approach for Getting to Win-Win in Industrial Collaboration under Strategic Coopetition

Vik Pant¹* and Eric Yu¹,²

¹Faculty of Information, University of Toronto, 140 St George St, Toronto, Canada
²Department of Computer Science, University of Toronto, 214 College St, Toronto, Canada

vik.pant@mail.utoronto.ca, eric.yu@utoronto.ca

Abstract. Interorganizational coopetition describes a relationship in which two or more organizations cooperate and compete simultaneously. Actors under coopetition cooperate to achieve collective objectives and compete to maximize their individual benefits. Such relationships are based on the logic of win-win strategies that necessitate decision-makers in coopeting organizations to develop relationships that yield favorable outcomes for each actor. We follow a strategic modeling approach that combines i* goal-modeling to explore strategic alternatives of actors with Game Tree decision-modeling to evaluate the actions and payoffs of those players. In this article, we elaborate on the method, illustrating one particular pathway towards a positive-sum outcome – through the introduction of an intermediary actor. This article demonstrates the activation of one component in this guided approach of systematically searching for alternatives to generate a new win-win strategy. We also present a meta-model for relating i* models and Game Trees. A hypothetical industrial scenario focusing on the Industrial Data Space, which is a platform that can help organizations to overcome obstacles to data sharing in a coopetitive ecosystem, is used to explain this approach.

Keywords: Coopetition, Win-Win, Design, Modeling.

1 Introduction

Coopetition refers to concomitant cooperation and competition among actors wherein actors “cooperate to grow the pie and compete to split it up” [1]. Actors under coopetition simultaneously manage interest structures that are partially congruent and partially divergent [2]. Partial congruence emerges from actors sharing in certain common objectives while partial divergence emanates from each actor’s pursuit of self-interest. Coopetition has become

* Corresponding author

© 2019 Vik Pant et al. This is an open access article licensed under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0).


Additional information. PII S225599221900110X. Received: 2 April 2019. Accepted: 27 May 2019. Available online: 19 July 2019.
“increasingly popular in recent years” [3] and is widely observed in various domains including business, politics, and diplomacy [4].

Coopetition is predicated on the rationale of positive-sum outcomes through which all actors are better off by coopeting rather than by purely competing or solely cooperating. This aspect of coopetition requires decision-makers in coopetition organizations to define, develop, and analyze win-win strategies. We apply a synergistic approach that combines i* goal-modeling with Game Tree decision-modeling to generate and discriminate win-win strategies in a structured and systematic manner. In [5], we illustrated a win-win scenario arrived at by generating a new alternative for achieving a goal, using the means-ends reasoning supported by i* goal-modeling. In [6], we illustrated a different pathway to get to win-win by introducing a new actor within an existing relationship between two actors. This article extends and elaborates our previous work [5], [6].

In this article, we formalize the semantic relationship between the modeling languages utilized in [5], [6] by proposing a metamodel to link them. We use a hypothetical industrial scenario adapted from practitioner and scholarly literatures to explain this approach.

Coopetition research originated in the field of economics where researchers applied concepts from game theory to explain the motivations of coopetitive actors [7]. According to game theory, three types of results are possible in strategic relationships between actors: positive-sum, zero-sum, and negative-sum [8]. In positive-sum outcomes, all players are better off and in negative-sum outcomes all players are worse off [8]. In zero-sum outcomes, the amount of gain by some players equals the amount of loss by other players.

These outcomes are correlated to distinct types of strategies that are adopted by players in coopetitive relationships: win-win, lose-lose, and lose-lose. Win-win strategies are the only durable options for sustaining coopetitive relationships. Win-lose strategies are unsustainable in coopetitive relationships because some actors (i.e., those that are disadvantaged) will be worse off as a result and these actors are likely to withdraw from or abandon such relationships.

2 Motivating Example: Interorganizational Knowledge-sharing in Pharmaceutical Industry

Drug discovery and biopharmaceutical development is characterized by long innovation cycles and high capital requirements. Pharmaceutical companies share knowledge with each other to accelerate “product development processes”, “reduce costs”, and increase “development productivity” [9]. Coopetitive relationships within research and development (R&D) alliances in the pharmaceutical industry are described in [9]. The complexity of interorganizational knowledge-sharing in the pharmaceutical industry is discussed in [10], [11].

Knowledge-sharing can expose members of R&D alliances to the risk of knowledge expropriation through knowledge leakage [10], [11]. This is because R&D alliances can be among firms that are competitors in the marketplace. Such firms are coopetitors because they cooperate in the R&D domain but compete for customers in the marketplace. Knowledge leakage occurs when a “focal firm's private knowledge is intentionally appropriated by or unintentionally transferred to partners beyond the scope of the alliance agreement” [12]. Knowledge expropriation is an opportunistic behavior [13], [14] that is motivated by the desire of firms to engage in ‘learning races’ [15], [16] to ‘learn faster’ [17], [18] than each other in the pursuit of ‘competitive advantage’ [19], [20]. Knowledge management researchers refer to this phenomenon as ‘boundary paradox’ and ‘learning paradox’ [21].

The potential for knowledge expropriation through knowledge leakage implies that simple knowledge-sharing under cooperation can lead to win-lose or lose-lose outcomes. In such a scenario, no immediate solutions might exist for the firms under coopetition to get to positive-sum outcomes. Subject matter experts (SMEs) and domain specialists in such firms might contemplate different pathways for generating win-win strategies. For instance, one option might be for coopeting firms to engage other actors, illustrated in Section 5 in this article, into their relationship to help reduce opportunities for exploitation. Another option might be for coopeting
firms to jointly develop and operate knowledge-sharing systems in-house that mitigate the risks of knowledge misappropriation. Yet another option might be for the actors to change their motivations to disincentivize opportunistnic behavior through rewards and penalties.

The pathway selected by SMEs in cooeping firms will depend on the specifics of their firms as well as their relationships. In the real-world, the process of generating and discriminating among such options is complex and nontrivial due to two main reasons [22]. First, the decision space of each actor is constrained or enlarged by interdependencies with potential actions of other actors. Second, trade-offs between multiple competing objectives lead to different prioritization of alternatives by each actor due to the unique preference structure of that actor.

In the next section we detail an ontology and a methodology for generating new win-win strategies in inter-organizational relationships. In Section 4 and Section 5, we offer a catalog of objectives and options pertaining to inter-organizational knowledge sharing. These artefacts are complementary to the creativity and imagination of SMEs and domain specialists for identifying and generating win-win strategies. Therefore, SMEs and domain specialists can use these artefacts to search the solution space as well as synthesize new alternatives and options in it.

3 A Framework with i* and Game Trees for Modeling Win-Win Strategies

In this article, we illustrate the use of a mediating actor to get to win-win by applying the modeling approach that is depicted in Figure 1. This process interleaves steps from i* and Game Tree modeling in an incremental and iterative manner. It is useful for co-developing complementary models that jointly offer greater explainability, interpretability, and transparency than either can individually. Figure 2 presents a UML class diagram that depicts relevant modeling concepts from i* and Game Trees in terms of their semantics and relationships. The methodology, in Figure 1, and metamodel, in Figure 2, are needed jointly for discriminating and generating win-win strategies in a systematic and structured manner.

i* (denoting distributed intentionality) is a goal- and actor-oriented modeling language that supports strategic reasoning. The semantics and notation of i* are explained in [8]. Game Trees are decision trees that support representation of decisions and payoffs associated with actors in a game. They are typically visualized as directed acyclic labeled tree graphs. In Game Theory, a game refers to any social situation in which two or more players are involved. A player is an active participant in a strategic relationship with one or more players. A payoff is the reward (positive) or penalty (negative) associated with a specific course of action. A course of action is a sequence of decisions and actions undertaken by the players in a game. Solving a game refers to selecting a reward maximizing or penalty minimizing strategy for one or more players.

A player takes decisions and makes moves to optimize its payoff (e.g. by maximizing reward or minimizing penalty). A player has none or many options and an option is available to one player. The player compares its options and takes a decision. A player may take none or many decisions while a decision is taken by one player. The player makes a move that operationalizes its decision. A player may make none or many moves while a move is made by one player. A move determines the payoff received by a player. A player receives one or more payoffs while a payoff is received by one player. The characteristics and features of Game Trees are described in [23].

In Figure 2, entities from i* and Game Tree are separated by labeled boundaries. This helps to show i* and Game Tree segments in the metamodel clearly and in a self-contained manner. Relationships that cross these boundaries depict semantics that are necessary for linking relevant i* and Game Tree entities. We follow original i* concepts [8] but include them here for the win-win method to be self-contained. Game Tree concepts from [23] are included to develop the Game Tree segment because a search of the literature did not yield a Game Tree metamodel.
### Modeling Phase

- Represent Stakeholders that are Concrete Actors as Agents and Abstract Actors as Roles
- Identify additional actors
- Identify goals for each actor
- Identify alternative tasks for achieving each goal
- Identify softgoals for each actor, with priorities
- Identify contributions from tasks to softgoals
- Identify Dependencies among actors
- Evaluate goal satisfaction by propagating labels
- Compute Payoffs for Decision Paths

### Evaluation Phase

- Evaluate goal satisfaction by propagating labels
- Compute Payoffs for Decision Paths
- Is there a win-win strategy?
- Add/Remove some actor
- Generate a change in some actor's goal
- Generate additional alternatives for achieving goals of some actor
- Generate a change in softgoals of some actor
- Generate a change in some actor's goal
- Generate a change in relationships among two actors

### Exploration Phase

- Add/Remove some actor
- Generate a change in some actor's goal
- Generate additional alternatives for achieving goals of some actor
- Generate a change in softgoals of some actor
- Generate a change in some actor's goal
- Generate a change in relationships among two actors

---

**Figure 1.** Process steps for alternating between i* and Game Tree modeling to get to win-win (Introduction of new actor is highlighted in **bold**, softgoal and tasks in *italics*)
In earlier work [5], [22] we argued that even though Game Trees supported the expression of payoffs they did not explicitly portray the reasons for those payoffs. The payoffs in Game Trees, typically represented as scalar values, are assumed to encode the rationale for their calculation but do not reveal that rationale overtly. This signifies a perceived limitation of Game Trees in terms of their explainability, interpretability, and transparency. We had demonstrated that the payoffs in Game Trees could be derived from the internal intentional structures of actors in i*.
Strategic Rationale (SR) diagrams [5], [22]. This allowed us to document the reasoning behind the calculation of payoffs in Game Trees by linking Game Trees with i* SR diagrams.

In [22] we had proposed a process for co-developing Game Tree and its complementary i* SR diagram. In [5], [6] we refined and elaborated this process as well as tested it by applying it to case examples of coopetition. In [5] we demonstrated one pathway for getting to win-win which was by generating additional alternatives for achieving goals of some actor. In [6] we demonstrated a different pathway for getting to win-win which was by adding another actor in the relationship among existing actors. In this article, we propose a metamodel for linking Game Tree with i* to formalize the semantic relationship between these modeling languages.

The process depicted in Figure 1 comprises three phases: Modeling, Evaluation, and Exploration. In the Modeling phase, an i* SR [8] diagram and its corresponding Game Tree are instantiated and populated. In the Evaluation phase, impacts of extant choices on objectives are calculated to detect any win-win strategies. In the Exploration phase, a systematic search is performed to generate new alternatives that yield positive-sum outcomes. In the real world, SMEs and domain specialists are expected to supplement their creativity and imagination with this process to synthesize insights from encoded knowledge (e.g. catalog in Section 4 and Section 5). This process can be repeated to generate as many win-win strategies as needed by decision-makers.

3.1 Modeling Phase

In this phase, strategic relationships among actors are modeled in terms of goals, tasks, resources, softgoals, and dependencies among them that are denoted in an i* SR diagram. The sequence of decisions and payoffs of these players are codified in a Game Tree. An actor is an active entity that performs actions by applying its know-how to accomplish its goals. Two kinds of actors are distinguished here, based on their manifestation in the world. A role is an abstract actor while an agent is a concrete actor. An agent can play none or many roles while a role can be played by none or many agents. An actor in i* represents a player in a Game Tree while a player in a Game Tree serves as an actor in i*. A goal is a state of affairs in the world that an actor wishes to achieve. An actor wants to achieve or satisfy none or many goals and none or many goals are wanted by an actor. None or many goals in i* determine the payoff in a Game Tree.

Task is a concrete method for addressing a goal and satisfying some quality requirements (i.e., softgoals). In i*, softgoals denote quality objectives that do not have clear-cut satisfaction criteria. They are evaluated as being satisfied or denied from the subjective perspective of an actor. None or many softgoals in i* influence the payoff in a Game Tree. Furthermore, the priority of a softgoal in i* impacts its influence on the payoff in the Game Tree.

Softgoals can be analyzed in terms of type and topic wherein type denotes the quality that is desired while topic denotes the intended behavior or structure that should encompass a desired quality [24]. Type and topic are depicted in Table 1, Table 2, Table 3, and Table 4 as well as Figure 3, Figure 4, Figure 5, and Figure 6 in the form of Type [Topic]. An actor wants to achieve or satisfy none or many softgoals and none or many softgoals are wanted by an actor.

A task is an activity that can be used to accomplish a goal. The relationship between a goal and its associated tasks is shown via means-ends links. A goal (the “end”) is achieved when any of its associated tasks (the “means”) are completed. An end may be associated with none or many means and a means may be related to none or one end. An option in a Game Tree represents a task in i* while none or one task in i* serves as an option in a Game Tree.

A task can be decomposed into subsidiary goals, tasks, softgoals, and resources. A resource is a physical or informational entity that is necessary for completing a task. The relationship between a task and its subsidiary entities is depicted via a task-decomposition link. A task may be comprised of none or many parts and this is portrayed using none or many task-decomposition
links. A task-decomposition link may be related to one task. A task-decomposition may comprise none or one task, goal, softgoal, or resource. Similarly, any task, goal, softgoal, or resource may be a part-of none or one task-decomposition.

In i*, actors are associated with each other using dependency links whereby a depender depends on a dependee for a dependum. A depender and dependee are actors wherein an actor can be none or many dependees and an actor can also be none or many dependers. A dependum can be a goal to be achieved, task to be completed, softgoal to be satisfied, or resource to be obtained. Similarly, a goal, task, softgoal, or resource can be associated with none or many dependency links.

Contribution links (see Figure 3) relate tasks to softgoals and softgoals to other softgoals. Contribution links can be of type Help (denoted by a green line accompanied with a plus symbol) or Hurt (denoted by a red line accompanied with a minus symbol). A Help contribution link contributes positively towards the achievement of a softgoal. A Hurt contribution link contributes negatively towards the achievement of a softgoal. Contributions can be intentional (denoted by a solid line) or incidental (denoted by a dashed line). A task may contribute to none or many softgoals and none or many tasks may contribute to a softgoal. A softgoal may contribute to none or many softgoals and none or many softgoals may contribute to a softgoal. Further details about i* modeling can be found in [8].

3.2 Evaluation Phase

In this phase, Contribution links are used to propagate and trace the impact of relatively lower-level tasks and softgoals on higher-level softgoals. Softgoals can either be fully satisfied (denoted by a checkmark) or partially satisfied (denoted by a dot underneath a checkmark). Conversely, softgoals can either be fully denied (denoted by a cross) or partially denied (denoted by a dot underneath a cross).

Forward propagation of labels can be used to answer ‘is this solution viable’ type of questions. The process for forward propagation of satisfaction labels in goal models is explained in [25]. This process involves the iterative application of propagation rules to attach current values from each offspring to its parent and then resolving softgoal labels at the parent level [25]. We apply the rules for satisfaction analysis in goal models that are explained in [26] in the Evaluation phase.

4 As-Is Scenario: Discriminating Win-Win Strategies with i* and Game Trees

We now illustrate the application of the method using the pharmaceutical industry example outlined in Section 2.

4.1 Modeling Phase

Figure 3 presents a goal model of an As-Is knowledge-sharing scenario between firms under coopetition. This goal model focuses on interdependencies among softgoals, and tasks that operationalize those softgoals while deferring consideration of relationships among actors. In this goal model, the nodes are softgoals or tasks while the edges are contribution links. Table 1 and Table 2 expand on the meanings of these softgoals and tasks.

In this industry scenario, a firm has two top-level softgoals which are No Leakage of knowledge assets and No Blocking of knowledge transfers. No Leakage of knowledge assets is a softgoal because separate firms may judge the presence or absence of knowledge leakage differently. Similarly, No Blocking of knowledge transfers is another softgoal because different firms may use dissimilar criteria to determine whether knowledge-sharing is being blocked.
A firm can adopt a Strict knowledge-sharing policy or a Permissive knowledge-sharing policy. A Strict policy prioritizes minimization of knowledge-leakage over circumvention of knowledge-blocking. Conversely, a Permissive policy treats avoidance of knowledge-blocking with greater importance than prevention of knowledge-leakage. In the knowledge sharing setting considered here, the same goal model applies equally to all sharing parties. In other settings, a separate goal model may be needed to represent the perspective of each actor.

Table 1. Softgoal types and topics in As-Is scenario in Figure 3

<table>
<thead>
<tr>
<th>Softgoal Type [Topic]</th>
<th>Description of softgoal</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Leakage [Knowledge Assets]</td>
<td>Assets should not be misappropriated by partners. [10], [11]</td>
</tr>
<tr>
<td>No Blocking [Knowledge Transfers]</td>
<td>Transfers should be seamless and frictionless. [27], [28]</td>
</tr>
<tr>
<td>Synergetic [Knowledge Assets]</td>
<td>Assets should be more valuable jointly than individually. [29], [30]</td>
</tr>
<tr>
<td>Leveragability [Knowledge Assets]</td>
<td>Assets should be useful and usable to generate benefits. [29], [30]</td>
</tr>
<tr>
<td>No Negative Cross Impact [A. Val.]</td>
<td>Sharing with partner should not reduce value of asset for self. [29], [30]</td>
</tr>
<tr>
<td>Complementarity [Partner Assets]</td>
<td>Partner assets should enhance each others asset value. [31]</td>
</tr>
<tr>
<td>Transferability [Knowledge Assets]</td>
<td>Assets should be distributable to partners. [32]</td>
</tr>
<tr>
<td>Appropriability [Knowledge Assets]</td>
<td>Assets should be receivable by partners. [14]</td>
</tr>
<tr>
<td>Irreducible [Asset Value]</td>
<td>Benefits from asset should be indestructible and renewable. [33]</td>
</tr>
<tr>
<td>Protectable [Knowledge Assets]</td>
<td>Assets should be containable and isolatable. [34]</td>
</tr>
<tr>
<td>Mutuality [Partner Assets]</td>
<td>Sharing should encompass assets that are inter-reliant. [35]</td>
</tr>
<tr>
<td>Annotatable [Asset Ownership]</td>
<td>Identity of the owner of each asset should be discernible. [21]</td>
</tr>
<tr>
<td>Combinable [Partner Assets]</td>
<td>Assets should be integrable with other assets. [36]</td>
</tr>
<tr>
<td>Compatible [Knowledge Assets]</td>
<td>Assets should function normally in conjunction with other assets. [37]</td>
</tr>
<tr>
<td>Available [Partner Assets]</td>
<td>Assets should be easily reachable when needed. [38]</td>
</tr>
<tr>
<td>Absorbable [Partner Assets]</td>
<td>Assets should be easily consumable when needed. [38]</td>
</tr>
<tr>
<td>Dynamic [Knowledge Assets]</td>
<td>Content and functionality of asset should be changeable. [33]</td>
</tr>
<tr>
<td>Concealable [Asset Content]</td>
<td>Asset contents should be capable of being hidden from partners. [21]</td>
</tr>
<tr>
<td>Licensable [Knowledge Assets]</td>
<td>Assets should support deactivation and decommissioning. [39]</td>
</tr>
</tbody>
</table>
Softgoals are operationalized by tasks (bottom of Figure 3). For instance, Processing involves generating machine-readable metadata for each knowledge asset. This makes it easier to distinguish among individual knowledge assets such as based on their ownership. Therefore, Processing is a task that operationalizes the softgoal Annotatable asset ownership. Similarly, Integrating involves mixing together knowledge assets from various partners. This makes it simpler for each firm to avail of the knowledge of their partners. Therefore, Integrating operationalizes the softgoal Available partner assets.

In this example, we use the notation wherein the inclusion of a task in a Strict or Permissive policy is inscribed within each task. A circle inscribed with an S and a numerical identifier in the top left corner of a task denotes the inclusion of that task in a Strict policy. A square inscribed with a P and a numerical identifier in the top right corner denotes the inclusion of that task in a Permissive policy. For instance, Auditing of knowledge transfers is a part of a Strict policy and Integrating of partner assets is a part of a Permissive policy.

A task can also be included simultaneously in Strict and Permissive policies while being implemented differently in each policy type. For instance, Modularizing the boundary of a knowledge asset is part of both Permissive as well as Strict policies even though modularization may be implemented differently in Strict and Permissive policies. It should be noted that these inscriptions (i.e. S with identifier in circle on top left of task and P with identifier in circle on top right of task) are specific to this example.

Table 2. Task types and topics in As-Is scenario in Figure 3

<table>
<thead>
<tr>
<th>Task Type [Topic]</th>
<th>Policy</th>
<th>Description of task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auditing [Knowledge Transfers]</td>
<td>S</td>
<td>Reviewing actions performed by users and processes. [21]</td>
</tr>
<tr>
<td>Processing [Asset Metadata]</td>
<td>S</td>
<td>Generating machine-readable metadata for each asset. [40]</td>
</tr>
<tr>
<td>Exposing [Asset Interface]</td>
<td>P</td>
<td>Registering input and output parameters of an asset. [41]</td>
</tr>
<tr>
<td>Documenting [Asset Schema]</td>
<td>P</td>
<td>Explaining types of entities and relationships in an asset. [32]</td>
</tr>
<tr>
<td>Integrating [Partner Assets]</td>
<td>P</td>
<td>Commingling content from disparate partner assets. [42]</td>
</tr>
<tr>
<td>Publishing [Asset Directory]</td>
<td>P</td>
<td>Advertising sharing of an asset via a repository. [41]</td>
</tr>
<tr>
<td>Modifying [Asset Behavior]</td>
<td>S</td>
<td>Reprogramming the content and functionality of an asset. [43]</td>
</tr>
<tr>
<td>Modularizing [Asset Boundary]</td>
<td>S, P</td>
<td>Setting perimeter of each asset specifying its scope. [44]</td>
</tr>
<tr>
<td>Reconfiguring [Knowldg. Assets]</td>
<td>S</td>
<td>Asset should be amenable to packaging in many ways. [45]</td>
</tr>
</tbody>
</table>
Figure 3. Goal model of As-Is scenario representing knowledge sharing goals and potential tasks, synthesized from sources listed in Table 1 and Table 2
Goal models aid in detecting and analyzing tradeoffs that exist among different softgoals. The goal model in Figure 3 shows that various tasks impact softgoals differently. For instance, posting a knowledge asset into an asset directory helps to make that knowledge asset more combinable (i.e., easier to integrate) with other knowledge assets. Conversely, modifying the behavior of a knowledge asset can make it less compatible with knowledge assets with which it is already interoperable (i.e., hurts link).

Specific combinations of tasks within a strict or permissive policy can also impact softgoals differently. For instance, auditable is a task that is part of a strict policy and operationalizes the softgoal mutuality of partner assets. It also helps the softgoal licensable knowledge assets. Similarly, reconfiguring knowledge assets is a task that is also part of a strict policy and operationalizes the softgoal licensable knowledge assets. This softgoal licensable knowledge assets is considered as satisfied in a strict policy since multiple tasks that are part of a strict policy make positive contributions to it. Conversely, the softgoal dynamic knowledge assets is only partially satisfied in a strict policy due to the conflicting interaction of two tasks which are part of a strict policy. These are modifying asset behavior and processing asset metadata. While modifying asset behavior operationalizes the softgoal dynamic knowledge assets this softgoal is hurt by processing asset metadata.

(i) i* SR diagram showing strategic goals of the two coooperating actors

(ii) Game Tree showing payoffs from possible moves by BPC followed by GPC moves

**Figure 4.** As-Is scenario
In the real world, each actor assesses such trade-offs between softgoals in line with its preferences and prioritizes those softgoals differently depending on its proclivities. The goal model in Figure 3 is instantiated in Figure 4 to demonstrate this with respect to two actors in a coopetitive relationship. Figure 4 depicts co-developed i* SR diagram and Game Tree of the As-Is scenario pertaining to two business partners in the pharmaceutical industry.

In this i* diagram, Branded Pharmaceutical Company (BPC) and Generic Pharmaceutical Compounder (GPC) are two actors. BPC develops and markets prescription medicines based on its R&D initiatives as well as its protected intellectual property (IP) (not shown†). GPC manufactures ingredients that are used in BPC’s medicines and produces medicines for BPC that BPC sells in the market (not shown†). GPC also sells generic medicines that are analogous to the prescription medicines sold by BPC only if their IP is not protected (not shown†). In this hypothetical example, the goal structures of both actors appear to be identical but their softgoals have different priorities. This is done to demonstrate the impact of softgoals’ priorities on the calculation of payoffs when actors have seemingly identical goal configurations but dissimilar softgoal priorities.

These two actors depend on each other to meet their respective goals pertaining to Know-how be Gained. GPC depends on Market Forecasts of BPC (shown) so that GPC can approximate the upcoming requirements of BPC (not shown†). This helps GPC to plan its production runs based on medicines that BPC will likely contract GPC to produce (not shown†). BPC depends on the Production Traces of GPC (shown) to verify that GPC is only manufacturing those quantities of ingredients of BPC’s high margin medicines that are ordered by BPC (not shown†). This helps BPC to verify that GPC is not manufacturing extra quantities of those ingredients to produce substitute medicines that GPC can sell by itself (not shown†).

Dependencies among BPC and GPC are shown as softgoals because each is satisfied from the perspective of the depender. Both actors can achieve their respective goals of Know-how be Gained by performing the task Share Knowledge. Knowledge sharing Policy be adopted is a sub-goal of this task Share Knowledge. This sub-goal is associated with two tasks which pertain to the adoption of either a Strict or a Permissive knowledge sharing policy. The tasks labeled Strict Policy and Permissive Policy for knowledge sharing in Figure 4 map to the set of tasks in Figure 3 with the inscriptions of S and P respectively. This is shown in Figure 4 via the decomposition of two tasks, which are Strict Policy and Permissive Policy, into their respective sub-tasks, which are denoted by P1...Pn and S1...Sn. Contributions from the tasks labeled Strict Policy and Permissive Policy to softgoals labeled Synergetic knowledge assets, Leverageability of knowledge assets, and No negative-cross impact of asset value are depicted indirectly via a partially dotted contribution link. This is done to hide the full intentional structure in the i* SR diagram since the complete goal model in Figure 3 contains these details.

Potential benefits from knowledge sharing serve as incentives for BPC and GPC to adopt Permissive policies. However, the countervailing threat of opportunism serve as motivations for BPC and GPC to adopt Strict policies. Since BPC and GPC are autonomous actors, they are free to select either Permissive or Strict policy in line with their preferences and proclivities. In this example, as shown in Figure 4, BPC prioritizes a Strict policy over a Permissive policy while GPC prioritizes a Permissive policy over a Strict policy. The selection of one policy over another in the real-world is likely to be the result of deliberation and contemplation by subject matter experts (SMEs) and domain specialists. This modeling approach complements and supplements their reasoning and analysis rather than substitute or obviate it.

†In this instance, and in the remainder of this article, certain aspects of the relationship between actors are not shown in order to simplify the visual presentation of the models.
4.2 Evaluation Phase

In the Evaluation phase, payoffs in the Game Tree are estimated by analyzing softgoal satisfaction in the i* SR diagram. A preliminary analysis of softgoal satisfaction in the goal model in Figure 3 reveals that neither Strict nor Permissive knowledge-sharing policies satisfy all top-level softgoals in the As-Is scenario. The i* SR diagram in Figure 4 shows that neither BPC nor GPC satisfy every softgoal through their chosen policies. For instance, BPC is not able to satisfy one of its top-level softgoals of No Blocking of knowledge transfers by choosing a Strict policy while GPC is not able to satisfy one of its top-level softgoals of No Leakage of knowledge assets by choosing a Permissive policy. The i* SR diagram in Figure 4 can be used to calculate the relative payoffs for these players in the Game Tree.

On the Game Tree, in the first case, BPC and GPC select Permissive policies. Since GPC prioritizes a top-level softgoal that is satisfied when this type of policy is chosen then it earns a payoff of 1. However, BPC prioritizes a top-level softgoal that is denied when this policy is chosen then it earns a payoff of -1. In the second case, BPC selects a Permissive policy but GPC selects a Strict policy. In this case neither BPC nor GPC achieve their higher priority top-level softgoals and thus both earn payoffs of -1. In the third case, BPC selects a Strict policy but GPC selects a Permissive policy. In this case while both BPC and GPC satisfy their higher priority top-level softgoals they do not satisfy some of their, albeit lower priority, softgoals. Thus, both earn payoffs of 0. In the fourth case, BPC and GPC select Strict policies. Since BPC prioritizes a top-level softgoal that is satisfied when this type of policy is chosen then it earns a payoff of 1. However, GPC prioritizes a top-level softgoal that is denied when this type of policy is chosen then it earns a payoff of -1.

These payoffs in the Game Tree can be used to detect the presence of any positive-sum outcomes. In the As-Is scenario, there are no win-win strategies since neither Permissive nor Strict policies allow BPC and GPC to satisfy each of their top level softgoals. This motivates their systematic search for new alternatives to generate positive-sum outcomes.

5 To-Be Scenario: Generating Win-Win Strategies with i* and Game Trees

In Section 4.1 and Section 4.2 we discussed the As-Is configuration of the knowledge sharing relationship between BPC and GPC from modeling and evaluation perspectives respectively. The evaluation phase shows that there are no win-win strategies available to BPC and GPC in the As-Is configuration.

In this section we discuss the exploration and generation of new win-win strategies by BPC and GPC with the support of modeling. These new strategies are predicated on the creation of additional quality objectives as well as new methods for addressing those requirements. Realization of these methods for satisfying new quality requirements necessitates the introduction of an intermediary actor in the relationship between BPC and GPC. Through modeling, we demonstrate the development of win-win strategies for BPC and GPC in the To-Be configuration.

5.1 Exploration Phase:

In the Exploration phase, an SME can pursue any of five non-deterministic lines of action incrementally and iteratively. As depicted in Figure 1, they can add/remove some actor, generate additional alternatives for achieving goals of some actor, generate a change in relationships among some actors, generate a change in softgoals of some actor, or generate a change in some actor’s goals. For example, as shown in the goal model in Figure 5, new softgoals and tasks can be introduced that favorably impact (i.e. Help) top-level softgoals. These new softgoals and tasks can be used to satisfy previously denied top-level softgoals.
Figure 5 is a goal model of a hypothetical To-Be knowledge-sharing scenario between businesses under coopetition. Model elements, from the As-Is scenario in Figure 3, that are unimpacted by new softgoals and tasks in Figure 5 are greyed-out. This improves the presentation of the goal model to highlight the To-Be scenario. New softgoals and tasks in Figure 5 are shown in blue color while existing softgoals that are impacted by new softgoals and tasks are shown in black color. New contribution links are shown in green (Help) and red (Hurt) colors while existing contribution links are greyed-out. We anticipate that, with tool support in the future, one would be able to collapse or expand portions of the model to hide or reveal details as necessary.

Loops in the process depicted in Figure 1 indicate that any step in the Exploration phase of this modeling approach can trigger other steps. For example, in the pursuit of a win-win strategy, an SME may decide to generate new tasks to improve overall satisfaction of top-level softgoals. These new tasks, depicted in Figure 5, may trigger the generation of new softgoals. Collectively, these additional tasks and softgoals represent new system requirements that expand the set of existing system requirements depicted in Figure 3.

Table 3 and Table 4 describe these new softgoals and tasks along with their sources. A comparison of the As-Is and To-Be scenarios reveals a contrast between the softgoals and tasks in these scenarios. Each of the softgoals (Table 1) and tasks (Table 2) in the As-Is scenario can be achieved by BPC and GPC without requiring support from any other actor. However, certain softgoals (Table 3) and tasks (Table 4) in the To-Be scenario cannot be satisfied by BPC and GPC alone. These softgoals and tasks in the To-Be scenario necessitate the involvement of an intermediary actor in the relationship between BPC and GPC.

For example, the softgoal Compliant knowledge assets requires format of knowledge assets to be consistent with third-party specifications. This means that an intermediary actor that publishes specifications as well as certifies compliance of knowledge assets with those specifications is required. Similarly, the task External Tracking of knowledge transfers entails surveilling content in inter-partner transfers and such tracking is performed by an intermediary actor that is external to BPC as well as GPC. The task Certifying of the asset specification involves attesting system specification by standards organization and here the standards organization represents an actor that is neither BPC nor GPC.

5.2 Evaluation Phase

The i* SR diagram in Figure 6 can be used to calculate the relative payoffs for the players in the Game Tree. In the first case, BPC and GPC select Permissive policies. Since all top-level softgoals of GPC are satisfied and it acts in accordance with its preference (i.e. adopts Permissive policy) then it earns a payoff of 2. Each top-level softgoal of BPC is also satisfied in this case but since it does not act in line with its preference (i.e. does not adopt Strict policy) then it earns a payoff of 1. In the second case, BPC selects a Permissive policy but GPC selects a Strict policy. In this case both BPC nor GPC achieve their higher priority top-level softgoals but neither acts according to their preferences and thus both earn payoffs of 1. In the third case, BPC selects a Strict policy but GPC selects a Permissive policy. In this case both BPC and GPC satisfy each of their higher priority top-level softgoals and act according to their preferences. Therefore, both earn payoffs of 2. In the fourth case, BPC and GPC select Strict policies. Since all top-level softgoals of BPC are satisfied and it acts in accordance with its preference (i.e. adopts Strict policy) then it earns a payoff of 2. Each top-level softgoal of BPC is also satisfied in this case but it does not act in line with its preferences (i.e. does not adopt Permissive policy) so it earns a payoff of 1.
Figure 5. Goal model of To-Be scenario representing knowledge sharing goals and potential tasks, synthesized from sources listed in Table 1, Table 2, Table 3, and Table 4.
Table 3. Softgoal types and topics and topics in To-Be scenario in Figure 5

<table>
<thead>
<tr>
<th>Softgoal Type [Topic]</th>
<th>Description of softgoal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balanced [Asset Sharing]</td>
<td>Quantity of contents transferred must be equal among partners [46].</td>
</tr>
<tr>
<td>Reportable [Asset Sharing]</td>
<td>Quantity and quality of contents transferred must be auditable [47].</td>
</tr>
<tr>
<td>Compliant [Knowledge Assets]</td>
<td>Format of assets must be consistent with third-party specifications [48].</td>
</tr>
<tr>
<td>Redundant [Knowledge Assets]</td>
<td>Copies of assets must be stored for safeguarding [49].</td>
</tr>
</tbody>
</table>

Table 4. Task types and topics in To-Be scenario in Figure 5

<table>
<thead>
<tr>
<th>Task Type [Topic]</th>
<th>Policy</th>
<th>Description of task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metering [Knowledge Transfers]</td>
<td>S</td>
<td>Measuring quantity of inter-partner transfers [50].</td>
</tr>
<tr>
<td>External Tracking [Knowledge Transfers]</td>
<td>P</td>
<td>Surveilling content in inter-partner transfers [51].</td>
</tr>
<tr>
<td>Canonical Template [Knowledge Model]</td>
<td>S</td>
<td>Establishing uniform format to be used by partners [52].</td>
</tr>
<tr>
<td>Certifying [Asset Specification]</td>
<td>P</td>
<td>Attesting system specification by standards organization [53].</td>
</tr>
<tr>
<td>Replicating [Knowledge Assets]</td>
<td>S, P</td>
<td>Creating multiple copies of asset [54].</td>
</tr>
</tbody>
</table>

The i* SR diagram of the To-Be scenario shows that all the top-level softgoals of BPC and GPC are satisfied. This is due to their addition of new softgoals and tasks as well as the introduction of a new actor, which is Industrial Data Space (IDS). IDS is a data-sharing initiative for industry and academic organizations [55]. IDS provides an architecture, blueprint, standard, and platform for data-sharing among member organizations in a reliable, transparent, compliant, and accountable manner [56]. IDS functions as an intermediary actor that enables member organizations to share knowledge in a fair and seamless manner by enabling the formation and enforcement of data-sharing commitments and obligations [57].

IDS affords its member organizations with an alternative to ad-hoc data-sharing arrangements. Ad-hoc data-sharing can expose partners to various risks including loss of competence and leakage of technology [58]. To mitigate such risks, organizations can utilize mechanisms to monitor, regulate, and secure data transfers across organizational boundaries [59]. IDS enables its member organizations to enshrine the terms and conditions of their data-sharing agreements into executable commitments. These agreements are implemented and enacted by an intermediary actor thereby minimizing such risks and uncertainties.

IDS constitutes several modules that offer many features to its users [60]. IDS comprises a Connector module that offers pre-defined data templates with mappings between heterogeneous data schemas. These templates can be used for meaningfully interconnecting disparate systems. If existing templates are unavailable for certain systems (e.g. legacy or proprietary) then users can develop custom templates by following the IDS blueprint for building templates. Such data templates can be shared for reuse because they are compatible with the IDS standard and specification.
IDS provides a Catalog functionality that allows users to list their data catalog. A catalog serves as an index that can be searched or browsed to identify potential sources of data that are...
needed by users. This use case positions IDS as a marketplace wherein buyers and sellers of data can transact with each other in a trusted environment. Providers of data can advertise datasets that they are willing to transfer along with relevant terms and conditions. Consumers of datasets can find the datasets they need and then bargain and negotiate with prospective suppliers on the platform. The data catalog is a key component of this marketplace.

IDS includes a Logging system that is necessary for tracking the sharing of data between actors. Details about content and scope of the datasets that are accessed as well as volume and variety of data that are transferred is recorded. Monitoring of data sharing is necessary for ensuring that actors only obtain data they are authorized to access. Data are also encoded at the source with metadata to define their terms of use and specify their permitted use cases. Tracking the application of transferred data is needed to verify that data are only utilized for purposes that are agreed to by the relevant actors.

IDS consists of a Reporting system that is useful for generating data reports to analyze compliance of the actors with their data-sharing commitments as well as obtain insights related to data transfers. Users can analyze metrics at the operational level, key performance indicators (KPIs) at the tactical level, and critical success factors (CSFs) at the strategic level. IDS users can generate data reports to perform historical analysis. They can also use the raw data logged by the Reporting system to train predictive or prescriptive models.

The payoffs associated with the To-Be scenario in the Game Tree reflect higher values than their corresponding options in the As-Is scenario due to the benefits and advantages afforded by IDS to BPC and GPC. Following the process described in Figure 1 shows that multiple win-win strategies can be created in an industrial collaboration scenario where none existed originally. A comparison of Figure 3 and Figure 4 with Figure 5 and Figure 6 highlights a primary benefit of using this approach to co-develop i* SR diagrams with Game Trees.

In Figure 4 and Figure 6, the Game Trees are structurally similar but have different payoffs and, in the i* SR diagrams, the internal intentional structure of BPC and GPC is identical except for certain contribution links. Figure 3 and Figure 5 are crucial for understanding the reasons for these differences. The goal models in Figure 3 and Figure 5 explain the reasons for the differences in the payoffs on the Game Trees and the changes in the contribution links within the i* SR diagrams.

6 Related Work

This article contributes to the body of knowledge pertaining to intentional modeling of strategic coopetition. Majority of the research on coopetition modeling has focused on game-theoretic approaches [4]. Such approaches encode the intentionality of the players within the payoffs thereby eliding their goal structures. Recent research in the enterprise modeling literature has focused on the intentionality of actors engaged in strategic coopetition. [4], [61] include detailed reviews of coopetition research from scholarly literature in Strategic Management and Economics domains. This knowledge is used to design requirements for enterprise modeling of strategic coopetition that are presented in [4], [61]. As discussed in Section 1 and Section 2, the intentional modeling approach that is applied in this article was introduced in [22] and refined in [5] and [6].

In [5] a basic example of cake-cutting is presented to demonstrate the application of this process. That example of cake-cutting is drawn from game theory and is used to demonstrate the co-design and co-evolution of i* SR diagrams and their corresponding Game Trees. That example shows the introduction of a new alternative in an ultimatum game between two players to generate a new win-win strategy when originally none existed. That pathway to win-win is further illustrated with a case of coopetition between software ecosystems of Apple and Adobe. Modeling of complementarity, which is a motivator of coopetition, and relevant in knowledge-
sharing scenarios, is discussed in [62]. More broadly, this research article also contributes to the scholarly literature on enterprise modeling of business strategy.

Researchers of conceptual modeling have developed modeling techniques that incorporate concepts from strategic management [63]–[67]. Giannoulis et al. compare various enterprise modeling approaches for representing business strategy concepts [63]. The business strategy concepts covered in [63] do not include the notions of win-win strategies or positive-sum games. López & Franch [64] propose a framework for developing Business Strategy Models (BSM) of Open Source Software (OSS) organizations using i*. The BSM framework [64] focuses solely on OSS organizations and its application is not demonstrated in the context of other business settings. Bleistein et al. [65] offer an approach for developing conceptual models of strategies for e-business systems. An explicit demonstration of this approach [65] in terms of its application to non e-business system scenarios is not offered.

Giannoulis et al. [66] posit a meta-model to formalize the concepts related to Strategy Maps (SM). SMs are based on the Balanced Scorecard (BSC) approach that focuses on specific aspects of an organization (Learning and Growth, Internal, Financial, Customer). SMs and BSCs focus on the creation and appropriation of value by an organization rather than the development and realization of win-win strategies by multiple organizations in an ecosystem. Osterwalder et al. [67] elucidate the Business Model Canvas (BMC) approach which encompasses nine building blocks of an organization’s business model. However, while business models and business strategies are interrelated – they are conceptually distinct entities. Since the BMC approach mostly focuses on the internal aspects of an organization it does not cover business strategies that manifest at the external interfaces of that organization.

Conceptual modeling researchers have also developed modeling techniques that incorporate concepts from entrepreneurship [68], [69], [70]. Pant et al. [68], propose a framework for representing and reasoning about pivoting by startups and large enterprises. Pivoting is a strategic endeavor that requires an organization to make and undertake significant changes to its strategy and business model. We [69] propose a modeling approach for expressing and evaluating a specific type of pivot (i.e. Larger Goal pivot). In this type of pivot an organization attempts to generate new lower-level options for achieving a higher-level goal. Such a pivot is necessary when the available lower-level options are inadequate for satisfying a higher-level goal. We [70] also utilize conceptual modeling to articulate and analyze a revenue model pivot. While some pivots are motivated by the pursuit of win-win strategies – others result from tactical or operational factors such as internal restructuring or reorganization. This shows that while pivoting is related to the pursuit of win-win strategies it is a different conceptual entity.

7 Conclusions and Future Work

We utilized a strategic modeling approach to systematically search for win-win strategies and generate new alternatives for organizations under coopetition. This integrative approach incrementally and iteratively elaborated and refined the i* SR diagram and its corresponding Game Tree. No win-win strategies were detected in the As-Is scenario due to threats related to knowledge leakage and knowledge blocking. However, in the To-Be scenario, multiple win-win strategies were generated by applying this strategic modeling approach to the As-Is scenario. New softgoals and tasks were added that obviated the threats from knowledge leakage and knowledge blocking. These softgoals and tasks could be satisfied by the actors by themselves (e.g. by building a system that meets necessary requirements) or with the help of another actor (e.g. by subscribing to a service that meets necessary requirements). In this article we depicted the latter option with reference to IDS serving an intermediary actor.

This strategic modeling approach incorporates three practical and reasonable assumptions to ensure its usefulness in real-world applications [22]. However, the efficacy and viability of these assumptions needs to be tested via empirical investigation. Our future work will comprise
achievement of three objectives that must be satisfied to encourage mainstream adoption of these models by industry professionals.

Firstly, these models may need to be simplified to gain broader acceptance by practitioners. This would be done by developing collaboration patterns that represent common behaviors in the real world (e.g. collaborating to avoid common threat). Secondly, these models may need to support more sophisticated and nuanced methods for calculating payoffs. Game Theorists have proposed many methods for calculating payoffs under different circumstances and these methods could be supported by these models. Thirdly, these models may need to be commingled with existing processes that are used by organizations to manage coopetitive relationships. For instance, organizations use contracts and legal agreements to set the terms and conditions of such relationships. These models could be used to support the contract negotiation and agreement formation processes. Fourthly, the goal models presented in Figures 4 and 6 depict identical goal hierarchies within both actors (albeit with different softgoal priorities). This is done to show the impact of dissimilar softgoal priorities on the calculation of payoffs. In future work, we plan to depict the calculation of payoffs for different actors resulting from dissimilar goal hierarchies within actors. These areas of future work shall increase the value and utility of these models in the industry.

In this article, we adopted original i* [8] instead of iStar 2.0 [71] to take advantage of the separation between means-ends (OR) and task-decomposition (AND) links. In i* 1.0, goals (means) are related to tasks (ends) with means-ends (OR) links while parent tasks are related to their child elements (i.e. task with sub-task, sub-goal, sub-softgoal, resource) with task-decomposition (AND) links. Our future work also involves the adaptation of the metamodel and methodology presented in Section 3 to comport with the semantics of iStar 2.0 core [71] because iStar 2.0 core is designed to provide a simpler and more learnable experience to modelers.

References


