# Collaborative Design Between Intelligent Agents Through Resource Sharing

Jules Chenou(✉), Tonya Fields, and Aurelia Williams

Cybersecurity Complex, Norfolk State University, Norfolk, VA 23504, USA
{jchenou,tlfields,atwilliams}@nsu.edu
https://www.nsu.edu/Academics/Research/Cybersecurity

**Abstract.** The integration of intelligent agents is not only transforming but also revolutionizing collaborative design. This work goes deep into detail on the complexities of collaborative design processes among intelligent agents. This topic represents the frontier of fast-evolving distributed systems and artificial intelligence research. Equipped with sophisticated algorithms and learning mechanisms, intelligent agents are increasingly harnessed for complex problem-solving and creativity to support human designers or operate autonomously in collaborative environments.

Thus, the 'resource sharing' abstract is a powerful tool, covering a wide spectrum of resources, including data, computational power, expertise, and decision-making capabilities. Resource sharing enables intelligent agents to combine their capabilities to effectively amplify strengths and mitigate weaknesses in achieving collective objectives efficiently and effectively. The present paper discusses some of the aspects of resource sharing in collaborative design settings, focusing on mechanisms for sharing and allocation.

Strategies, negotiation protocols, and coordination mechanisms all bear significant practical implications and promise a more collaborative and efficient future. Interoperability, privacy concerns, conflicting objectives, and trust among agents are daunting but manageable hurdles in this race. This paper further explores the exciting challenges and promising collaborative design opportunities among intelligent agents. They are not roadblocks but opportunities for growth and learning. On the other hand, opportunities are not stepping stones but paths to success. They bring possibilities to exploit various competencies, tap collective genius, and realize bigger innovation and problem-solving success. These possibilities are not an aspiration of some distant future but a tangible reality we can shape and explore.

The theoretical bases, practical methodologies, and emerging trends in the domain also indicate the possibility of further research and development in collaborative design methodologies. Such a broad understanding

This material is based upon work supported by the National Science Foundation under grant no. 2229876 and is supported in part by funds provided by the National Science Foundation, the Department of Homeland Security, and IBM. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or its federal agency and industry partners


H. R. Arabnia et al. (Eds.): CSCI 2024, CCIS 2501, pp. 254–267, 2025.
https://doi.org/10.1007/978-3-031-90341-0_19

will eventually provide the knowledge and perspectives necessary to conduct business in this ever-evolving domain.

Moreover, this paper delves into the thrilling challenges and promising collaborative design opportunities among intelligent agents. The challenges, akin to hurdles in a race, such as interoperability, privacy concerns, conflicting objectives, and trust among agents, are daunting and manageable. They are not roadblocks but opportunities for growth and learning. Conversely, the opportunities are not just stepping stones to success; they are the path to success. They offer the potential to leverage diverse expertise, harness collective intelligence, and achieve higher innovation and problem-solving efficacy.

By leveraging interdisciplinary research in artificial intelligence, multi-agent systems, design theory, and cognitive science, this paper comprehensively examines collaborative design facilitated by resource-sharing among intelligent agents. It illuminates the theoretical foundations, practical methodologies, and emerging trends in this domain and highlights the potential for further research and development in collaborative design methodologies. This comprehensive understanding will equip one with the knowledge and perspectives necessary to navigate this ever-evolving domain.

**Keywords:** Information-sharing · Intelligent Agents · Regular Theory · Agent-Interactions

## 1 Introduction

Resource sharing enhances general efficiency, as agents use other's resources, solve problems in less time, or use available resources better. Resource sharing also allows intelligent agents to avoid infrastructure duplication and, therefore, save resources, mainly when these resources are rare or costly [1]. Resource sharing by agents enables them to scale up and down their capabilities dynamically according to demand, hence flexibility and adaptability in resource allocation. Agents enable cooperation by borrowing resources from each other in case of need and working together toward common goals [3].

[4] points out that resource pooling and expertise in resource sharing systems provide a comprehensivesynergistic effect that agents cannot achieve themselves; tthus,the,the same synergyconducive to better problem solving, agents offer novel approaches to solutions. Hence, optimism encourages innovation in processing complex tasks. However, this sharing of resources embeds dependencies among agents and presents a possible point of vulnerability should such provider resources become unavailable or fail to deliver. These dependencies allow disruptions and instability in the system because the dependent agents cannot continue their operations. In addition, resource sharing introduces privacy and security concerns, especially when sensitive or proprietary information is involved. Whether data breaches or information leakage, unauthorized access to it is always a potential risk during resource sharing.

Another challenge facing intelligent agents in dynamic and heterogeneous environments is resource management. Resources can be conflicts of use, leading to rivalry, inefficiency, or unfair allocation [4]. A further complication that resource sharing could impose on system performance is the presence of bottlenecks if the network has low bandwidth or high latency, thus causing delays, reduced throughput, and general degradation in system performance [5]. The coordination of resource allocation, access control, and sharing protocols by multiple intelligent agents complicates the design and implementation of a system. Coordinating such mechanisms requires sophisticated means that guarantee the fairness and efficiency of resource distribution without compromising security or performance.

### 1.1 Methodologies

Classification, infomorphism, and channel theory [6] are the primary theoretical lens through which this paper approaches how an intelligent agent might allocate and employ resources. Unlike Shannon's information theory [7], which deals with measurement and quantization aspects of information measurement by entropy and communication efficiency, the framework used here is more symbol-based and qualitative. This shift is essential to describe the communication of agents in scenarios that involve their interactions in the task-oriented setting, where the content and meaning of the information are more critical than its probability distribution.

Within this framework, information is represented as structured statements, such as affirming that agent $i$ can perform task x and that y is an accessible resource. By reading these statements, a classification table with a knowledge base for each agent provides the list of tasks that can be done, the capability needed, and the resources to accomplish them.

Interactions between agents are described through infomorphisms, creating pathways for conveying other agents' capabilities, resources, and constraints. In this regard, a channel cannot be looked at simply as a conduit that facilitates the transfer of information; it represents an official framing of how information gets passed between agents to compose information that is consonant with the way the agents classify information internally and consonant with the tasks at hand .

As such, the proposed systematic approach empowers the researchers to unambiguously articulate the foremost questions within the systems-to substantively theorize about the strategic positioning of the resources and the kinds of interactions and information processing by the agents. Each agent has its own type and methodical theory defining its activity. Furthermore, infomorphisms are an implicit mathematical operation that determines how this theory is aligned with other system elements. It also encourages an improved understanding of cooperative patterns in which an agent's behavior and resource usage depend on the information shared with them by the other agents.

## 2 Basic of Category Theory

### Category Theory, Functors and Natural Transformation

A category $\mathcal{C}$ consists of a class of objects and a class of morphisms (or arrows or maps) between the objects. Each morphism $f$ has a unique source object $a$ and target object $b$; we write $f : a \rightarrow b$. The composition of $f : a \rightarrow b$ and $g : b \rightarrow c$ is written as $g \circ f$ and is required to be associative: if in addition $h : c \rightarrow d$, then $h \circ (g \circ f) = (h \circ g) \circ f$. It is also required that, for every object x, there exists a morphism $1_x : x \rightarrow x$ (the identity morphism for x) such that, for every morphism $f : a \rightarrow b$, we have $1_b \circ f = f = f \circ 1_a$. These properties show that precisely one identity morphism exists for every object. A functor from one category to another is a structure-preserving mapping that preserves the identity and composition of morphisms. More exactly, if $\mathcal{C}$ and $\mathcal{D}$ are categories, then a functor $\mathbf{F}$ from $\mathcal{C}$ to $\mathcal{D}$ is a mapping that associates with each object $x \in Obj(\mathcal{C})$ an object, $\mathbf{F}(x) \in \mathcal{D}$ and, with each morphism $f : x \rightarrow y \in \mathcal{C}$, a morphism $\mathbf{F}(f) : \mathbf{F}(x) \rightarrow \mathbf{F}(y) \in \mathcal{D}$ such that $\mathbf{F}(id_x) = id_{\mathbf{F}x}$ for every object $x \in \mathcal{C}$, and $\mathbf{F}(g \circ f) = \mathbf{F}(g) \circ \mathbf{F}(f)$ for all morphisms $f : x \rightarrow y$ and $g : y \rightarrow z$.

In category theory, a commutative diagram is a diagram of objects (as vertices) and morphisms (arrows between objects) such that all directed paths in the diagram with the same start and end points lead to the same result by composition. The classic presentation of category theory can be found in [7]. Two reasonably comprehensive and rigorous texts accessible to most readers with mathematical backgrounds in classical engineering are [8,9]. [10] provides a light introduction, while [11,12] are category-theory texts addressed specifically to computer scientists; [13] addresses category theory in the context of software engineering (Fig. 1).

### Pushout

In category theory, a pushout is a construction that allows you to glue two objects together along a common *subobject*. Formally, given a diagram in a category consisting of three objects A, B, and C, and morphisms $f : A \leftarrow C$ and $g : B \leftarrow C$, the pushout of this diagram is an object $P$ along with morphisms $i : A \rightarrow P$ and $j : B \rightarrow P$ such that the following conditions hold: 1) $i \circ f = j \circ g$ 2) P is universal concerning the above property, meaning that for any other object Q and morphisms $i' : A \rightarrow Q$ and $j' : B \rightarrow Q$ satisfying the same conditions as $i$ and $j$, there exists a unique morphism $k : P \rightarrow Q$ such that $k \circ i = i'$ and $k \circ j = j'$. For example, to compute a pushout in the category Set of sets and functions, we follow these steps: 1) Begin by forming the disjoint union of A and B, denoted $A \sqcup B$. This set consists of all elements of A and B, treating them as disjoint, i.e., there are no common elements between A and B. 2) Identify and "glue together" elements that are equivalent under the functions $f$ and $g$. Specifically, for each element $c \in C$, find all pairs $(a, b)$ where $f^{-1}(a) = g^{-1}(b) = c$. Then, consider all such pairs as equivalent and merge them into a single element in the pushout set P. 3)The pushout set P is constructed by taking the disjoint union $A \sqcup B$ and collapsing equivalent elements identified in the previous step. This means that equivalent elements under $f$ and $g$ are considered the same in P. and, Finally, the inclusion maps $i : A \rightarrow P$ and $j : B \rightarrow P$ by mapping elements of A and B to their corresponding elements in P obtained after merging equivalent elements.

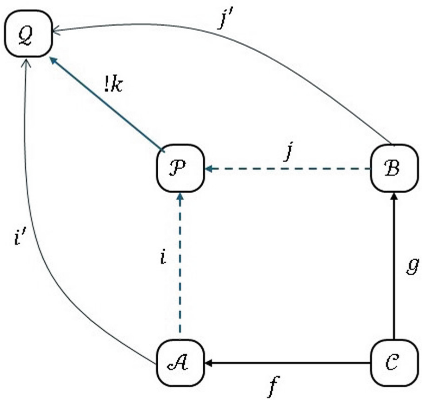


**Fig. 1.** A pushout diagram.

The resulting set P, along with the inclusion maps $i$ and $j$, constitutes the pushout of the diagram $A \rightarrow C \leftarrow B$ in the category Set. To illustrate, consider an example where $A = \{1, 2\}$, $B = \{3, 4\}$ and $C = \{1, 3\}$ with functions $f : A \rightarrow C$ and $g : B \rightarrow C$ such that $f(1) = 1$, $f(2) = 3$, $g(3) = 1$, and $g(4) = 3$. The pushout set P would be formed by taking the disjoint union $A \sqcup B = \{1, 2, 3, 4\}$ and identifying 1 with 3 as they are both images of elements under $f$ and $g$. So, $P = \{1 \sim 3, 2 \sim 4\}$, and the inclusion maps $i$ and $j$ would simply map 1 to 1 and 3 to 1, respectively.

## Pullback

Similarly, in a category, a pullback is a dual construction of a pushout. More explicitly, given two morphisms $f : A \rightarrow C$ and $g : B \rightarrow C$, their pullback is a triple $(P, \pi_1, \pi_2)$ where P is an object of the category and $\pi_1 : P \rightarrow A$, $\pi_2 : P \rightarrow B$ are morphisms such that $f \circ \pi_1 = g \circ \pi_2$. Furthermore, $(P, \pi_1, \pi_2)$ should be universal in a sense that for any other triple $(Q, \mu_1, \mu_2)$ with $\mu_1 : Q \rightarrow A$, $\mu_2 : Q \rightarrow B$, satisfying $f \circ \mu_1 = g \circ \mu_2$, then there exists a unique morphism $h : Q \rightarrow P$ such that $\mu_1 = \pi_1 \circ h$ and $\mu_2 = \pi_2 \circ h$.

A schematic description of the pullback is encased in Fig. 2.

In the category Set of sets and functions, the pullback of functions $f : A \rightarrow C$ and $g : B \rightarrow C$ can be constructed as follows:

The pullback object P is the set of pairs $(a, b) \in A \times B$ such that $f(a) = g(b)$. The morphisms $\pi_1 : P \rightarrow A$ and $p2 : P \rightarrow B$ are the projections onto the first and second coordinates, respectively.

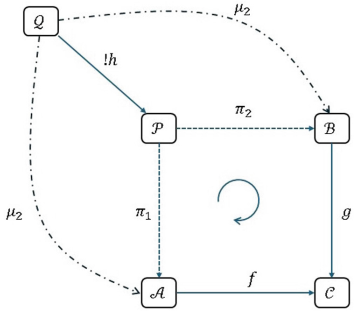


**Fig. 2.** A pullback diagram.

Formally,
$P = \{(a, b) \in A \times B | f(a) = g(b)\}$, $\pi_1 : P \rightarrow A, (a, b) \mapsto a$ and, $\pi_2 : P \rightarrow B, (a, b) \mapsto b$

## 3 Basic of Classification and Channel Theory

Barwise and Seligman [14] presented a framework for the "flow of information" in (generally implicitly) category-theoretic terms. They address the question, "How does information about some system component carry information about other components?"

They define a classification $\mathcal{A}$ to be a structure with non-empty sets typ($\mathcal{A}$) of types and tok($\mathcal{A}$) of tokens as well as a binary relation $\vDash_{\mathcal{A}}$ between tok($\mathcal{A}$) and typ($\mathcal{A}$) such that, for $a \in tok(A)$ and $\alpha \in typ(A)$, $a \vDash_{\mathcal{A}} \alpha$ indicates that a is of type $\alpha$. The theory does not limit what a or $\alpha$ might be (as long as it makes sense for a to be of type $\alpha$. It could be that a is an object and $\alpha$ a property (monadic first-order relation), or a might be a situation and $\alpha$ a type of situation; often, different tokens of a classification amount to the same physical system across different time points and types are instantaneous partial state descriptions of the system.

For classifications $\mathcal{A}$ and $\mathcal{C}$, an infomorphism f from $\mathcal{A}$ to $\mathcal{C}$ is a pair of functions

$$(f^{\wedge}, f^{\vee}), f^{\wedge} : typ(\mathcal{A}) \longrightarrow typ(\mathcal{C}) \; and \; f^{\vee} : tok(\mathcal{C}) \longrightarrow tok(\mathcal{A}) \tag{1}$$

satisfying, for all tokens $c \in tok(\mathcal{C})$ and all types $\alpha \in typ(\mathcal{A})$

$$f^{\vee}(c) \vDash_{\mathcal{A}} \alpha \;\; iff \;\; c \vDash_{\mathcal{C}} f^{\wedge}(\alpha) \tag{2}$$

Turning to regularities in a classification's types, let $\mathcal{A}$ be a classification and $\Gamma$ and $\Delta$ be sets of types in A. A token a of A satisfies the "sequent" $\langle \Gamma, \Delta \rangle$, provided that, if a is of every type in $\Gamma$, then it is of some type in $\Delta$. If every token of A satisfies $\langle \Gamma, \Delta \rangle$, then $\Gamma$ is said to entail $\Delta$ and $\langle \Gamma, \Delta \rangle$ is called a constraint supported by $\mathcal{A}$. The set of all constraints supported by $\mathcal{A}$ is called the complete theory of $\mathcal{A}$, denoted by $Th(\mathcal{A})$.

These constraints are system regularities, and information about some components of a distributed system carries information about other components because of regularities among connections. These regularities are relative to the analysis of the distributed system in terms of information channels. Barwise and Seligman's summary statement of their study of information flow, restricted to the simple case of a system with two components, a and b, is as follows.

### 3.1 Example of Theory of a Classification

Consider the classification $\mathcal{A}$ in Table 1 with tok($\mathcal{A}$)={a,b,c}, typ($\mathcal{A}$)=$\{\alpha,\beta,\delta\}$ and $\models_{\mathcal{A}} = \{((a,\alpha),(a,\delta),(b,\delta),(b,\beta),(c,\beta)\}$.

The theory of $\mathcal{A}$ [Barwise] is $Th(\mathcal{A}) = \{\langle\alpha,\delta\rangle, \langle\emptyset,\{\alpha,\beta\}\rangle, \langle\{\alpha,\beta\},\emptyset\rangle\}$.

| $\mathcal{A}$ | $\alpha$ | $\beta$ | $\delta$ |
|---|---|---|---|
| a | 1 | 0 | 1 |
| b | 0 | 1 | 1 |
| c | 0 | 1 | 0 |

**Table 1.** Example of classification

Given Table 1, the only token of type $\alpha$ is a; we can observe that a is also of type $\delta$; here, we remove the curly bracket around the singleton to make it readable. This explanation makes $\langle\alpha,\delta\rangle$ a sequent constraint supported by $\mathcal{A}$.

Given a classification A and a set $\Gamma$ of types of $\mathcal{A}$; let $\bigwedge\Gamma = \{x \in tok(\mathcal{A}) | \forall\alpha \in \Gamma, x \models_{\mathcal{A}} \alpha\}$ is a subset of tok($\mathcal{A}$); with $\bigwedge\emptyset =$ tok($\mathcal{A}$) and alternatively, $\bigvee\Gamma = \{x \in tok(\mathcal{A}) | \exists\alpha \in \Gamma, x \models_{\mathcal{A}} \alpha\}$. The following proposition gives an algebraic perspective on the validity of a constraint in a classification.

Proposition Given a classification $\mathcal{A}$, and a sequent $\langle\Gamma,\Delta\rangle$ in typ($\mathcal{A}$); $\langle\Gamma,\Delta\rangle$ is a constraint of $\mathcal{A}$ if and only if $\bigwedge\Gamma \subseteq \bigvee\Delta$.

Returning to the classification $\mathcal{A}$ in Table 1, $\bigwedge\emptyset = tok(\mathcal{A})$ and $\bigvee\{\alpha,\beta\} = tok(\mathcal{A})$ thus, $\langle\emptyset,\{\alpha,\beta\}$ is a constraint of $\mathcal{A}$. Similarly, $\bigwedge\{\alpha,\beta\} = \emptyset = \bigvee\emptyset$; validating $\langle\{\alpha,\beta\},\emptyset\rangle$ as a constraint of $\mathcal{A}$.

**Algorithm 1.** Theory Extraction from a Classification Table

Input: Classification table $\mathcal{A}$
$A_1 = typ(\mathcal{A})$; set of types of A
$A_2 = tok(\mathcal{A})$; set of tokens of A
$Th(\mathcal{A}) = P(A_1) \times P(A_1)$; set of all sequents
**for** $x \in A_2$ **do** Compute $\langle\Gamma,\Delta\rangle$ where $\Gamma = \{\alpha \in A_1 | \alpha \vdash x\}$ and $\Delta = \{\alpha \in A_1 | \alpha \nvDash x\}$
  **for** $\langle\Gamma_1,\Delta_1\rangle \in Th(\mathcal{A})$ **do**
    **if** $\langle\Gamma_1,\Delta_1\rangle \sqsubseteq \langle\Gamma,\Delta\rangle$ **then**
      $Th(\mathcal{A}) = Th(\mathcal{A}) - \{\langle\Gamma_1,\Delta_1\rangle\}$
    **end if**
  **end for**
**end for**
$Th(\mathcal{A})$=minimal of $Th(\mathcal{A})$
Output Th(A)

### 3.2 Channel Composition

One classification should be present in both channels to compose two channels. Without loss of generality, We will restrict our reasoning to binary channels and construct the composition for this case (Fig. 3).

Let $(f_1 : \mathcal{A} \rightarrow \mathcal{D}, f_2 : \mathcal{B} \rightarrow D)$ and $(g_1 : \mathcal{B} \rightarrow \mathcal{E}, g_2 : \mathcal{C} \rightarrow \mathcal{E})$ be two binary channels. We have in detail $f_1^{\wedge} : tok_{\mathcal{D}} \rightarrow tok_{\mathcal{A}}, f_1^{\vee} : typ_{\mathcal{A}} \rightarrow typ_{\mathcal{D}}, f_2^{\wedge} : tok_{\mathcal{D}} \rightarrow tok_{B}$ and,$f_2^{\vee} : typ_{\mathcal{B}} \rightarrow typ_{D}$ such that $\forall x \in tok_{\mathcal{D}}, \alpha \in typ_{\mathcal{A}}, \beta \in typ_{\mathcal{B}}\ f_1^{\wedge}(x) \models_{A} \alpha \iff x \models_{D} f_1^{\vee}(\alpha)$ and, Similarly, $g_1^{\wedge} : tok_{\mathcal{E}} \rightarrow tok_{B}, g_1^{\vee} : typ_{\mathcal{B}} \rightarrow typ_{\mathcal{E}}, g_2^{\vee} : tok_{\mathcal{E}} \rightarrow tok_{\mathcal{C}}$ and, $g_2^{\vee} : typ_{\mathcal{C}} \rightarrow typ_{\mathcal{E}}$ such that $\forall z \in tok_{\mathcal{E}}, \in typ_{\mathcal{B}}, \delta \in typ_{\mathcal{C}}$, $g_1^{\wedge}(z) \models_{B} \gamma \iff z \models_{\mathcal{B}} g_1^{\vee}(\gamma)\ g_2^{\wedge}(z) \models_{\mathcal{C}} \iff z \models_{\mathcal{C}} g_2^{\vee}(\delta)$

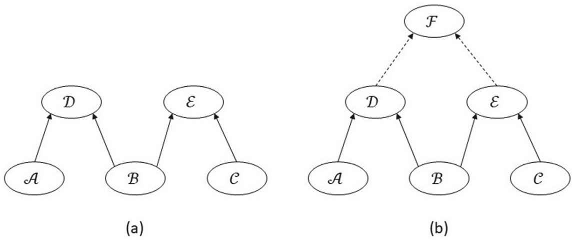


**Fig. 3.** Two Composable Channels.

By taking the pushout of $f_2^{\vee} : typ_{\mathcal{D}} \leftarrow typ_{\mathcal{B}} \rightarrow typ_{\mathcal{E}} : g_1^{\vee}$and the pullback of $f_2^{\wedge} : tok_{\mathcal{D}} \rightarrow tok_{\mathcal{B}} \leftarrow tok_{\mathcal{E}} : g_1^{\wedge}$, one obtained $i_1 : typ_{\mathcal{D}} \rightarrow typ_{\mathcal{D}} +_{typ_{\mathcal{B}}} \leftarrow typ_{\mathcal{E}} : i_2$ and $\pi_1 : tok_{\mathcal{D}} \leftarrow tok_{\mathcal{D}} \times_{tok_{\mathcal{B}}} tok_{\mathcal{E}} \rightarrow tok_{\mathcal{E}} : \pi_2$. Elements of $typ_{\mathcal{D}} +_{typ_{\mathcal{B}}} typ_{\mathcal{E}}$ are equivalent classes of $typ_{\mathcal{D}} \sqcup typ_{\mathcal{E}}$ the disjoint union; under the equivalence relation generated by $\{f_2^{\vee}(x) \sim g_1^{\vee}(x) | x \in typ_{\mathcal{B}}\}$; and elements of $tok_{\mathcal{D}} \times_{tok_{\mathcal{B}}} tok_{\mathcal{D}}$ set of couples $(d, e) \in tok_{\mathcal{D}} \times tok_{\mathcal{D}}$ such that $f_2^{\wedge}(d) = g_1^{\wedge}(e)$. We have defined classification elements, except we still need to include how those types classified the tokens Table 2.

### 3.3 Example of Channel Composition

Let $\mathcal{B}, \mathcal{D}, \mathcal{E}$ be three classifications.

**Table 2.** Base classifications for channel composition

| $\mathcal{B}$ | $\pi$ | $\rho$ | $\sigma$ |
|---|---|---|---|
| u | 0 | 1 | 1 |
| v | 0 | 0 | 1 |

| $\mathcal{D}$ | $\alpha$ | $\beta$ | $\gamma$ | $\delta$ |
|---|---|---|---|---|
| a | 0 | 1 | 1 | 1 |
| b | 0 | 0 | 1 | 1 |
| c | 0 | 0 | 0 | 1 |

| $\mathcal{E}$ | $\alpha$ | $\beta$ | $\theta$ | $\sigma$ | $\mu$ |
|---|---|---|---|---|---|
| a | 0 | 1 | 0 | 1 | 1 |
| x | 0 | 0 | 1 | 1 | 1 |
| y | 0 | 0 | 0 | 0 | 1 |

$tok(\mathcal{B}) = \{\pi, \rho, \sigma\}$; $typ(\mathcal{B}) = \{u, v, w\}$; $tok(\mathcal{D}) = \{\alpha, \beta, \delta\}$ $typ(\mathcal{D}) = \{a, b, c, d\}$; $tok(\mathcal{E}) = \{\alpha, \beta, \theta, \mu\}$; $typ(\mathcal{E} = \{a, x, y\})$
To define infomorphisms $f_2 : \mathcal{B} \to \mathcal{D}$ and $g_1 : \mathcal{B} \to \mathcal{E}$, we will used the following proposition.

**Proposition 1.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be two classifications and $f : tok(\mathcal{X}) \to tok(\mathcal{Y})$ an application. $f$ is the token part of an infomorphism if and only if $\forall \alpha \in typ(\mathcal{X}) \exists \beta \in typ(\mathcal{Y})$ such that $tok(\alpha) = f[tok(\beta)]$.*

*Proof.* $\Leftarrow$): With $f^\vee : tok(\mathcal{X}) \to tok(\mathcal{Y})$ as $f$, define $f^\wedge : typ(\mathcal{X}) \to typ(\mathcal{Y})$ by $\alpha \mapsto f^\wedge(\alpha) = \beta$
$\Rightarrow$) If $f$, as defined in the proposition, is the token part of an infomorphism, let prove that $tok(\alpha) = f^\vee[tok(f^\wedge(\alpha))]$
Let $u \in f^\vee[tok(f^\wedge(\alpha)] \Leftrightarrow \exists y \in tok(f^\wedge(\alpha))$ such that $u = f^\vee(y)$; $y \in tok(f^\wedge(\alpha)) \Leftrightarrow y \vDash_{\mathcal{Y}} f^\wedge(\alpha) \Leftrightarrow f^\vee(y) \vDash_{\mathcal{X}} \alpha \Leftrightarrow u \in tok(\alpha)$ since $u = f^\vee(y)$.

This proposition circumscribes those applications $f : tok(\mathcal{X}) \to tok(\mathcal{Y})$ that can serve as a token part of an infomorphism.

**Proposition 2.** *$f : typ(\mathcal{X}) \to typ(\mathcal{Y})$ is the type part of an infomorphism if and only if $\forall x \in tok(\mathcal{X}) \exists y \in tok(\mathcal{Y})$ such that $typ(x) = f[typ(y)]$*

Now let's define two infomorphisms $f : \mathcal{B} \to \mathcal{D}$ and $g : \mathcal{B} \to \mathcal{E}$; with $f^\vee : typ(\mathcal{B}) \to typ(\mathcal{D})$, $g^\vee : typ(\mathcal{B}) \to typ(\mathcal{E})$, $f^\wedge : tok(\mathcal{D}) \to tok(\mathcal{B})$, $g^\wedge : tok(\mathcal{E}) \to tok(\mathcal{B})$. $f^\wedge(a) = u, f^\wedge(b) = u, f^\wedge(c) = v$; $g^\wedge(a) = u, g^\wedge(x) = u, g^\wedge(y) = v$; $f^\vee(\pi) = \alpha, f^\vee(\rho) = \gamma, f^\vee(\sigma) = \delta$; $g^\vee(\pi) = \alpha, g^\vee(\rho) = \theta, g^\vee(\sigma) = \mu$.

We can use Proposition 1 to verify that f and g are indeed infomorphism. $tok(\pi) = \emptyset = f^\wedge(tok(\alpha))$; $tok(\rho) = u = f^\wedge(tok(\gamma))$ and $tok(\sigma) = u, v = f^\wedge(tok(\delta))$. Likewise, $tok(\pi) = \emptyset = g^\wedge(tok(\alpha))$;$tok(\rho) = u = g^\wedge(tok(\sigma))$ and $tok(\sigma) = u, v = g^\wedge(tok(\mu))$. As explained above, with these two infomorphisms, we can construct a classification table $\mathcal{F}$ where $typ(\mathcal{F}) = typ(\mathcal{D}) +_{typ(\mathcal{B})} typ(\mathcal{E}) = \{\{\alpha_1, \alpha_2\}, \{\beta_1\}, \{\beta_2\}, \{\gamma, \theta\}, \{\delta, \mu\}, \{\sigma\}\}$ and $tok(\mathcal{F}) = \{(r, s) \in tok(\mathcal{D}) \times tok(\mathcal{E}) | f^\wedge(r) = g^\wedge(s)\} = \{\langle a, a\rangle, \langle a, x\rangle, \langle b, x\rangle, \langle c, y\rangle\}$

---

**Algorithm 2.** Infomorphism Checking

---

1: Input: Classification tables A, B and applications $f^\vee : typ(A) \to typ(B)$ $f^\wedge : tok(B) \to tok(A)$
2: Define a function *tok_set* that returns the set of tokens of a given classification
3: Define a function *typ_set* that returns the set of types of a given classification
4: Define a function $token_set$ that for a given type and classification the set of tokens classified by the type
5: Define a function *image_set* which, for a given function and a subset of its domain, will return the image of the subset
6: Evaluate the following formula to be True: $\forall \mu \, in \, typ\_set(ClaA) \exists \tau \, in \, typ\_set(ClaA) s.t. token\_set(\mu) = image\_set(token\_set(\tau))$

---

**Table 3.** Example of sum of classifications

| $\mathcal{F}$ | $\{\{\alpha_1, \alpha_2\}$ | $\{\beta_1\}$ | $\{\beta_2\}$ | $\{\gamma, \theta\}$ | $\{\delta, \mu\}$ | $\{\sigma\}$ |
|---|---|---|---|---|---|---|
| $\langle a, a\rangle$ | 0 | 1 | 1 | 1 | 1 | 0 |
| $\langle a, x\rangle$ | 0 | 1 | 1 | 1 | 1 | 1 |
| $\langle b, x\rangle$ | 0 | 0 | 0 | 1 | 1 | 1 |
| $\langle c, y\rangle$ | 0 | 0 | 0 | 0 | 1 | 0 |

## 4 Collaboration Between Intelligent Security Agents

### 4.1 Resource Sharing

Resource sharing increases general efficiency, as each agent can access and use the resources it does not have Table 3. As such, it increases the rate at which problems are solved, accelerates productivity, and improves resource utilization. Regarding resource sharing, studies have shown that systems reduce infrastructural duplication and redundancies of resources, enabling these systems to save costs. It is beneficial in an environment where resources might be too scarce or expensive for the consumer [14].

On the other hand, resource sharing builds dependencies among the agents. If many agents depend on some key resources, failure to deliver promptly depletes the whole system and creates instability. Data breach incidents through unauthorized access, disclosure of data, and information leakages remain serious issues within resource-sharing frameworks. The threat rate increases in this context if sensitive information is shared between the concerned parties.

Efficient resource allocation among agents presents many challenges, especially in dynamic and heterogeneous environments. Conflicting requirements for resource use may generate a competitive, efficient, or fair distribution of resources. Besides, limitations in the network could result in stricture bandwidth or high latency, introducing performance bottlenecks and contributing to delay and degraded system performance.

Resource sharing among large numbers of agents involves sophisticated mechanisms for allocation, access control, and communication protocols. Hence, the coordination of sharing becomes complex. For these reasons, the design and implementation of a system are more complicated. While beneficial in many aspects of collaboration, scalability, and efficiency, resource sharing introduces issues related to dependencies, privacy concerns, performance, and complexity. Only by overcoming these challenges can effective management and governance of these systems fully unleash the potential of collaborative intelligent systems. In this respect, Miller and Taylor [15] have stated, “There is mounting evidence to suggest that artificial intelligence is not a risk-multiplier, per se, but more an amplifier and accelerator of existing risks.”

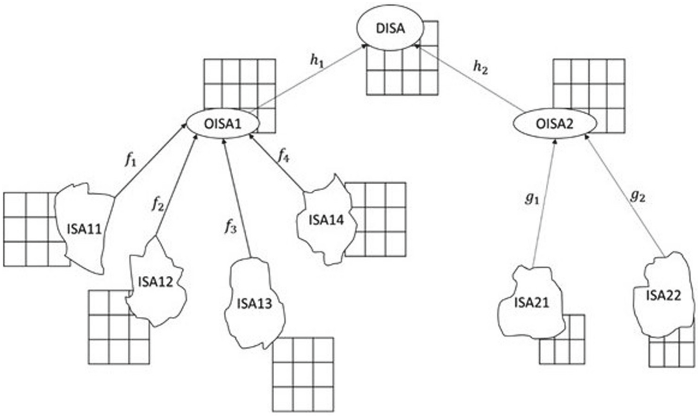


**Fig. 4.** Example picturing a network of intelligent security agents.

### 4.2 Agent Interactions: Game Theory Model for Resource Sharing

Each agent carries a classification table, as shown in Fig. 4 above, which, by Algorithm 1, extracts a set of sequents that constrain and govern its behavior. The game involves several steps. We use linear logic [?] as the underlying structure to ensure the correct handling of resources and game theory concepts to model agents' strategic interactions when accessing and sharing resources.

Here's how we approach this problem: Framework There are N agents $A_1, A_2, \ldots, A_N$, and each agent is constrained by a set of logical sequents, representing their access rights, strategies, and constraints on resource usage. There are M resources $R_1, R_2, \ldots, R_M$ that agents compete for. These resources can be limited and may be consumed once or shared under certain conditions. Each agent has a set of strategies $S_i$, which determines how they will attempt to access and use the resources constrained by their logical rules. Each agent aims to maximize their payoff, which is a function of the resources they obtain and the costs associated with acquiring them.

**Components of the Game.** 1. Sequents for Each Agent Each agent has a set of logical sequents that govern how they interact with the resources. A sequent describes: Access rights (The resources the agent is allowed to use). Consumption rules (Whether the agent must consume a resource entirely or can share it with others). Cost/utility: How much utility does the agent derive from obtaining a resource, or how much is the associated cost? Let's define the sequent constraints for an agent $A_i$: $\Gamma_i \vdash \Delta_i$: where $\Gamma_i$ represents the agent's conditions (available resources and current state), and $\Delta_i$ represents the agent's potential actions or outcomes, such as accessing a resource or sharing it with another agent.

2. Agents' Strategic Interaction (Game Setup) Agents engage in a game where their strategies are defined by their possible sequents, aiming to maximize their payoffs. The game proceeds in rounds, and during each round: Each agent selects

a strategy from their set $S_i$, constrained by their sequents. Agents interact with each other by competing for or sharing resources. Resources are allocated based on the outcome of the interactions, which depend on the priority rules, sharing protocols, or auctions.

3. Game Structure: Players: The players are the agents $A_1, A_2, \ldots, A_N$. Actions: Each agent chooses a strategy that specifies which resource(s) they want to access or share within the constraints of their sequent system. Payoff Function: Each agent's payoff function $U_i$ depends on the resources they obtain and the costs they incur. If an agent obtains resource $R_j$, their payoff might increase, but attempting to access the resource incurs costs, especially if there's competition.

**Example Game Design.** Consider a game with three agents $A_1, A_2, A_3$ and two resources $R_1$, $R_2$. The agents are constrained by the following sequents: Agent $A_1$: Can access either $R_1$ or $R_2$, but not both simultaneously. $\Gamma_1 \vdash (R_1 \otimes R_2)$ This means $A_1$ can choose one resource, and once they consume it, the other becomes unavailable. Agent $A_2$: Can only access $R_1$, but can share it with others if they agree to a cost-sharing arrangement. $\Gamma_2 \vdash (R_1 \oplus shared(A_1))$ $A_2$ can either fully consume $R_1$ or share it with other agents at a reduced payoff. Agent $A_3$: Has access to $R_2$ and can consume it fully but must pay a high cost if they compete with another agent for the resource. $\Gamma_3 \vdash R_2$ high cost if contended $A_3$ faces a high cost if they must compete with another agent for $R_2$.

Sequence of Play Round 1 (Strategy Selection): Each agent selects a strategy based on their sequents and resources available. $A_1$ can choose between $R_1$ and $R_2$. $A_2$ chooses to either fully consume $R_1$ or offer to share it with others (e.g., at a reduced cost). $A_3$ decides whether to attempt to access $R_2$, knowing that they will face a high cost if they need to compete. Round 2 (Resource Allocation): Resources are allocated based on agents' strategies. Cost-based auctions will decide allocation if multiple agents compete for the same resource. For example, if both $A_1$ and $A_3$ choose $R_2$, they enter into a bidding, with the losing agent either paying a cost. Round 3 (Payoff Calculation): Each agent's payoff is calculated based on the resources they successfully acquire and the costs they incur. For instance: $A_1$'s payoff depends on whether they secured $R_1$ or $R_2$. $A_2$ gains a higher payoff if they fully consume $R_1$, but a lower payoff if they had to share it. $A_3$ incurs a high cost if they had to compete for $R_2$. Repeat (Multiple Rounds): The game continues over multiple rounds, with agents adjusting their strategies based on previous outcomes. Over time, agents develop optimal strategies (such as agreeing to share resources to avoid high competition costs).

Payoff Structure The payoff for each agent $A_i$ will be represented as: $U_i(S_i, R_j) =$ utility from resources acquired cost of competition or sharing Where: $S_i$ is the strategy selected by agent $A_i$. $R_j$ is the set of resources $A_i$ acquires. The utility increases when more valuable resources are obtained. The cost increases if there is competition or if the agent had to share the resource.

Nash Equilibrium and Resource Sharing The Nash equilibrium of the game is reached when no agent can unilaterally change their strategy and improve their payoff. In this context, we have two possibilities:

Cooperative Equilibria in which Agents might agree to share resources to avoid competition costs, leading to a cooperative solution where resources are divided equitably and,

Non-cooperative equilibrium is where agents act selfishly, and the equilibrium might involve higher competition and costs, where agents try to maximize their share of resources without regard for others.

Incorporating Sequent Calculus for Constraints Each agent's strategy set $S_i$ is constrained by the sequents governing their behavior. For instance, if an agent has a sequent that only allows access to specific resources, they cannot violate this rule. The game's logic ensures that resources are not over-consumed, and agents must adhere to the regulations imposed by their sequent systems.

## 5 Conclusion and Further Research

This paper develops a general framework grounded in classification, infomorphism, and channel theory for principled information-sharing amongst multiple agents. We develop a channel-composition construction using specific limit and colimit constructions, namely categorical notions of pullback and pushout. We apply this to model how agents can interact with one another depending on their resources and knowledge.

The heart of the composable informorphisms does capture the mutual information in an integral that considers more than one isolated agent would consider; indeed, the audience would be 'informed'. This setting emphasizes the compositional aspects of information flow between agents and illustrates the flexibility of category theory when modeling such multi-agent and complex communication systems.

This binary model is helpful but oversimplifies the rich complexity of real-world applications often present, in which tasks usually require varied degrees or resources.

Therefore, our model needs to be extended from this simple binary case to a more general multi-relation framework to accommodate better the spectrum of resource-task relationships rather than a binary true-or-false condition.

This would allow us to model more realistic dynamics in which resources might be available to a greater or lesser degree, partial or graded resource availability. These will generally have huge impacts on the success or efficiency of completions. Further study in the multi-relation model might lead to understanding resource management in a multi-agent system, which may change how we do resource management. An architecture like this might also suggest ways to embed probabilistic models or introduce more fuzzy relationships between tasks and resources within a flexible and scalable framework for managing information sharing and coordination across diverse systems.

## References

1. Suryanegara, M., Halim, R., Anwar, K.: Cooperative communication for resource sharing. Proc. Comput. Sci. **98**, 469–476 (2016)
2. Jennings, N.R., Bussmann, S.: Agent-based control systems. IEEE Control Syst. Mag. **23**(3), 61–74 (2003)
3. Mancarella, P., Strbac, G.: Modeling of economics and reliability for multi-agent systems in power networks. IEEE Trans. Power Syst. **14**(1), 135–140 (1999)
4. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: IEEE Workshop on Mobile Computing Systems and Applications, pp. 85–90 (2003)
5. Bonomi, F., Milito, R., Natarajan, P., Zhu, J.: Fog computing: A platform for the Internet of Things and Analytics, pp. 169–186. A Roadmap for Smart Environments, Big Data and Internet of Things (2012)
6. Barwise J., Seligman J.: Information flow: The logic of Distributed Systems. Cambridge Tracts in Theoretical Computer Science (44), Cambridge, UK (1997)
7. Shannon, C.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423 (1948)
8. MacLane, S.: Categories for the Working Mathematician, 2nd edn. Springer-Verlag, New York (1998)
9. Simmons, H.: An Introduction to Category Theory. Cambridge University Press, Cambridge, UK (2011)
10. Awodey, S.: Category Theory, 2nd edn. Oxford University, Press, Oxford, UK (2010)
11. Lawvere, F.W., Schanuel, S.H.: Conceptual Mathematics. A First Introduction to Categories (2nd Ed.). Cambridge University Press: Cambridge (2009)
12. Pierce, B.C.: Basic Category Theory for Computer Scientists. MIT Press, Cambridge, MA (1991)
13. Barr, M., Wells., C.: Category Theory for Computing Science. Prentice Hall: Upper Saddle River, NJ (1990)
14. Fiadeiro, J.L.: Categories for Software Engineering. Springer, Berlin (2005)
15. Liao, Z., Zhang, Y.: Enhancing collaboration through dynamic resource sharing among intelligent systems. Syst. Control Lett. **45**(2), 67–78 (2019)
16. Miller, D., Taylor, K.: Coordination complexities in shared-resource environments. AI Rev. **35**(1), 215–230 (2020)
17. Girard, J.-Y.: Linear logic. Theoret. Comput. Sci. **50**, 1–102 (1987)