Supplementary Materials
The Actor–Gate Dependency Notation (AGDN): A Visual Boundary Object for Structuring Multi-Actor Intellectual Property Decisions and Negotiations

[Author name removed for double-blind review]

These Supplementary Materials accompany the main article. Section S1 gives a one-page quick reference to the notation; Section S2 develops the patent tokenization example in full; Section S3 gives the BPMN counterfactual and the response to the notation-extension objection; Section S4 sets out the decision-structure foundations and the worked transformations to formal DSS models; Section S5 reports the formative expert walkthrough in full; Section S6 gives the controlled experiment's apparatus (participants, power, randomisation, materials, and scoring); and Section S7 works through the SPC portfolio misspecification example. Cross-references of the form "the main paper" point to the article these materials accompany. Numbering of sections, tables, and figures is prefixed with S; the participant-level data files are referenced as D1–D3 in the main paper's Data Availability statement.

Contents

S1 Notation Quick Reference

For convenience, the notation comprises seven core element types and a small set of annotations. The **sphere** (◯) denotes an institutional actor, in four variants: University (open circle), Industry (hatched), Government (double outline), and Hybrid (half-patterned, with a mandatory annotation such as (U/I) or (Inst/Alg)). The **asset** (□) denotes a governance-relevant artefact, with subscripts for type ($\square_{\text{pat}}$, $\square_{\text{app}}$, $\square_{\text{doc}}$, $\square_{\text{disc}}$, and others). The **approval** (hexagon) denotes a regulatory authorisation; the **value** (◊) an economic interest derived from an asset; the **gate** (black bar) a decision point annotated (Inst), (Alg), or (Hyb); and the **time** marker ($|\tau|$) a temporal constraint. The **flow** element has three connector variants—transfer (solid arrow), control (gray arrow), derivation (dotted, no arrowhead)—and an orthogonal status channel: solid (active), dashed (pending), bar overlay (blocked, with a mandatory qualifier). A diagram is well-formed when every value element grounds, through its derivation chain, in exactly one asset; every cross-sphere transfer passes through at least one gate; control connectors may cross spheres freely; status is orthogonal to connector type; and every hybrid sphere carries its constituent-type annotation.

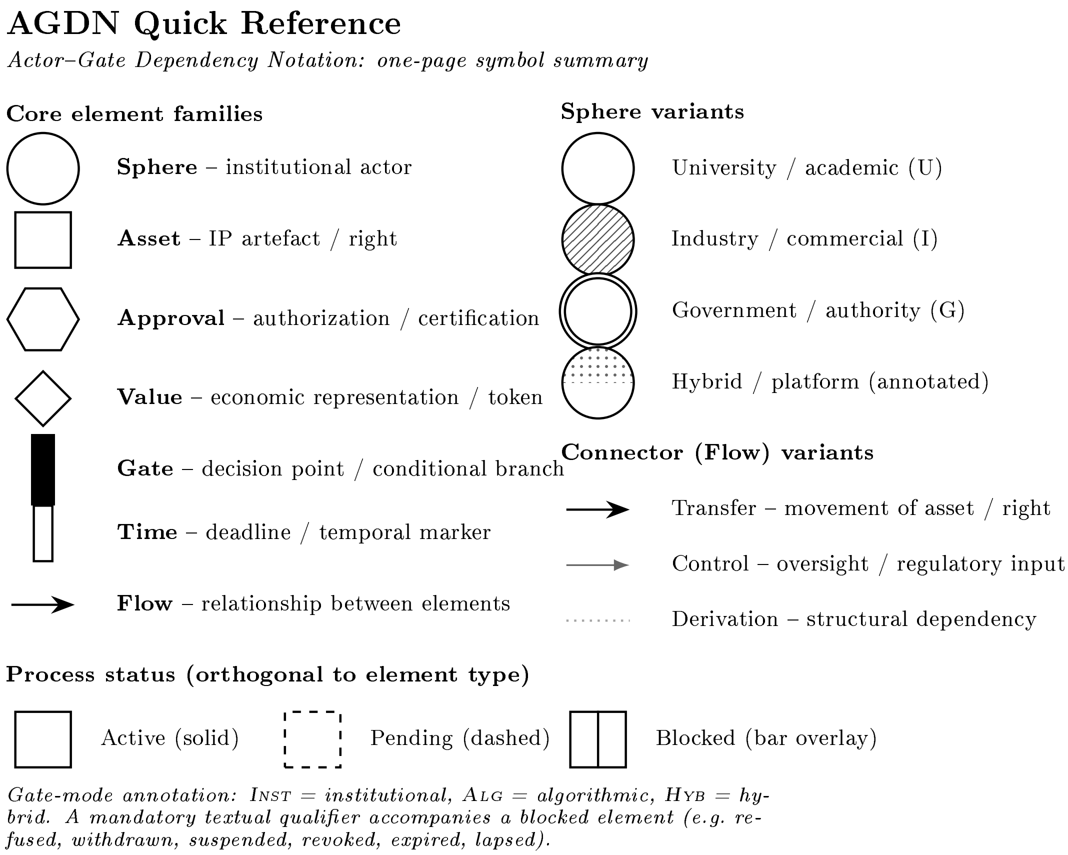


Figure S1: AGDN quick reference: the seven core element families, the four sphere variants, the connector variants, and the status encoding on a single page.

S2 Patent Tokenization on Distributed Ledger: Extended Analysis

Extended analysis of patent tokenization follows, explicitly addressing decision tensions and illustrating AGDN's capacity to represent hybrid decision arrangements. Tokenization is treated as a stress-test scenario rather than a core domain claim; the methodological contribution lies in demonstrating that AGDN can represent institutional-algorithmic coupling, not in advocating for tokenization. While still in early stages, patent tokenization has moved beyond purely theoretical speculation: in 2021, IBM and IPwe announced plans to represent corporate patents as NFTs on blockchain infrastructure, aiming to create more liquid and accessible patent markets [IBM Corporation and IPwe, 2021]. Whether such initiatives will achieve their stated goals remains uncertain, but their existence confirms that the decision configurations AGDN depicts are not merely hypothetical.

S2.1 Basic Tokenization Structure

Figure S2 presents a complete representation.

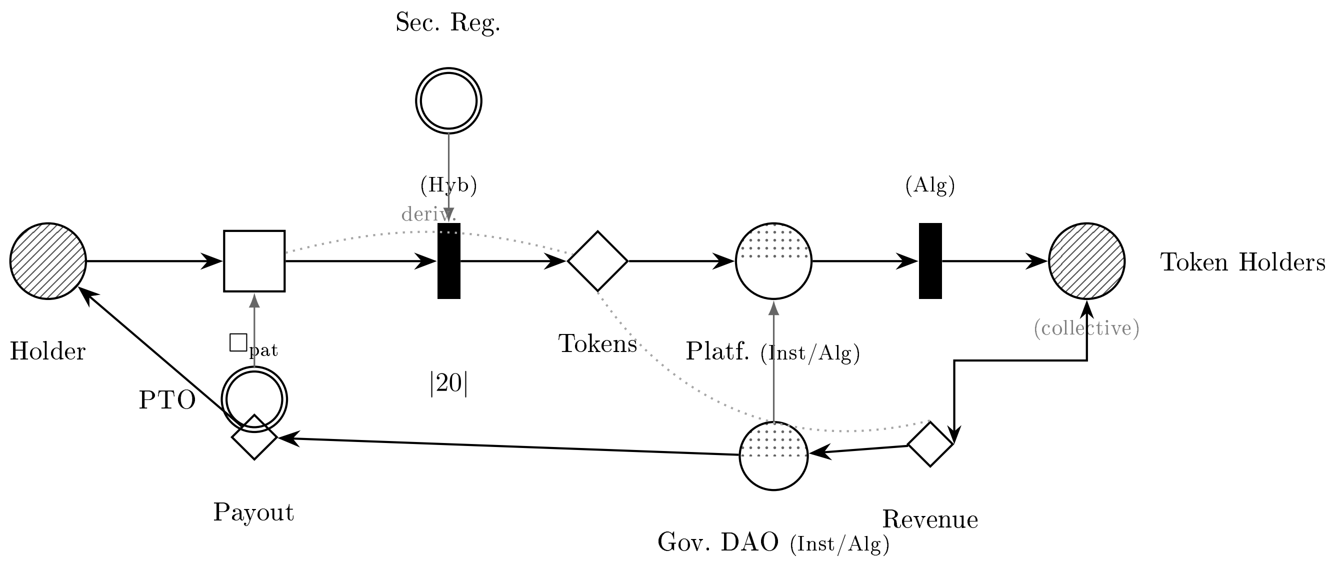


Figure S2: AGDN diagram: Patent tokenization on distributed ledger. Patent ($\square_{\text{pat}}$) held by industry proceeds through hybrid gate (Hyb) with securities regulatory oversight to create tokens. Platform (Inst/Alg hybrid) distributes through algorithmic gate (Alg) to token holders (modeled as collective market actor). Governance DAO manages platform; revenue flows through explicit payout value element back to patent holder, ensuring no direct sphere-to-sphere value transfer. Dotted curved line (no arrowhead) shows tokens derive from underlying patent, representing a structural dependency, not a process flow. Patent term |20| constrains value horizon.

Key features address decision complexity:

Multiple government spheres. Patent office and securities regulator exercise authority over different aspects. The PTO governs patent validity; the securities regulator governs token offering compliance. Compliance with one does not ensure compliance with the other.

Operational compliance layer. Operational compliance requirements (AML/KYC screening, sanctions compliance, custody/safekeeping obligations, and marketing communication rules) are modeled as additional institutional gates and control connectors attached to the platform and distribution stages. These constraints often persist beyond initial issuance, reflecting that governance requirements extend throughout the token lifecycle. In the diagram, such gates would appear between platform and distribution, with control flows from relevant financial supervisory authorities.

Differentiated gates. The tokenization gate carries (Hyb) for hybrid governance: securities compliance involves both institutional review and algorithmic implementation.

The distribution gate carries (Alg) for algorithmic governance: once deployed, distribution executes through smart contract logic.

Token Holders as collective actor. Rather than depicting individual token holders ($TH_1 ... TH_n$), which would violate the design principle that AGDN models institutional actors, token holders appear as a single collective sphere. This represents them as market participants whose aggregate behavior is institutionally relevant, preserving governance accountability tracing while avoiding individual-level representation outside AGDN's scope.

Platform and DAO as Inst/Alg hybrids. Both platform and governance DAO carry (Inst/Alg) annotation, indicating that they combine algorithmic coordination with institutional structures (legal wrappers, custody arrangements, compliance functions). The mandatory annotation ensures interpretability.

Derivation annotation. The dotted curved gray line connecting patent to tokens, distinct from process flows and rendered without arrowhead, emphasizes that tokens derive from the underlying asset. This is a structural dependency annotation, not a process flow: tokens are economic claims on value derived from the patent, which persists independently. Tokens represent participation rights or revenue claims; they do not constitute fractional ownership of the patent right itself.

S2.2 Governance Tensions Made Visible

The diagram enables identification of specific tensions:

Legitimacy conflict: If the DAO votes to enforce the patent but a court determines no infringement, the legitimacy claims conflict. The connection between patent and tokens (derivation line) and the flows involving enforcement (not shown but implied) identify where this tension arises.

Rights unbundling: The diagram shows tokens deriving from the patent, but what exactly do they convey? The derivation annotation indicates structural dependency; accompanying documentation must clarify whether tokens represent economic claims only, governance participation rights, or other specified interests. AGDN explicitly distinguishes assets (the patent) from values (the tokens): the former has legal existence as intellectual property, the latter are economic representations.

Regulatory arbitrage: The securities regulator sphere could represent different regulators (SEC, ESMA, MAS) with different requirements. The diagram structure remains the same; annotation specifies jurisdiction.

Accountability: Revenue flows through the DAO, but who is accountable if funds are mismanaged? The DAO's hybrid status (Inst/Alg) indicates mixed governance resembling polycentric governance arrangements [Ostrom, 2010], yet often lacking the graduated sanctions emphasized in commons governance design [Ostrom, 1990] as essential for stability.

The polycentric reading also reframes the negotiation question the diagram makes visible. A token-holder collective is a decision center in Ostrom's sense, exercising distributed control that must nonetheless negotiate with the concentrated institutional authority of a patent office or court whose determinations it cannot override. AGDN represents both poles explicitly—the collective sphere with its algorithmic gate, the institutional sphere with its (Inst) gate—and the control connectors between them mark exactly where the distributed and the institutional centers must reach accommodation. The notation does not resolve that negotiation, but it locates it structurally, which is the precondition for analyzing it.

S2.3 Token Regulatory Classification

Regulatory classification determines which compliance regime applies to tokenized IP interests. Figure S3 illustrates the classification branch.

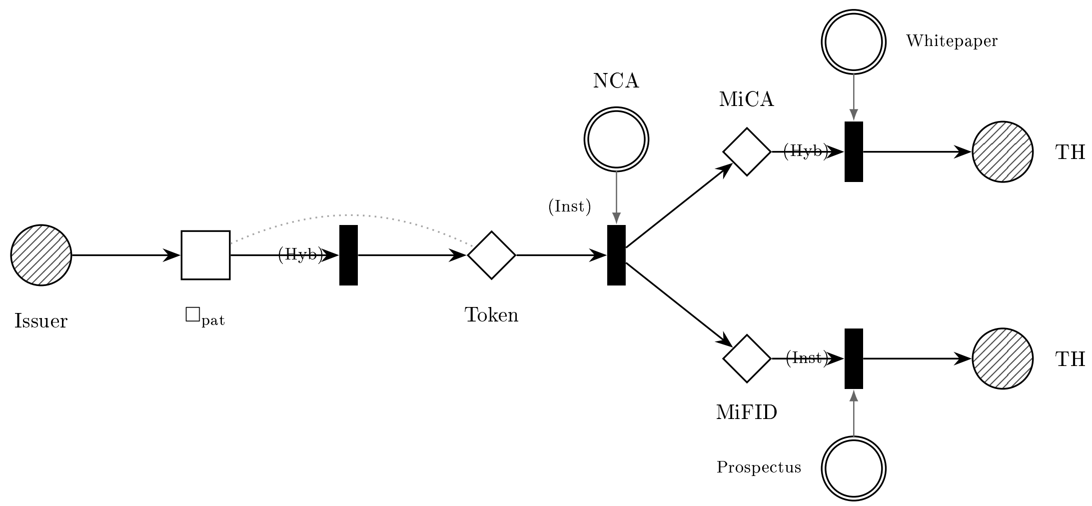


Figure S3: AGDN diagram: Token regulatory classification branch. Token issuance proceeds through hybrid gate (Hyb) to classification gate where national competent authority (NCA) determines applicable regime. MiCA path (crypto-asset regime) requires whitepaper compliance through hybrid gate. MiFID path (financial instrument regime) requires prospectus through institutional gate (Inst). Different downstream compliance obligations depending on classification outcome. This branching structure (same token, different decision pathways based on regulatory determination) represents a core challenge for IP tokenization.

The classification gate creates a fundamental branch: tokens classified under MiCA face different disclosure requirements (whitepaper), trading venue rules, and supervisory frameworks than tokens classified as financial instruments under MiFID II (prospectus, authorized venues, market abuse regulation). AGDN represents this as a single gate with branching outcomes, each leading to distinct downstream decision configurations. What the diagram makes visible is that regulatory classification is not merely a compliance checkbox but a decision-shaping decision affecting the entire token lifecycle.

S3 BPMN Counterfactual and the Notation-Extension Objection

S3.1 Counterfactual Analysis: BPMN Representation of Patent Tokenization

To illustrate AGDN's analytical advantages concretely, consider how the patent tokenization scenario of Section S2 would appear in BPMN notation. A BPMN representation would require: (1) multiple pools representing patent holder, platform operator, token holders, and regulatory authorities, with swimlanes for internal departments; (2) message flows crossing pool boundaries for every interaction, resulting in a dense network of crossing arrows; (3) gateway symbols (XOR, AND) for decision points, without distinguishing institutional from algorithmic execution; (4) no native representation for the hybrid status of platforms or DAOs—BPMN pools assume organizationally coherent actors; (5) no status encoding distinguishing pending from granted assets—BPMN represents process flow, not artifact state; (6) no derivation concept linking tokens to underlying patents—the structural dependency would require textual annotation or a separate data object with association lines.

The resulting BPMN diagram would contain approximately 25–30 elements (vs. AGDN's 12) with significantly higher visual complexity due to message flow crossings. More critically, BPMN would obscure the governance-relevant distinctions that AGDN foregrounds: the institutional-algorithmic boundary at the distribution gate, the hybrid character of platform governance, and the derivation relationship grounding token value in patent rights. These distinctions matter for decision support because they identify where governance mode affects decision dynamics—information that generic workflow notation does not capture.

This counterfactual analysis does not claim empirical superiority (which would require controlled comparison studies) but demonstrates analytically that AGDN's design choices address governance representation needs that general-purpose notations structurally cannot accommodate without extensive adaptation.

Why Not Extend Existing Notations? A natural question is whether AGDN's features could be added to BPMN, UML, or influence diagrams through extensions rather than introducing a new notation. Three factors motivated a standalone design. First, **semantic incompatibility**: BPMN's process-flow semantics conflict with AGDN's decision-accountability focus—BPMN swim lanes represent organizational units, not institutional actors with distinct governance logics; extending BPMN to support Triple Helix sphere semantics would require fundamental metamodel revision. Second, **cognitive load**: Adding AGDN's status encoding and sphere variants to BPMN would further increase its already problematic symbol count (68 event types, 8 gateway types), exacerbating the cognitive effectiveness deficits that Moody [2009] documented. Third, **target audience**: AGDN addresses problem structuring for multi-actor governance, a different use case than BPMN's workflow automation or UML's software specification; notation design should match use case rather than forcing one tool to serve incompatible purposes. Rather than extending notations optimized for other functions, AGDN occupies a distinct position in the DSS toolkit—upstream of workflow design, orthogonal to software architecture, and complementary to probabilistic inference tools.

The BPMN+DMN combination. A more specific version of the objection points to the Decision Model and Notation (DMN) standard, often paired with BPMN: DMN decision tables and Decision Requirement Graphs (DRGs) capture decision logic and the dependencies among decisions, which seems close to what AGDN represents. The resemblance is partial. A DRG models how one decision depends on sub-decisions and input data within a single decision-making entity, and a DMN decision table specifies the logic that turns inputs into an outcome. Neither encodes *who* holds the decision right, whether the decision mode is institutional or algorithmic, or the legal status of the artifact on which the decision operates—precisely the dimensions AGDN foregrounds. DMN answers "given these inputs, what is the decision?"; AGDN answers "which actors, under which authority and constraints, govern this decision, and on which assets does it bear?" The two are therefore complementary rather than competing: an AGDN diagram identifies the decision points and the actors that own them, and a DMN table could subsequently specify the internal logic of any single institutional gate. We treat DMN as a candidate downstream target alongside decision trees and influence diagrams, not as an alternative to the structuring layer AGDN provides.

S4 Decision Structures and Transformation to Formal DSS Models

Decision science has long distinguished between decision optimization and decision structure. While normative decision theory focuses on preferences, utilities, and optimal choice rules, the behavioral and organizational decision literature emphasizes that outcomes are critically shaped by how decision situations are structured, decomposed, and represented.

Research on bounded rationality argues that organizations do not solve complex decision problems through continuous optimization, but through heuristics, rules, and structured representations that simplify the decision environment [Simon, 1955, March and Simon, 1958]. Subsequent work demonstrates that the structure and presentation of decision information systematically affect attention, error patterns, and judgment, even when underlying preferences remain unchanged [Payne et al., 1993, Gigerenzer and Selten, 2001].

Visual representation plays a functional role in this tradition. Diagrams can reduce cognitive load, support relational inference, and render structural dependencies explicit in ways that purely textual descriptions cannot [Larkin and Simon, 1987]. The effectiveness of such representations depends on disciplined mapping between graphical elements and semantic constructs rather than on visual richness [Moody, 2009].

AGDN is positioned within this strand of decision science as a descriptive representation tool for decision structures surrounding intellectual property. It does not model preferences, utilities, or choice rules, nor does it prescribe decisions. Instead, it renders observable constellations of actors, artefacts, approvals, temporal constraints, and dependencies explicit, thereby enabling comparative and critical analysis of decision environments across institutional contexts.

In this sense, AGDN complements decision-analytic and econometric models rather than competing with them: decision models address how choices should be evaluated given a structure, while AGDN addresses how such structures are constituted and coupled.

S4.1 Implications for Decision Support Systems

Decision support systems research distinguishes between problem structuring, model formulation, and solution analysis [Keen and Scott Morton, 1978, Sprague and Carlson, 1982], and critical reviews of the field have repeatedly noted that the structuring phase receives less methodological attention than solution techniques [Arnott and Pervan, 2005, Alter, 2004]. Prior work consistently identifies inadequate structuring as a primary source of downstream modeling error, independent of the sophistication of analytical techniques [Simon, 1960, Pomerol and Adam, 2004]. AGDN is positioned explicitly within the problem structuring phase of DSS, providing a visual representation of institutional, regulatory, and asset-related decision dependencies before formal modeling.

Within a DSS lifecycle, AGDN diagrams function as an intermediate representation between narrative case descriptions and formal decision models such as decision trees, influence diagrams, or optimization formulations. The notation clarifies who decides what, under which constraints, and in which sequence, thereby reducing model misspecification and supporting communication among stakeholders involved in DSS design and use.

AGDN is not a decision support system itself. It complements DSS by improving the quality and completeness of decision problem formulation, which remains a recognized bottleneck in complex, multi-actor decision environments [Power, 2002].

S4.2 Integration into DSS workflows

In a DSS development workflow, AGDN diagrams can be used as a front-end representation that precedes formal modeling. The diagram clarifies decision ownership, institutional

Table S1: Positioning AGDN within the DSS toolkit.

Tool	DSS phase	Primary function	Structural limitation addressed by AGDN
Decision trees	Evaluation	Sequential choice under uncertainty	Weak representation of multi-actor institutional gates
Influence diagrams	Modeling	Dependency representation for probabilistic models	Limited representation of accountability and institutional decision modes
BPMN / process maps	Process design	Workflow sequencing	Often conflates process flow with decision accountability and status
Optimization models	Solution	Compute optimal decisions given a model	Sensitive to structural misspecification in problem formulation
AGDN	**Problem structuring**	**Decision architecture visualization**	**Makes actors, gates, and dependencies explicit prior to analytics**

gates, and dependency structure; this output can then be translated into decision trees, influence diagrams, or optimization formulations as appropriate. AGDN does not claim to improve solution quality directly; rather, it supports DSS by improving the completeness and transparency of the structuring inputs on which downstream analytics depend.

S4.3 Transformation to DSS Models

Table S2 provides indicative mappings from AGDN elements to common DSS model constructs, including formal definition examples demonstrating mathematical operationalization. These mappings are not automatic transformations but guidelines for analysts translating structured problem representations into formal models.

For example, a simple SPC acquisition scenario (the SPC figure in the main paper) translates to a decision tree where each gate becomes a decision or chance node, patent and marketing authorization become state variables conditioning subsequent decisions, and the regulatory delay parameter informs branch probabilities or temporal discounting. The AGDN diagram does not specify these quantitative parameters—that requires domain expertise and empirical data—but it identifies which structural elements require parameterization.

Table S2: Indicative AGDN to DSS model mappings, with formalization examples.

Element	Decision Tree	Influence Diag.	Optimization	Formal Definition
Gate (Inst/Alg)	Decision node	Decision node	Binary var.	$x_{\text{gate}} \in \{0, 1\}$
Asset (□)	State variable	Chance node	State var.	$s_t \in \{\text{grant}, \text{pend}, \text{void}\}$
Approval (hex)	Constraint	Determ. node	Feasibility	$g(\mathbf{x}) \leq 0$
Transfer	Sequence edge	Info. arc	Flow var.	$f_{ij} > 0$ if $x_{\text{gate}} = 1$
Control	(implicit)	Info. arc	Constr. link	$\mathbf{Ax} \leq \mathbf{b}$
Time ($\lvert\tau\rvert$)	Horizon	Temp. discount	Time constr.	$t_{\text{end}} - t_{\text{start}} \leq \tau_{\max}$
Derivation	Value dep.	Value arc	Obj. coeff.	$V_{\text{tok}} = f(V_{\text{asset}}, \delta)$
Status (pend.)	Prob. branch	Uncertainty	Scenario par.	$p(\text{grant}) \sim \text{Beta}(\alpha, \beta)$

S4.4 From AGDN to Multi-Agent Control Models

Although AGDN is deliberately descriptive, its well-formed diagrams can be compiled into standard multi-agent decision models. This provides an explicit bridge from problem structuring to formal multi-agent control, without changing AGDN's non-normative scope: AGDN specifies the *architecture* of who can act, where decisions occur, which constraints apply, and which value representations are structurally dependent on underlying assets. *Terminological note:* AGDN uses the term *actor* to denote institutional bearers of responsibility without behavioral assumptions; the term *agent* is reserved for downstream formalizations (multi-agent control, game-theoretic models) derived from AGDN diagrams.

Typed decision-structure graph. A AGDN diagram induces a typed directed multigraph $D = (V, E, \tau, \sigma, \kappa, \mu)$ where V is the set of nodes partitioned into spheres (actors), assets, approvals, values, gates, and time markers; E is the set of typed edges partitioned into transfer, control, and derivation relations; τ assigns temporal constraints to edges or gates; σ assigns status labels (active/pending/blocked) to stateful nodes and process edges; κ assigns sphere types (U/I/G/Hybrid) to actor nodes; and μ assigns gate modes (Inst/Alg/Hyb) to gate nodes.

Agents, actions, and constraints. Each sphere node corresponds to an agent $i \in \{1, \ldots, N\}$ with an (explicitly bounded) set of decision points given by the gate nodes it controls. Gate outcomes define discrete action variables $a_g \in \mathcal{A}_g$; transfer edges define preconditions and postconditions for state transitions; and control edges define feasibility constraints on actions, capturing regulatory or supervisory restrictions as admissible-action sets $\mathcal{A}_g(x)$ or inequality constraints $c(x, a) \leq 0$.

State representation and partial observability. Asset, approval, and value statuses provide the natural state factors of a factored multi-agent system. Pending states correspond to stochastic transitions; blocked states correspond to absorbing or terminal conditions (with an explicit qualifier). AGDN does not commit to an information structure, but it makes observability questions explicit: control edges and gate ownership indicate which agents impose constraints, while cross-sphere flows indicate where private information and accountability boundaries arise. This enables downstream analysts to select an appropriate formalism (e.g., stochastic games under full observability, or Dec-POMDPs under partial observability).

Utility linkage via derivation. Derivation relations ($\lozenge \dashrightarrow \square$) specify which value representations depend structurally on which assets. In control terms, derivation links define reward factors or value functions, e.g., $V_{\text{token}} = f(\text{asset state}, t, \delta)$, making explicit which legal and temporal contingencies must enter the objective specification of any optimization or equilibrium analysis.

Table S3 summarizes the mapping from AGDN constructs to multi-agent control model components.

Table S3: Indicative AGDN to multi-agent control and game-model mappings.

AGDN construct	**Multi-agent counterpart**	**Operational interpretation**
Sphere (U/I/G/Hyb)	Agent i with type $\kappa(i)$	Decision-maker with bounded authority and role logic
Gate (INST/ALG/HYB)	Action node $a_g \in \mathcal{A}_g$	Discrete control input; mode indicates human, code, or hybrid execution
Asset / Approval	State factor x_t^k	Legal/regulatory state variables (grant, pending, revoked)
Transfer edge	State transition operator	Preconditions and effects for moving artefacts across stages
Control edge	Constraint $c(x, a) \leq 0$	Regulatory oversight; restricts feasible actions
Derivation edge	Reward factor $r(x, a)$	Links economic value to underlying legal assets
Time marker $\lvert\tau\rvert$	Temporal constraint	Deadlines, stochastic delays, term limits

Interpretive payoff. This compilation perspective strengthens AGDN's role in DSS workflows: it clarifies how a diagrammatic structuring artifact can produce the variable set, action set, constraints, and value dependencies required by multi-agent control and game-theoretic models, while preserving AGDN's descriptive stance. AGDN uses the term *actor* to avoid behavioral assumptions; the term *agent* is reserved for downstream formalizations (multi-agent control, games) derived from AGDN diagrams.

A notation's value for decision support depends not only on representational clarity but also on its capacity to inform downstream formal analysis. The preceding sections have claimed that AGDN diagrams systematically map to canonical DSS model constructs.

This subsection substantiates that claim through a concrete, complete transformation of the SPC pathway (the SPC figure in the main paper) into three formal model types: decision tree, influence diagram, and mixed-integer optimization formulation.

The worked example serves three purposes. It demonstrates, first, that AGDN's structural explicitness translates directly to formal model constraints, reducing ad hoc judgment in problem formulation. It also reveals how structural misspecification—actor elision, gate collapse, status conflation, convergence oversight—becomes visible when AGDN precedes formal modeling. Not least, it provides a template transferable to other governance contexts: the same transformation logic applies to MDR pathways, technology transfer scenarios, or tokenization structures.

S4.5 Decision Tree Formulation

Decision trees represent sequential decisions and uncertain outcomes as branching structures, with nodes corresponding to decision points or chance events and branches corresponding to alternatives or outcomes [Raiffa, 1968].

AGDN Input Identification. The SPC diagram contains the following elements requiring representation: two institutional gates (g_{MA} for marketing authorization, g_{SPC} for certificate application), two approval-dependent assets (MA status, SPC status), one prerequisite asset (patent $\square_{\mathrm{pat}}$), convergent dependency (SPC requires both patent validity and MA grant), and stochastic regulatory delay (τ_{reg}).

Mapping Rules. AGDN elements map to decision tree constructs as follows:

- Gates annotated (Inst) → Decision nodes (firm chooses to apply or not)
- Approval elements with pending status → Chance nodes (authority grants or refuses)
- Transfer edges → Sequential ordering of nodes
- Convergent dependencies → Conditional branches (SPC branch exists only if MA granted)
- Time markers → Temporal discount factors or branch probabilities

Formal Decision Tree Structure. The SPC pathway induces the following tree structure, where decision nodes (d) represent firm choices and chance nodes (c) represent authority decisions:

$$\text{Root} \to d_{\mathrm{MA}} \to \begin{cases} \text{Apply} \to c_{\mathrm{MA}} \to \begin{cases} \text{Grant } (p_{\mathrm{MA}}) \to d_{\mathrm{SPC}} \to \begin{cases} \text{Apply} \to c_{\mathrm{SPC}} \to \begin{cases} \text{Grant } (p_{\mathrm{SPC}}) \to V_{\mathrm{SPC}} \\ \text{Refuse} \to V_{\mathrm{pat}} \end{cases} \\ \text{No SPC} \to V_{\mathrm{pat}} \end{cases} \\ \text{Refuse} \to V_0 \end{cases} \\ \text{No MA} \to V_0 \end{cases} \tag{1}$$

where d_{MA} and d_{SPC} denote decision nodes, c_{MA} and c_{SPC} denote chance nodes, p_{MA}

and p_{SPC} denote approval probabilities, and V_{SPC}, V_{pat}, V_0 denote terminal payoffs.

Key Translation Insight. The AGDN diagram's convergent dependency (two arrows entering g_{SPC}) translates to a structural constraint: the SPC decision node d_{SPC} appears *only* in branches where MA is granted. A naive modeler might place d_{SPC} at the root level, treating MA and SPC as independent decisions. The AGDN representation makes this error visible before tree construction begins.

S4.6 Influence Diagram Formulation

Influence diagrams compactly represent probabilistic dependencies and information flow using directed acyclic graphs with typed nodes [Howard and Matheson, 1984, Shachter, 1986].

AGDN Input Identification. The same elements apply, with additional attention to information structure: which decisions observe which states before being made.

Mapping Rules.

- Gates → Decision nodes (rectangles)
- Approval elements → Chance nodes (ovals)
- Asset/Value elements → Value nodes (diamonds) or deterministic nodes
- Control edges → Informational arcs (what decision-maker observes)
- Transfer edges → Relevance arcs (probabilistic dependencies)
- Derivation edges → Functional arcs to value node

Formal Influence Diagram Structure.

$$
\begin{aligned}
D &= \{d_{\text{MA}}, d_{\text{SPC}}\} && \text{(decision nodes)} \\
C &= \{S_{\text{pat}}, c_{\text{MA}}, c_{\text{SPC}}\} && \text{(chance nodes)} \\
V &= \{v\} && \text{(value node)}
\end{aligned}
$$

Informational arcs: $S_{\text{pat}} \to d_{\text{MA}}$ (patent status known before MA decision); $c_{\text{MA}} \to d_{\text{SPC}}$ (MA outcome observed before SPC decision).

Relevance arcs: $d_{\text{MA}} \to c_{\text{MA}}$ (application affects approval); $d_{\text{SPC}} \to c_{\text{SPC}}$; $S_{\text{pat}} \to c_{\text{SPC}}$ (patent validity affects SPC eligibility).

Value arcs: $c_{\text{MA}} \to v$; $c_{\text{SPC}} \to v$; $S_{\text{pat}} \to v$.

Key Translation Insight. The AGDN control edge from patent office to patent asset translates to an informational arc $S_{\text{pat}} \to d_{\text{MA}}$, indicating that patent status is *observed* before the MA decision. The convergent dependency at g_{SPC} generates two relevance arcs into c_{SPC}. Without the AGDN diagram, a modeler might omit the patent validity arc, generating an influence diagram that permits SPC pursuit regardless of patent status.

S4.7 Optimization Model Formulation

Mixed-integer programming (MIP) formulates decision problems as constrained optimization over discrete and continuous variables [Nemhauser and Wolsey, 1988].

AGDN Input Identification. For a portfolio of n drug candidates, each with potential SPC, the diagram structure applies to each candidate with shared budget constraints.

Mapping Rules.

- Gates → Binary decision variables
- Approval probabilities → Expected value coefficients
- Convergent dependencies → Inequality constraints (prerequisite conditions)
- Asset status (pending) → Probability-weighted objective terms
- Time markers → Temporal discount factors

Formal MIP Formulation.

Decision Variables:

$$x_i^{\text{MA}} \in \{0,1\} \quad \forall i \in I \qquad \text{(pursue MA for drug } i\text{)}$$
$$x_i^{\text{SPC}} \in \{0,1\} \quad \forall i \in I \qquad \text{(pursue SPC for drug } i\text{)}$$

Objective (expected net present value):

$$\max \sum_{i \in I} \left[p_i^{\text{MA}} \cdot p_i^{\text{SPC}} \cdot \delta^{\tau_i} \cdot V_i^{\text{SPC}} \cdot x_i^{\text{SPC}} - c_i^{\text{MA}} \cdot x_i^{\text{MA}} - c_i^{\text{SPC}} \cdot x_i^{\text{SPC}} \right]$$

Constraints:

$$x_i^{\text{SPC}} \leq x_i^{\text{MA}} \qquad \forall i \in I \quad \text{(gate sequencing)} \tag{2}$$
$$x_i^{\text{SPC}} \leq S_{i,\text{pat}} \qquad \forall i \in I \quad \text{(patent prerequisite)} \tag{3}$$
$$\sum_{i \in I} \left(c_i^{\text{MA}} \cdot x_i^{\text{MA}} + c_i^{\text{SPC}} \cdot x_i^{\text{SPC}} \right) \leq B \qquad \text{(budget)} \tag{4}$$

where $p_i^{\text{MA}}, p_i^{\text{SPC}}$ are approval probabilities, δ is the discount factor, τ_i is expected regulatory delay, V_i^{SPC} is SPC value, $c_i^{\text{MA}}, c_i^{\text{SPC}}$ are application costs, $S_{i,\text{pat}} \in \{0,1\}$ indicates patent validity, and B is budget.

Key Translation Insight. Constraints (2) and (3) directly encode the convergent dependencies visible in the AGDN diagram. The two arrows entering the SPC gate (from MA approval and from patent asset) become two inequality constraints ensuring that SPC pursuit is conditional on both prerequisites. A naive formulation with single decision variable $x_i \in \{0,1\}$ (pursue SPC) would lack these constraints, potentially selecting drugs without valid patents or approved MAs, generating infeasible solutions.

S4.8 Transformation Principles: Summary

Table S4 consolidates the mapping patterns across all three model types.

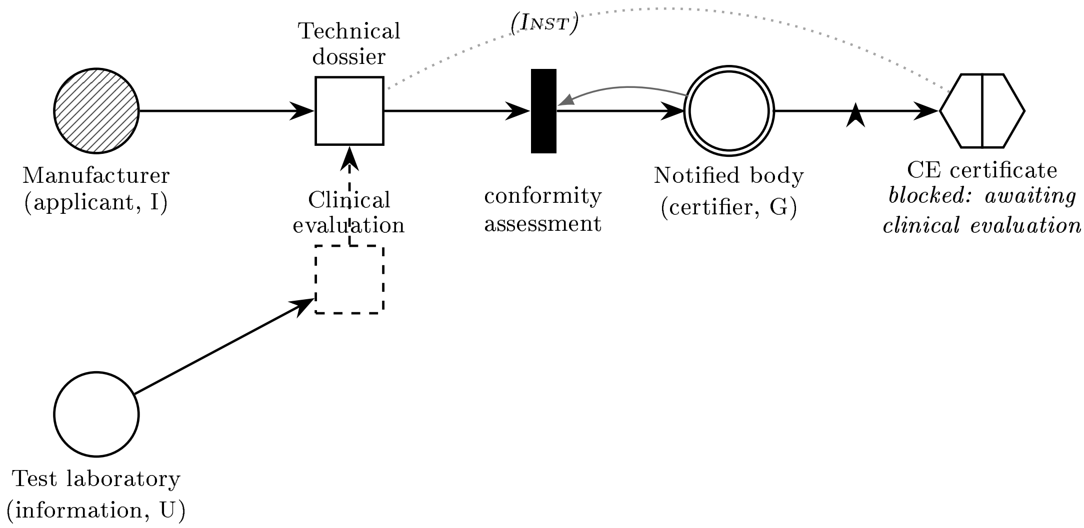
Table S4: AGDN to DSS model-transformation mappings.

AGDN Element	Decision Tree	Influence Diagram	MIP Optimization
Gate (Inst)	Decision node	Decision node (rectangle)	Binary variable $x \in \{0, 1\}$
Approval (pending)	Chance node	Chance node (oval)	Probability coefficient p
Asset	State variable	Deterministic node	Parameter $S \in \{0, 1\}$
Transfer edge	Branch sequence	Relevance arc	(implicit in sequencing)
Control edge	Observation timing	Informational arc	Constraint coefficient
Convergent dependency	Conditional subtree	Multiple arcs to node	Inequality constraint
Time marker	Discount factor	Temporal index	Discount δ^τ
Derivation edge	Terminal payoff link	Value arc	Objective coefficient

The worked example demonstrates that AGDN's explicit structural elements, particularly convergent dependencies and status distinctions, systematically inform formal model construction. This is not automatic translation; human judgment remains essential for parameter estimation, probability assessment, and objective specification. What AGDN provides is a structured intermediary representation that makes the decision architecture visible before formal modeling begins, reducing the likelihood that structural features are inadvertently omitted.

S5 Formative Expert Walkthrough (Full Report)

To assess AGDN's practical usability, a formative evaluation was conducted following established design science research evaluation protocols [Venable et al., 2016]. Our validation logic follows established verification, validation, and evaluation (V&V&E) reasoning for modeling methods by combining feasibility demonstration, expert-based assessment, and task-based comprehension measures [Kahraman and Bilgen, 2015, Ralyté et al., 2025]. The evaluation aimed to establish baseline plausibility for comprehension and construction, not statistical generalization.



Reading: the manufacturer (I) submits the technical dossier through the institutional conformity-assessment gate (INST) controlled by the notified body (G); the test laboratory (U) supplies the clinical evaluation but does not control the certification decision. The CE certificate is blocked (bar overlay) and is unblocked once the pending clinical evaluation completes. Solid = active, dashed = pending, bar = blocked; gray arrow = control, dotted = derivation.

Figure S4: Medical-device certification pathway under the EU MDR, used as the comprehension stimulus in the formative walkthrough. The manufacturer (I) submits the technical dossier through the institutional conformity-assessment gate controlled by the notified body (G); the test laboratory (U) supplies the clinical evaluation but does not control certification. The CE certificate is blocked (bar overlay) until the pending clinical evaluation completes.

S5.1 Participants and Sampling Strategy

To assess the cognitive effectiveness and cross-domain applicability of AGDN beyond anecdotal evidence, a formative expert evaluation was conducted with a stratified sample of N=25 domain experts. The sample size was selected to ensure coverage across the four professional strata and to support stable descriptive estimates; qualitative coding proceeded until no new issue categories emerged in the final tranche of participants.

The panel was structured to mirror the multi-actor environments modeled by the notation. A stratified sampling strategy ensured balanced representation across the four institutional spheres central to AGDN's design:

- **Legal / Industry (n=7):** Patent attorneys and in-house IP counsel from law firms and corporate legal departments. Focus: Gate precision, status encoding, legal accuracy of decision representations.
- **Regulatory / Government (n=6):** Regulatory affairs managers and former patent office examiners with experience in pharmaceutical, medical device, or biotechnology sectors. Focus: Control flow accuracy, approval process fidelity.
- **Academic / Transfer (n=6):** Technology transfer office managers and innovation researchers from universities and research institutions. Focus: University-industry boundary representation, value flow modeling.

- **Tech / Hybrid (n=6):** Blockchain architects, smart contract developers, and legal engineers with tokenization experience. Focus: Algorithmic gate representation, hybrid governance structures.

Recruitment followed purposive sampling through professional associations and academic networks in Europe. Participants were senior practitioners and researchers with substantial domain experience across the four professional communities, which gives the formative round its ecological validity: participants have the expertise to judge whether AGDN captures governance-relevant distinctions. We did not record precise demographic counts for this formative round. No monetary compensation was provided; participants received a summary report of findings.

S5.2 Procedure

To minimize instructor bias and ensure replicability, all participants followed an identical standardized protocol:

Phase 1: Self-paced tutorial (10 minutes). Participants received a one-page notation summary (core elements, sphere variants, connector types, status encoding) and a short annotated reference diagram (SPC process, the SPC figure in the main paper). No verbal instruction was provided; participants read materials independently and could request clarification only via written questions answered from a standardized FAQ document.

Phase 2: Comprehension walkthrough (15 minutes). Participants worked through a comprehension instrument covering sphere-type identification, gate-mode identification, status-encoding interpretation, and dependency tracing, using the MDR certification diagram (Figure S4) as stimulus. In this formative round responses were used to surface points of confusion rather than scored into a per-participant accuracy figure; we therefore report the comprehension results qualitatively below.

Phase 3: Construction task (15 minutes, qualitative). Participants translated a textual vignette describing a simplified technology licensing scenario into an AGDN diagram. Scoring: structural completeness (required elements present, correctly connected, appropriately annotated) assessed against a reference solution.

Phase 4: Semi-structured feedback (5 minutes). Participants provided written responses to three open questions: (1) "Which elements were most/least intuitive?"; (2) "What distinctions were difficult to represent?"; (3) "What symbols or conventions would you add?" Responses were coded thematically to identify systematic issues.

S5.3 Results

Comprehension walkthrough. Across the four question blocks (sphere-type and gate-mode identification, status-encoding interpretation, and dependency tracing), participants read the diagrams with little difficulty. The points that did cause hesitation clustered on status-encoding interpretation rather than on identifying spheres or connectors, which informed the later emphasis on the active/pending/blocked encoding. We report this qualitatively; no per-participant accuracy score was recorded in this formative round.

Across the stratified sample, participants decoded the separation of relationship type and process status without notable difficulty, which suggests that the type–status distinction is accessible to domain experts irrespective of their primary professional background.

Cross-stratum consistency. Whether comprehension transfers across professional communities with different training backgrounds is a central question for any multi-actor notation. In this formative round we did not record per-prompt response data at a granularity that would support a formal between-group test, so we report the observation qualitatively: across the four strata (legal/industry, regulatory/government, academic/transfer, and technical/hybrid) we saw no systematic difference in how readily participants read the diagrams, and no stratum stood out as struggling with the core constructs. We treat this as suggestive evidence of cross-domain accessibility, not as a measured null effect; the controlled experiment reported below provides the quantified cross-stratum evidence on which the boundary-object claim rests.

Construction task. When asked to build a diagram against a reference solution, participants reproduced the core structure reliably: all correctly identified sphere boundaries and connector types, while the more frequent omissions concerned secondary annotations such as gate-mode markers and time markers. Two researchers reviewed the constructed diagrams and discussed disagreements until they agreed; we did not compute a formal inter-rater statistic for this small formative set. The pattern suggests that the core representational logic transfers even when annotation discipline is incomplete.

Qualitative feedback. Feedback was analysed thematically. Recurring points concerned the representation of conditional branches, a wish for an explicit "rejection" status beyond the bar overlay, and uncertainty about how to depict collective actors. These observations fed directly into the revisions described above.

On sample size. We make no formal saturation claim for the formative round. The recurring themes had stabilised before the full set of twenty-five participants had been seen, in the sense that later participants raised variants of concerns already recorded rather than new categories, which is the ordinary basis on which a formative study of this size is judged adequate [Guest et al., 2006]. We present this as a pragmatic judgement, not as a bootstrap-derived stopping rule.

S5.4 Interpretation

The formative expert evaluation establishes baseline plausibility for AGDN along three lines: comprehension, error patterns, and cross-stratum readability.

- **Comprehension supports learnability.** Participants decoded the diagrams reliably after a short self-paced tutorial, which speaks against the concern that a domain-specific notation imposes a prohibitive learning burden.
- **Error patterns reveal training priorities, not notation deficits.** The omissions that did occur concerned secondary annotations (gate markers, time markers) rather than the core constructs, which every participant applied correctly. The distinction matters: notation redesign addresses comprehension failures, whereas annotation omissions are a matter of training and practice.
- **Cross-stratum readability supports the multi-actor design intent.** Sampling deliberately across institutional spheres, we saw no stratum disadvantaged by the notation's design choices, which is consistent with the cross-domain accessibility the notation aims for. The quantified test of this property comes from the controlled experiment, not from this formative round.

Limitations of this evaluation. This formative round is qualitative. It establishes feasibility and surfaces design refinements rather than measuring effectiveness, and it involves no comparison condition and no inferential test [Venable et al., 2016]. Purposive sampling through professional associations may bias the sample toward notation-receptive participants, which stratification across four communities only partially mitigates. Confirmatory evidence comes from the controlled experiment that follows, with randomised assignment, an explicit comparison condition, and objective outcome measures.

S6 Controlled Experiment: Full Apparatus

S6.1 Power and Sample Size

The design and its analysis follow established guidance for controlled experimentation: Montgomery's [Montgomery, 2012] principles for factorial design and randomisation, and Dennis and Valacich's [Dennis and Valacich, 2001] recommendations for conducting experimental research in information systems, which together inform the blocking, randomisation, and outcome-measurement choices set out below. An a priori power analysis (G*Power 3.1) for a two-sided Mann–Whitney U test indicated that a total sample of about $N = 50$ would detect a medium effect ($r = 0.35$) at $\alpha = 0.05$ with power 0.80. Recruitment targeted just above this threshold to allow subgroup inspection by professional background and seniority. The realised sample was $N = 52$. All recruited participants completed both tasks; there were no withdrawals and no exclusions, so the analysed sample equals the recruited sample and no intention-to-treat correction is required.

We do not report post-hoc (observed) power, since it is a deterministic function of the obtained p-value and adds no information beyond it [Hoenig and Heisey, 2001]. Both task effects are assessed through effect-size estimation with bootstrap confidence intervals, and the analysis is checked against the crossed structure of the design (period and sequence) in the robustness section below.

S6.2 Participants

Fifty-two patent practitioners and patent engineers took part (32 male, 19 female, 1 non-binary). Age was recorded in bands rather than as an exact figure: 17 participants fell in the 30–39 band, 20 in 40–49, and 15 in 50–59. Mean professional experience was 13.1 years (SD = 6.1, range 4–25). Participants were run across six small-group sessions of eight or nine persons, between March and May 2025.

Recruitment proceeded through the patent attorneys' chamber mailing list (with permission), the LES Germany/Austria newsletter, the technology-transfer offices of three research universities, and the research team's professional contacts in pharmaceutical and engineering patent departments.

Eligibility required at least three years of practice in patent prosecution, licensing, supplementary protection certificates, technology transfer, or freedom-to-operate work; current involvement in multi-actor IP governance decisions; and self-reported modelling or visualisation of complex IP governance structures at least monthly. Purely administrative or paralegal roles were excluded, as was any prior exposure to AGDN, of which none was reported. Table S5 summarises the composition.

Table S5: Sample composition by professional context.

Professional context	n	Mean exp.	SD	EPA (%)	Prior (%)
Patent attorney firms	20	14.3	7.4	75	65
Industry IP departments	19	13.2	5.5	37	84
University TTOs	13	11.0	4.5	23	0
Total	52	13.1	6.1	48	56

Note. Mean exp. = mean years of professional experience; EPA = qualified European Patent Attorney; Prior = prior BPMN/UML exposure.

Twenty-five participants were qualified European Patent Attorneys and fourteen were qualified German patent attorneys. Fifty-six percent reported some prior BPMN or UML exposure, in most cases introductory and concentrated in the industry subgroup; none had formal training in visual notation or cognitive science, and none of the technology-transfer participants reported prior process-notation exposure. The sample thus combines high domain seniority with low notational priming, which is the configuration in which a structuring aid is most plausibly tested on its own merits.

S6.3 Randomisation

Within each session, participants were assigned to Group A or Group B by a computer-generated random sequence (block size four, stratified by professional context). An independent assistant not otherwise involved in the study carried out the allocation. Participants knew that they were using either a “new notation” or “free sketching” but were not told that the other group worked under a different condition. The suspicion check (the post-survey reported in the main paper) confirmed that none of the fifty-two participants correctly inferred the between-group comparison.

S6.4 Materials and Tasks

Two ecologically valid scenarios were developed with two European patent attorneys (one in-house at a pharmaceutical firm, one in private practice) and a senior technology-transfer director, then piloted with six practitioners outside the main sample for realism, complexity, comprehensibility, and timing (target 30–35 minutes per task). The pilot led to shortened case descriptions, an acronym glossary, and clarified wording.

Task 1 (modelling) presented “BioShield SPC”, a governance network around a supplementary protection certificate involving a university spin-off, an originator manufacturer, the German Patent and Trade Mark Office, a contract research organisation, and two prospective licensees. Participants had 35 minutes to produce a visual model capturing the actors and their formal decision rights, the temporal status (active, pending, blocked) of the core processes, the information and approval flows, and the dependencies and preconditions among them. The treatment condition supplied a one-page AGDN quick reference (Figure S1) and a blank A3 sheet; participants could ask syntax questions but received no help on case content. The control condition received the same A3 sheet with the instruction to use any notation of their choice (text, boxes, arrows) and a short “sketching tips” briefing, with no exposure to any systematic notation.

Task 2 (analysis) presented “NeuroFlex Licensing”, a negotiation scenario based on a fully anonymised real case involving a brain–computer interface technology, two patent families, a spin-off holding an exclusive option, the interest of a large medical-technology company, and a freedom-to-operate challenge from a competing start-up. In 35 minutes participants identified the three most critical bottlenecks capable of delaying or blocking a licence agreement, proposed two alternative governance configurations (one “accelerated”, one “risk-mitigated”), rated blocking risk per principal actor under each configuration, and gave a short written justification of their preferred configuration. The treatment condition could reuse or rebuild an AGDN diagram; all but two participants chose to rebuild, reporting that it refreshed their grasp of the case. The control condition again used free sketches and text.

S6.5 Scoring and Inter-Rater Reliability

A six-criterion analytic rubric was developed over three rounds. Three experts (a European patent attorney, an innovation-management researcher, and a visual-notation researcher) independently drafted candidate criteria; overlapping items were merged into six discriminating dimensions per task; two pilot rounds ($n = 6$ practitioners outside the main sample) were scored independently and discrepancies discussed until wording was sharpened. Each criterion was scored 0–4, and the per-task total is the unweighted mean of its six criteria, itself on the 0–4 scale. Table S6 reproduces the criteria for both tasks.

Table S6: Analytic scoring rubric: six equally weighted criteria per task.

Criterion	Anchor at 4 points
Task 1 (modelling)	
Status modelling (active/pending/blocked)	All relevant states correct and visually distinct
Actor and decision-right coverage	$> 85\%$ of actors with correct rights and duties
Dependency accuracy	All key dependencies and preconditions correct
Diagram clarity	Legible, immediately interpretable, professional
Analytical depth	Reasoning over the structure beyond mere depiction
Remaining quality criteria	Consistent quality on secondary aspects
Task 2 (analysis)	
Bottleneck identification	Binding constraints correctly located and ranked
Alternative configurations	Viable alternative arrangements proposed
Risk assessment	Governance risks identified and weighed
Actor-specific justification	Reasoning attributed to the correct actor
Dependency tracing	Chains of preconditions followed correctly
Cross-actor links	Inter-actor dependencies made explicit

Four trained raters (two research assistants with an IP-law background, one senior researcher, one patent attorney) scored all diagrams under anonymised participant codes in randomised order. Raters were blind to participant identity, session, and group assignment (A/B). They were not, and could not be, blind to the notation used: an AGDN diagram and a free sketch are visually distinct, so blinding to treatment condition is impossible when the conditions are themselves two different representational formats, as is standard when visually dissimilar notations are compared. To check whether this unavoidable visibility biased scoring, we ran a sensitivity analysis restricted to the rubric criteria least dependent on visual aesthetics (status modelling, dependency accuracy); the treatment effects on these criteria matched the overall pattern, which argues against an aesthetic-halo bias. Because all four raters scored every diagram, inter-rater reliability is computed on the full sample ($n = 52$ per task). Two-way random-effects, absolute-agreement single-rater intraclass correlation coefficients were $\mathrm{ICC}(2,1) = 0.80$ (95 % CI $[0.71, 0.87]$) for Task 1 and $\mathrm{ICC}(2,1) = 0.81$ ($[0.73, 0.88]$) for Task 2, in the good range;

the four-rater coefficients on which the analysis scores rest were $\mathrm{ICC}(2,4) = 0.94$ for both tasks. Per-criterion quadratic-weighted κ ranged from 0.46 to 0.88, the lowest being diagram clarity—the most interpretive criterion, where banded agreement is only moderate—and the highest the more concrete bottleneck and coverage criteria; the consensus scores absorb this criterion-level variation.

Because the single-rater reliability is good rather than excellent, we checked whether the between-condition effects depend on the rater averaging. Repeating the primary Mann–Whitney comparison on each rater's scores separately, all four raters individually reproduce the analysis-task effect at $p < 0.001$, whereas on the modelling task two raters reach significance ($p = 0.0004$ and $p = 0.003$) and two do not ($p = 0.15$ and $p = 0.21$). The analysis-task result is therefore robust to rater identity, while the weaker modelling result emerges from the four-rater consensus and is not present in every individual rater's scores—an asymmetry consistent with the smaller modelling effect size reported in the main text. The consensus score itself is reliable ($\mathrm{ICC}(2,4) = 0.94$), so disattenuation would raise the observed correlations by only three to four percent; we therefore report observed, uncorrected effects throughout. Where raters diverged, the final criterion value is the four-rater mean; disagreements wider than one scale point on a single criterion (8% of diagrams) were re-examined before the mean was fixed.

The per-criterion scores released as data file D3 are the means of the four raters' criterion scores, rounded to two decimals, and are therefore row-level reproducible from the rater matrix; the task totals in D1 are the unrounded means, so summing the six rounded D3 criteria recovers a task total to within about 0.02. One criterion-level result sits at the significance boundary and should be read as such: on the modelling task, dependency accuracy differs by +0.21 with $p = 0.04$, a marginal separation that we do not over-interpret. It is the weakest of the modelling-task criteria that reach the conventional threshold, its confidence interval runs close to zero, and with six criteria compared per task no correction for multiplicity has been applied; we report it because it is consistent with the direction of every other criterion, not as an independent confirmatory finding. The robust modelling-task result is status modelling, and the robust effect overall is on the analysis task, where every criterion separates the conditions and all four raters reproduce the effect individually.

S6.6 Procedure

Each session ran 90 minutes under a fixed protocol: welcome, consent, and a pre-survey (8 min); a common domain briefing on SPC and technology-transfer governance with a worked non-AGDN example (12 min), to which treatment groups added a five-minute AGDN introduction and control groups a five-minute free-sketching briefing; randomised allocation and separation into different rooms (2 min); Task 1 with a visible countdown

(35 min); a short break (5 min); Task 2 with countdown (35 min); and a post-survey covering perceived usefulness, cognitive load (NASA-TLX short form), the suspicion check, and open feedback, followed by debriefing (8 min). All materials were presented on identical laptops with a simple drawing application permitting geometric shapes and text; the treatment groups additionally had a digital AGDN stencil. Internet access was disabled. Completion time was logged automatically and all participants finished both tasks within the limit.

S7 Worked Example: SPC Portfolio Misspecification

S7.1 Worked Example: SPC Portfolio Optimization

Consider an SPC portfolio optimization problem: a pharmaceutical firm has 10 drug candidates and wants to maximize expected SPC value subject to budget.

Formulation A: Naive (without AGDN structuring).

Decision variables: $x_i \in \{0,1\}$ for $i \in \{1,\ldots,10\}$ (pursue SPC for drug i).

Objective: $\max \sum_i V_i^{\mathrm{SPC}} \cdot x_i$

Constraints: $\sum_i c_i \cdot x_i \leq B$ (budget)

Structural errors: (1) Gate Collapse: No distinction between MA and SPC pursuit—single variable conflates two sequential decisions. (2) Convergence Oversight: No patent prerequisite constraint—model may select SPC for drugs without valid patents. (3) Status Conflation: Value V_i^{SPC} is deterministic—ignores approval probability.

Consequence: Optimal solutions may select SPC pursuit for drugs without MA approval (infeasible), for drugs without valid patents (illegal), or overestimate expected value by factor of $(1/p^{\mathrm{MA}})(1/p^{\mathrm{SPC}})$.

Formulation B: AGDN-Informed.

After constructing an AGDN diagram of the SPC pathway, the analyst observes: two institutional gates, convergent dependency at SPC gate, pending status on both MA and SPC approvals.

Decision variables:

$$
\begin{aligned}
x_i^{\mathrm{MA}} &\in \{0,1\} && \text{(pursue MA for drug } i\text{)} \\
x_i^{\mathrm{SPC}} &\in \{0,1\} && \text{(pursue SPC for drug } i\text{)}
\end{aligned}
$$

Objective: $\max \sum_i \left[p_i^{\mathrm{MA}} \cdot p_i^{\mathrm{SPC}} \cdot V_i^{\mathrm{SPC}} \cdot x_i^{\mathrm{SPC}} - c_i^{\mathrm{MA}} \cdot x_i^{\mathrm{MA}} - c_i^{\mathrm{SPC}} \cdot x_i^{\mathrm{SPC}}\right]$

Constraints:

$$
\begin{aligned}
x_i^{\text{SPC}} &\le x_i^{\text{MA}} && \forall i && \text{(gate sequencing from AGDN)} \\
x_i^{\text{SPC}} &\le S_{i,\text{pat}} && \forall i && \text{(convergent dependency from AGDN)} \\
\sum_i (c_i^{\text{MA}} \cdot x_i^{\text{MA}} + c_i^{\text{SPC}} \cdot x_i^{\text{SPC}}) &\le B && && \text{(budget)}
\end{aligned}
$$

Structural corrections: Each constraint maps directly to an AGDN feature: gate sequencing reflects the two separate gates, the patent prerequisite reflects convergent dependency, and probability coefficients reflect pending status.

Critical Point. The AGDN diagram does not automatically generate Formulation B. A human analyst must understand the mapping rules and translate structure to mathematics. But by making structure *explicit*, AGDN makes errors *visible* when the diagram is reviewed, before formal modeling begins. This is Simon's [1960] principle: effective DSS support problem structuring, not just computation.

References

Steven Alter. A work system view of dss in its fourth decade. *Decision Support Systems*, 38(3):319–327, 2004.

David Arnott and Graham Pervan. A critical analysis of decision support systems research. *Journal of Information Technology*, 20(2):67–87, 2005.

Alan R. Dennis and Joseph S. Valacich. Conducting experimental research in information systems. *Communications of the Association for Information Systems*, 7(5):1–41, 2001.

Gerd Gigerenzer and Reinhard Selten. Rethinking rationality. In Gerd Gigerenzer and Reinhard Selten, editors, *Bounded Rationality: The Adaptive Toolbox*, pages 1–12. MIT Press, Cambridge, MA, 2001.

Greg Guest, Arwen Bunce, and Laura Johnson. How many interviews are enough? an experiment with data saturation and variability. *Field Methods*, 18(1):59–82, 2006.

John M. Hoenig and Dennis M. Heisey. The abuse of power: The pervasive fallacy of power calculations for data analysis. *The American Statistician*, 55(1):19–24, 2001.

Ronald A. Howard and James E. Matheson, editors. *Readings on the Principles and Applications of Decision Analysis*. Strategic Decisions Group, Menlo Park, CA, 1984.

IBM Corporation and IPwe. IPwe and IBM seek to transform corporate patents with next-generation NFTs using IBM blockchain. IBM Newsroom press release, April 2021. Available at: https://newsroom.ibm.com/2021-04-20-IPwe-and-IBM-Seek-to-Tra nsform-Corporate-Patents-With-Next-Generation-NFTs-Using-IBM-Blockchain.

Gökhan Kahraman and Semih Bilgen. A framework for qualitative assessment of domain-specific languages. *Software and Systems Modeling*, 14(4):1505–1526, 2015.

Peter G. W. Keen and Michael S. Scott Morton. *Decision Support Systems: An Organizational Perspective*. Addison-Wesley, Reading, MA, 1978.

Jill H. Larkin and Herbert A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11(1):65–100, 1987.

James G. March and Herbert A. Simon. *Organizations*. Wiley, New York, 1958.

Douglas C. Montgomery. *Design and Analysis of Experiments*. Wiley, Hoboken, NJ, 8 edition, 2012.

Daniel L. Moody. The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35(6):756–779, 2009.

George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, New York, 1988.

Elinor Ostrom. *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, Cambridge, 1990.

Elinor Ostrom. Beyond markets and states: Polycentric governance of complex economic systems. *American Economic Review*, 100(3):641–672, 2010. doi:10.1257/aer.100.3.641.

John W. Payne, James R. Bettman, and Eric J. Johnson. *The Adaptive Decision Maker*. Cambridge University Press, Cambridge, 1993.

Jean-Charles Pomerol and Frédéric Adam. Practical decision making: From the legacy of herbert simon to decision support systems. *Decision Support Systems*, 37(1):1–16, 2004.

Daniel J. Power. *Decision Support Systems: Concepts and Resources for Managers*. Quorum Books, Westport, CT, 2002.

Howard Raiffa. *Decision Analysis: Introductory Lectures on Choices under Uncertainty*. Addison-Wesley, Reading, MA, 1968.

Jolita Ralyté et al. Verification, validation, and evaluation of modeling methods. *Software and Systems Modeling*, 2025. doi:10.1007/s10270-025-01304-2. Advance online publication.

Ross D. Shachter. Evaluating influence diagrams. *Operations Research*, 34(6):871–882, 1986.

Herbert A. Simon. A behavioral model of rational choice. *Quarterly Journal of Economics*, 69(1):99–118, 1955.

Herbert A. Simon. *The New Science of Management Decision.* Harper and Row, New York, 1960.

Ralph H. Sprague and Eric D. Carlson. *Building Effective Decision Support Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

John Venable, Jan Pries-Heje, and Richard Baskerville. Feds: A framework for evaluation in design science research. *European Journal of Information Systems*, 25(1):77–89, 2016.