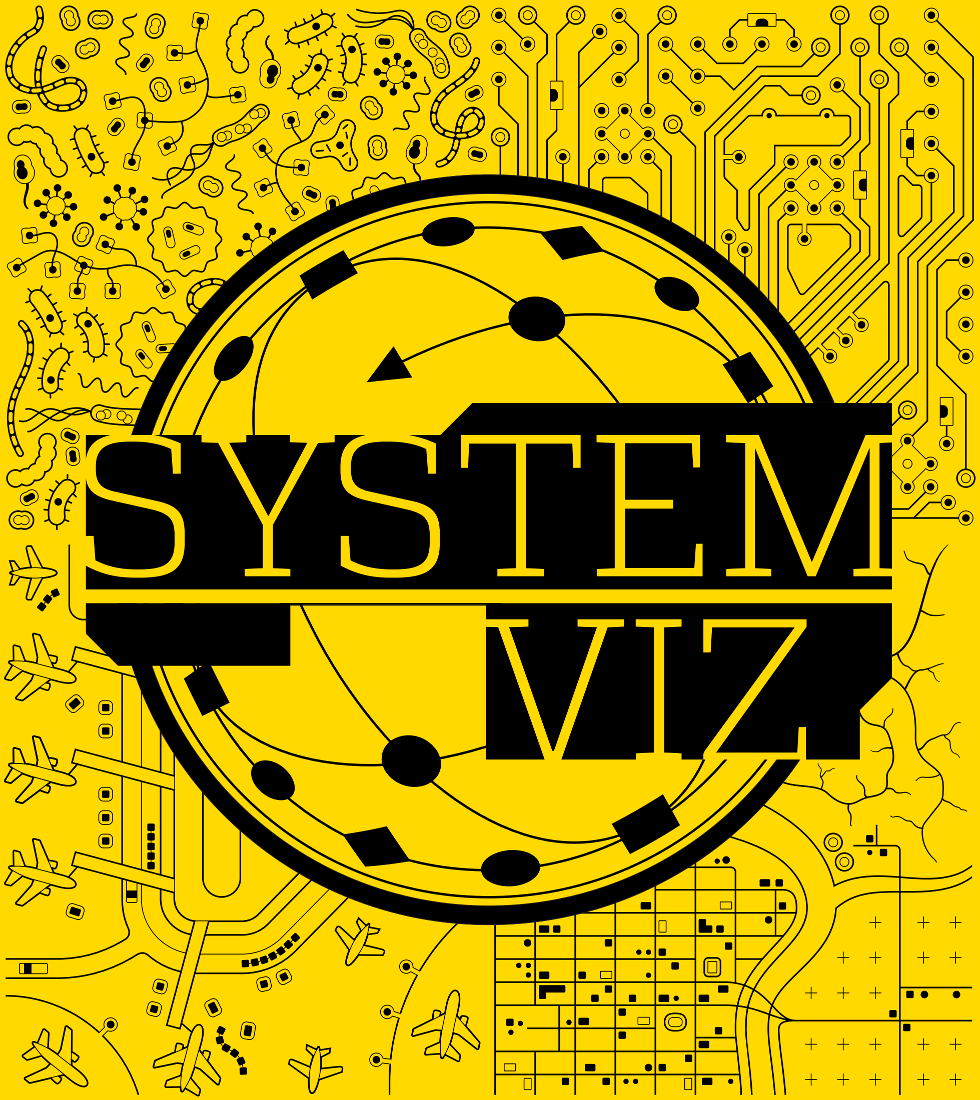


anatomy of system notations

PETER STOYKO

RSD12 ANNUAL SYMPOSIUM
SYSTEMIC DESIGN ASSOCIATION
WASHINGTON DC, USA

19 · 10 · 23

SystemViz is an ongoing research-and-design project that explores better ways of visualizing systems. Systems are difficult to grasp without some sort of visual aid. There are too many moving parts to juggle in the mind at once. Graphical notations are a popular option, particularly in scientific and engineering professions. Yet notations suffer from shortcomings. They tend to be cryptic and overwhelm the viewer with confusing tangles.

Most notations are highly specialized and hard to learn, limiting their general appeal. A stock-taking is long overdue. It is high time to figure out: (a.) what works and what doesn't; (b.) which aspects of systems are over- and under-emphasized; (c.) how diagrams and maps can be made tidy and intuitive. As an early step towards that goal, I propose a new stock-taking reference and welcome feedback from the systemic-design community.

new

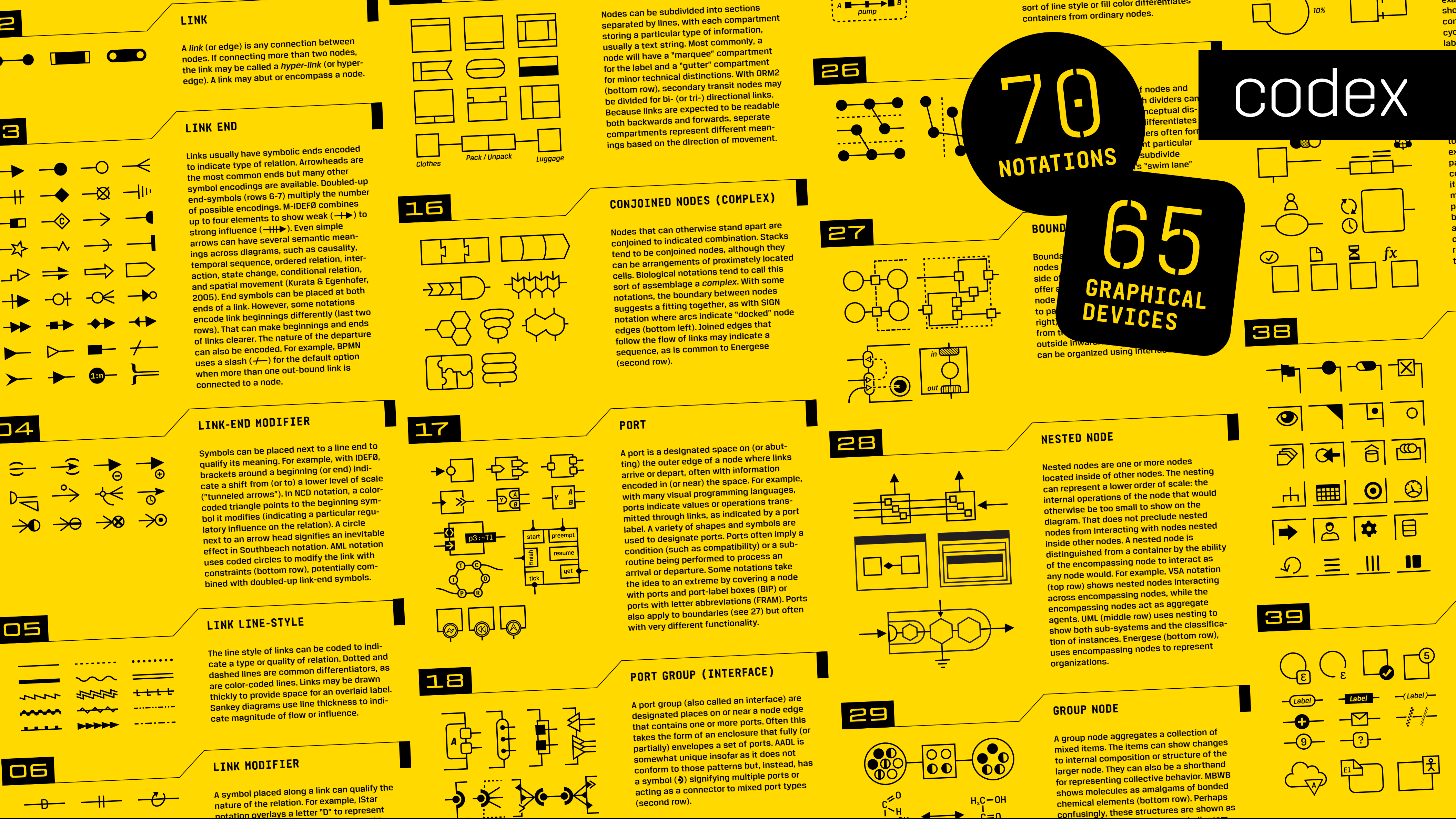
ANATOMY
OF SYSTEM NOTATIONS

codex

This presentation introduces the ***Anatomy of System Notations***, a comprehensive inventory of the visual conventions used to map and diagram systems. The inventory takes the form of an open-source visual codex. A *codex* is an organized compendium of items of interest within a particular subject-area. Items are gathered together in one place to serve as a reference material, in this case a poster. A *visual codex* shows illustrated examples, ideally more than one example per concept so as to not confuse the larger concept with a particular example. The "lumping and splitting" of items into kinds helpfully organizes under-studied subject areas. The *Anatomy of System Notations* is released with an open-source license, free for anyone to use and modify for fun or profit (with attribution). The aim is to encourage usage, especially unanticipated applications.

analogy

"Anatomy" invokes the idea of body parts, such as those take-apart models used to teach students about the human body. An analogy is being drawn. Diagram notations can also be dissected to identify all the different graphical devices in use. Moreover, different types of animal will have differently shaped body parts. Despite superficial differences, some of these body parts serve a similar function and can be grouped together as the same sort of thing. Occasionally, a highly unusual part will be discovered that sheds new light on how biology works. Likewise, graphical devices vary across notations but serve similar functions. Much can be learned from devices that fit their purpose well. And experimental graphical devices may also emerge that inspire a major rethinking about how a notation can work. Those innovations deserve special attention.

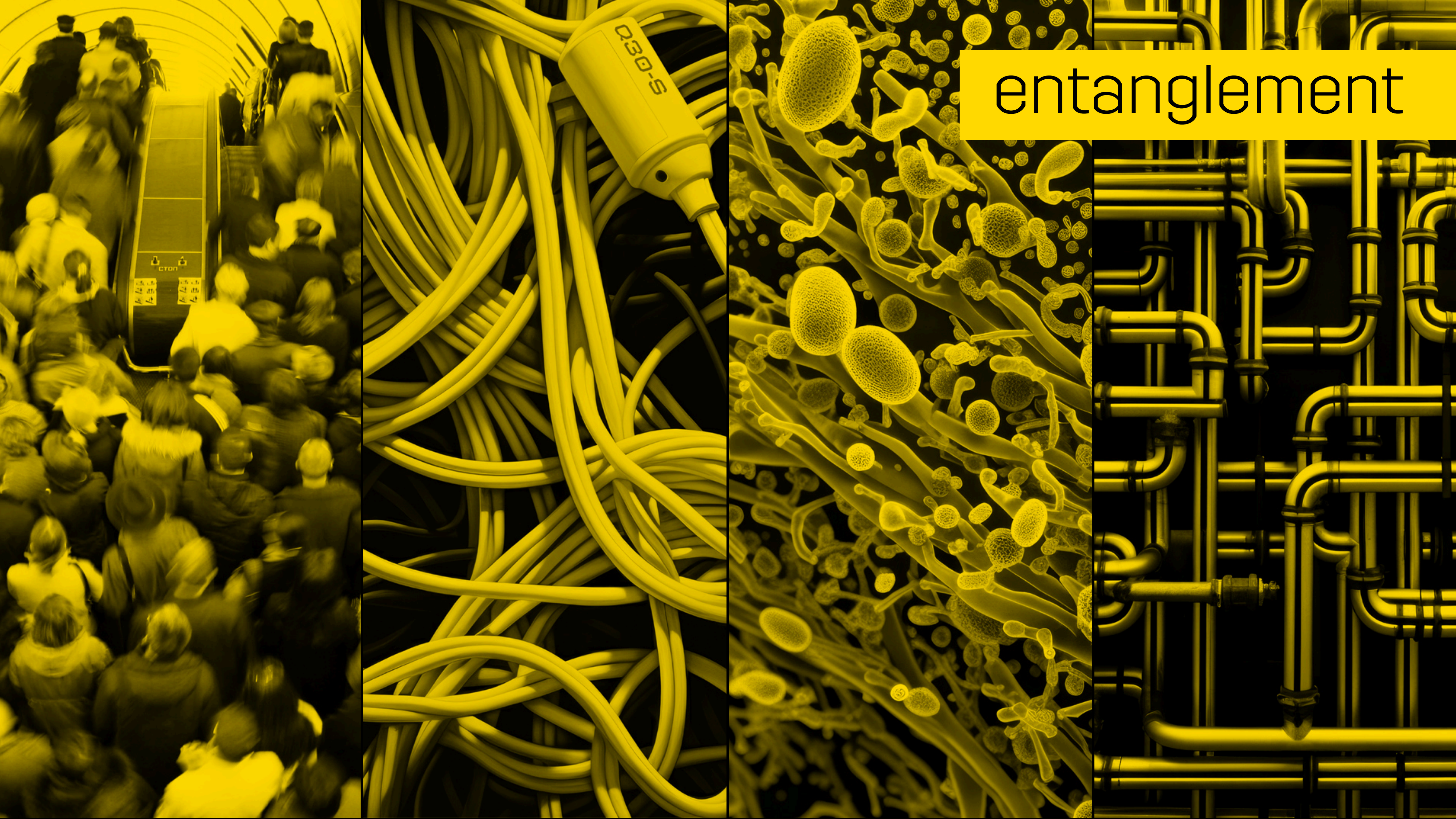


The codex reviews 70 system notations and lists 65 graphical devices. What counts as a system notation? As Donella Meadows puts it, "A system is a set of things—people, cells, molecules, or whatever—interconnected in such a way that they produce their own pattern of behavior over time."* By definition, a system has moving parts that interact. That rules out visual methods for depicting static structures or mere inputs and outputs of a system. Moreover, a notation has a stable, finite set of graphical devices to serve as points-of-reference for comparison across systems. These can be formal standards or widely observed conventions. Hundreds of named system notations exist. So what criteria determined inclusion in the codex?

* Donella H. Meadows, *Thinking in Systems: A Primer* (White River Junction, VT: Chelsea Green Publishing, 2008), p. 1.

methodology

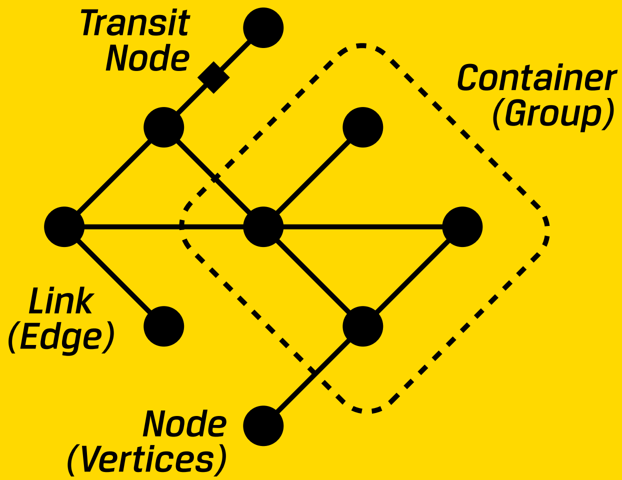
For those curious about **methodology**, the gathering of notations is best described as butterfly collecting. Much like with field biologists, long periods were spent culling through the wilds of scholarly and professional publishing to discover exotic specimens. That collecting took place over twelve years, with many deep dives into specialized disciplines and modelling practices. Hundreds of notations were discovered. Many vary in name only. Most are simple variations on network diagrams: arrowed links pointing to simply-shaped nodes, plus a few additional encodings to justify giving the notation its own name. For a notation to be included in the codex, it had to be: (a.) in widespread use as a recognized convention; or (b.) a notation with a novel graphical device, organizing concept, or set of encodings. Some 70 *reference notations* fit those criteria.



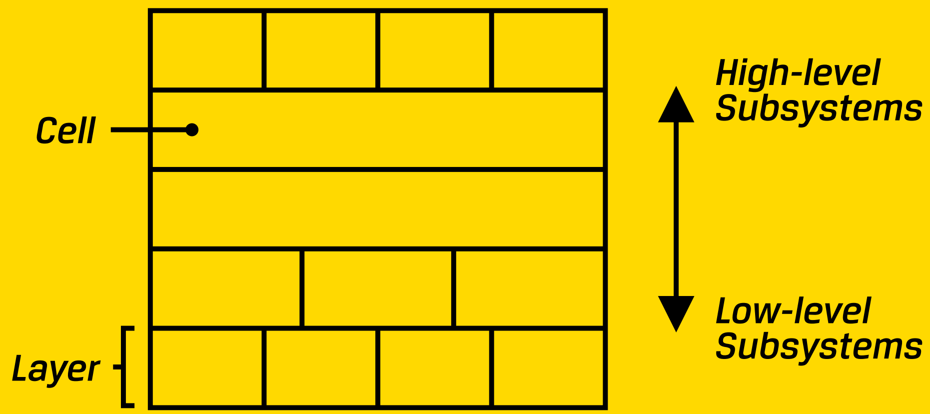
entanglement

The stock-taking matters because of an ongoing struggle to keep pace with the complexity of systems and their **entanglement** in our lives. While these notations promote systems thinking, they do so imperfectly and with a particular slant. Those biases can get "baked into" systems that a particular notation helped to create. At the same time, notations offer an opportunity to make the tangle of interconnected and overlapping systems more legible to us all. That knowledge is crucial for personal and collective agency, given how the tangle of systems pushes and pulls us around continually. A few notations may offer insightful techniques for showing how systems work—innovations that may be worth building upon. Sadly, systems notations have plateaued in their development while the system tangles become harder and harder to decipher.

basic forms



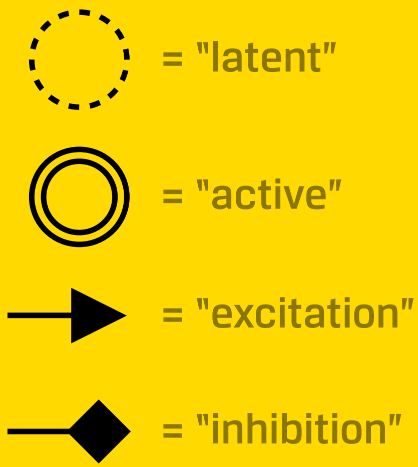
NETWORK DIAGRAM
(HYPERGRAPH)



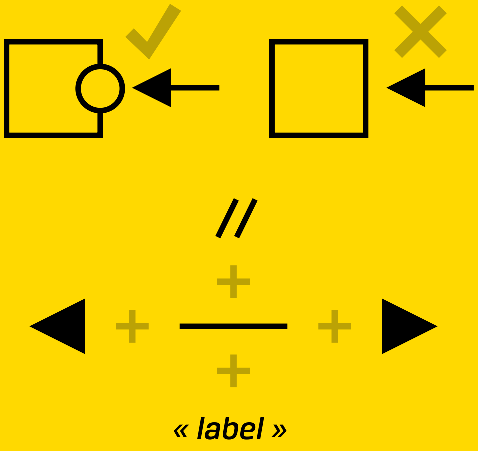
SYSTEM STACK

How do these notations work? In its **basic form**, a system diagram shows relationships between interacting parts to reveal a larger whole. That is usually done spatially with a network diagram (*hypergraph*) containing nodes (vertices) and links (edges). Links denote relations. Nodes can represent actors, objects, functions, or stations in a process. They can vary in size, shape, style, and color to indicate different types. They can be grouped using containers to indicate a contextual domain or subsystem. Systems can also be shown as abutting blocks (or "stacks") of subsystems, with positioning implying different levels of analysis or orders of functionality. Proximity or abutting boundaries imply the relations. To understand how these basic forms become a variety of distinct notations, let's go through a crash course on *visual syntax*.

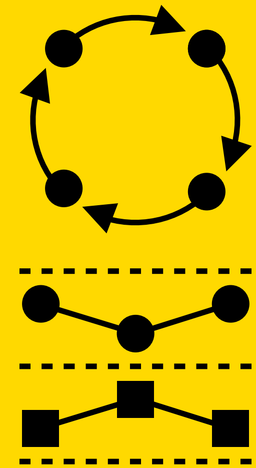
syntax



SEMANTIC ENCODINGS
(SYMBOLISM)



COMPOSITION RULES
(GRAMMAR)

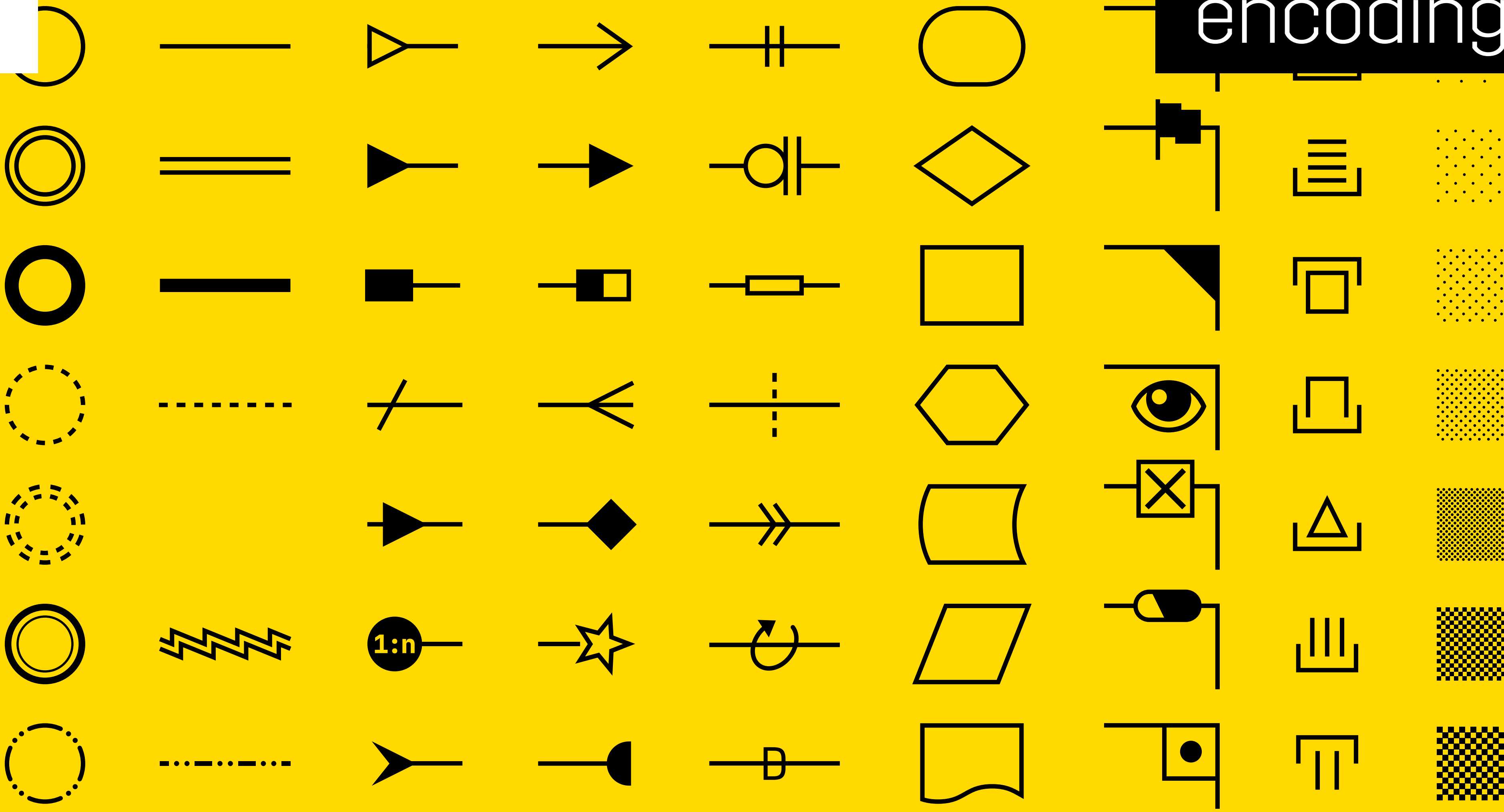


SPATIAL ARRANGEMENTS
(SYNTACTICS)

The design of diagrams follows conventions and specified rules called a **syntax**. There are at least three dimensions. First, visual items are encoded with an abstract meaning (*symbolism*) inso-far as they do not correspond a real-world object in a literal way. Second, these items are used consistently to allow the viewer to form expectations. As with a verbal-textual language, the rules of composition can be thought of as a *grammar*. Symbolism and grammar do not explain fully how diagrams are perceived and interpreted. The relative positioning of items in space has to be meaningful (*syntactics*).* Particular arrangements of items (such as loops or lanes) become recognizable and convey information at a glance. Let's explore each dimension further.

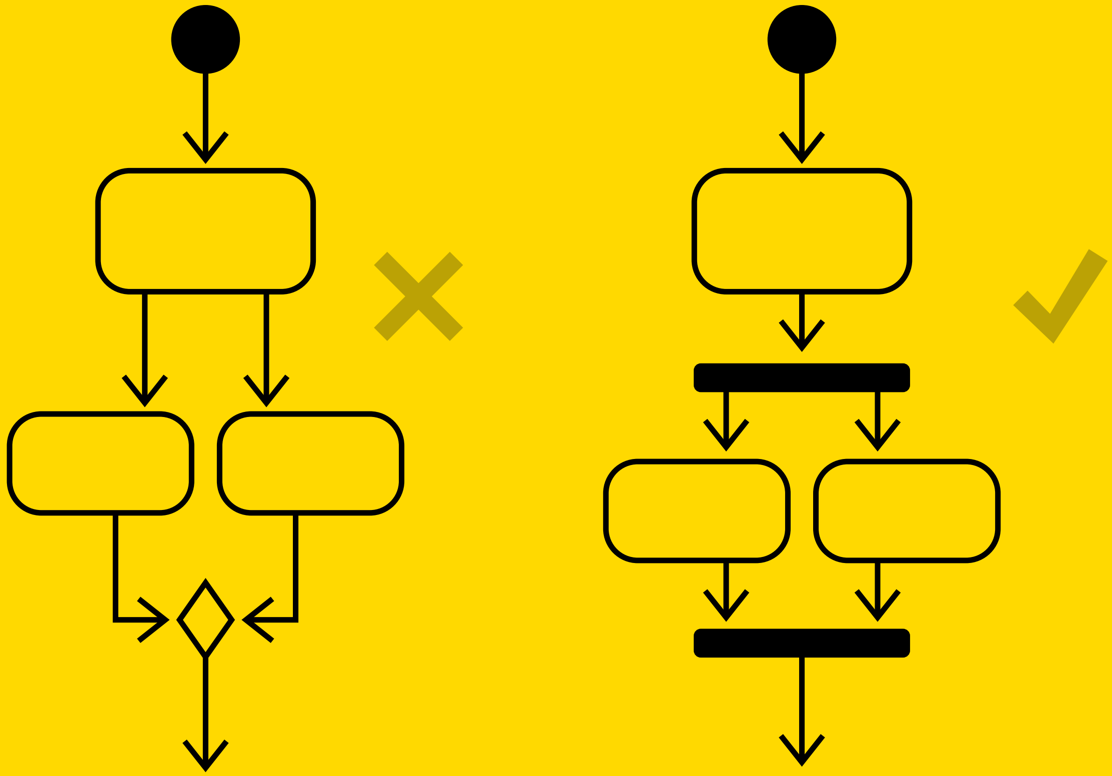
* Yuri Engelhardt, "Syntactic Structures in Graphics," *IMAGE: Zeitschrift für interdisziplinäre Bildwissenschaft* , vol. 5, no. 3 (2007), pp. 23-35.

1 encodings



Most diagram items are stand-ins for parts of actual systems. For example, a solid-line link may represent "physical flow," whereas a dashed-line link represents "communications signal." A solid arrow-head may mean "excitation" and a hollow arrow-head may mean "inhibition." Circular nodes may be "sellers" while a rectangular ones are "buyers." One symbol may represent "valve" while another is "switch." Red may be reserved for a failure state and amber a warning state. **Semantic encodings** are not obvious unless they have a widely recognized cultural meaning—most have to be learned. The challenge is that there is a limit to the number of encodings a viewer can be expected to remember. Complicated technical notations take a lot of effort to learn partly because of the sheer amount of symbolism that has to be memorized to make sense of the diagram.

Unified Modelling Language (UML)



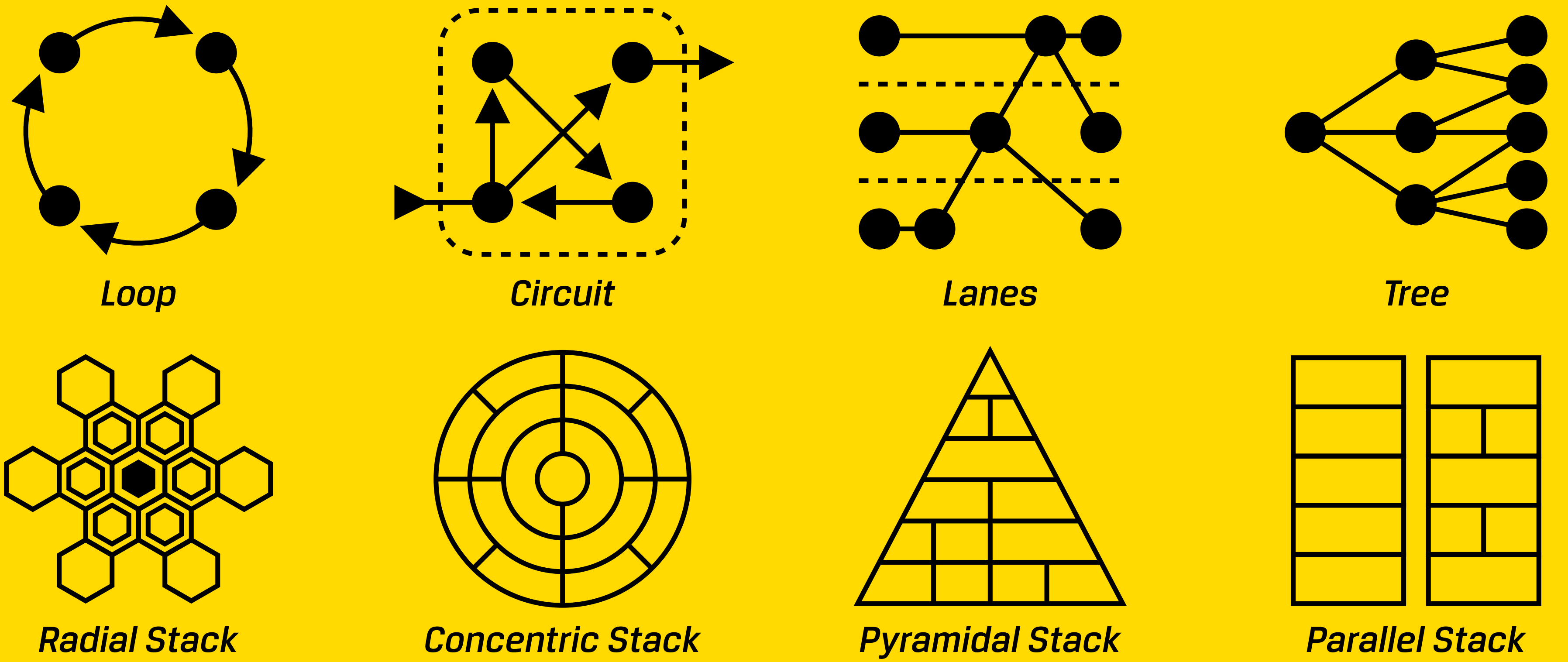
Source: Chren et al., 2019

Are the parts of the system labeled? Where do the labels go? How do the links connect to the nodes? Directly or through an intermediate object? These **composition rules** set expectations about how the parts of a notation fit together. They function like grammar. If words could appear in any order within a sentence, the sentence would be hard to understand. You might get the gist but the potential for misunderstanding would be great.

Likewise, a diagram without consistent rules about how things fit together might convey the gist but not reliably. One complaint about free-form system sketches is that they routinely set and defy expectations about how everything connects up.

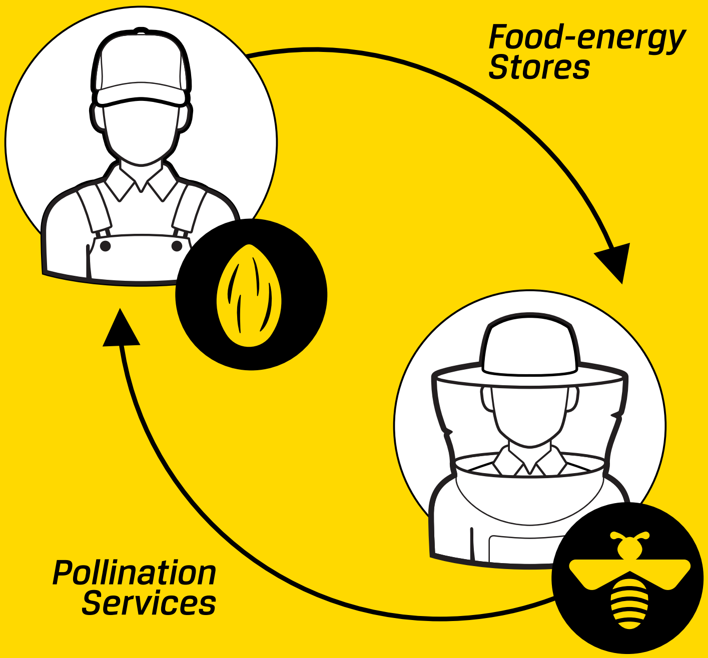
* The example above is adapted from an inventory of mistakes from UML diagrams, see: Stanislav Chren et al., "Mistakes in UML Diagrams: Analysis of Student Projects in a Software Engineering Course," Lasaris Lab, Masaryk University, Brno, Czech Republic (2019).

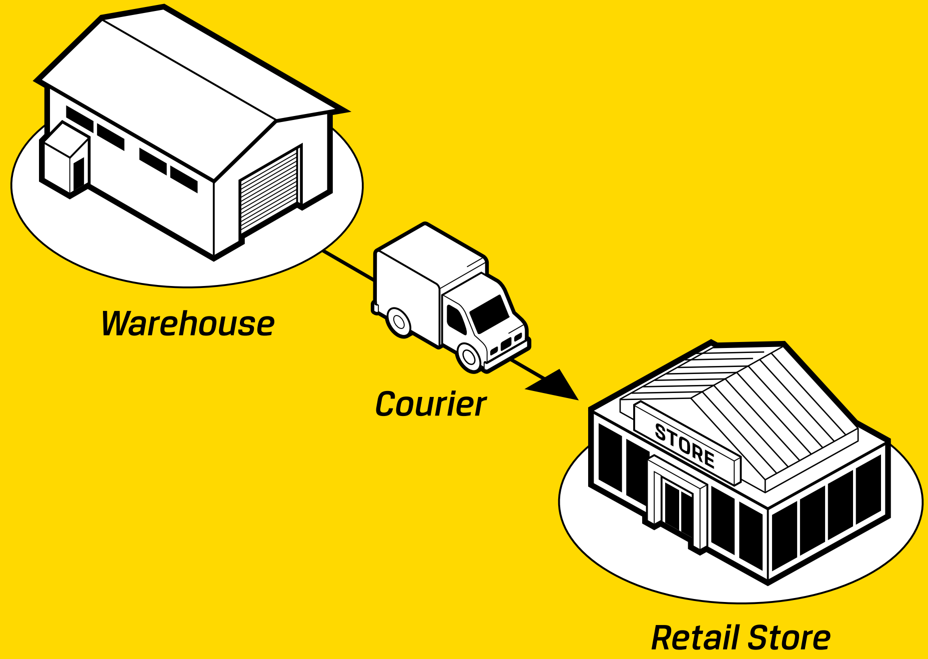
3 arrangements



Spatial arrangements say a lot about how a system works. A series of nodes and links arranged in a loop implies a recurring activity. A horizontal axes may imply some sort of order, such as hierarchy. Or a vertical axis may imply a time sequence. Common network arrangements (*topologies*) are recognizable as having a meaning in their own right. The patterns—lanes, trees, or hub-and-spoke radiation, as examples—imply system organization even if individual items have not been identified. Indeed, a few notations (such as *causal-loop diagrams*) are named after these arrangements. Yet **syntactics** does not simply refer to recognizable spatial patterns but also meaningful juxtapositions of items. Placing two items next to each other, or overlaying one atop another, can convey a limited set of meanings that is (more or less) easy to infer.

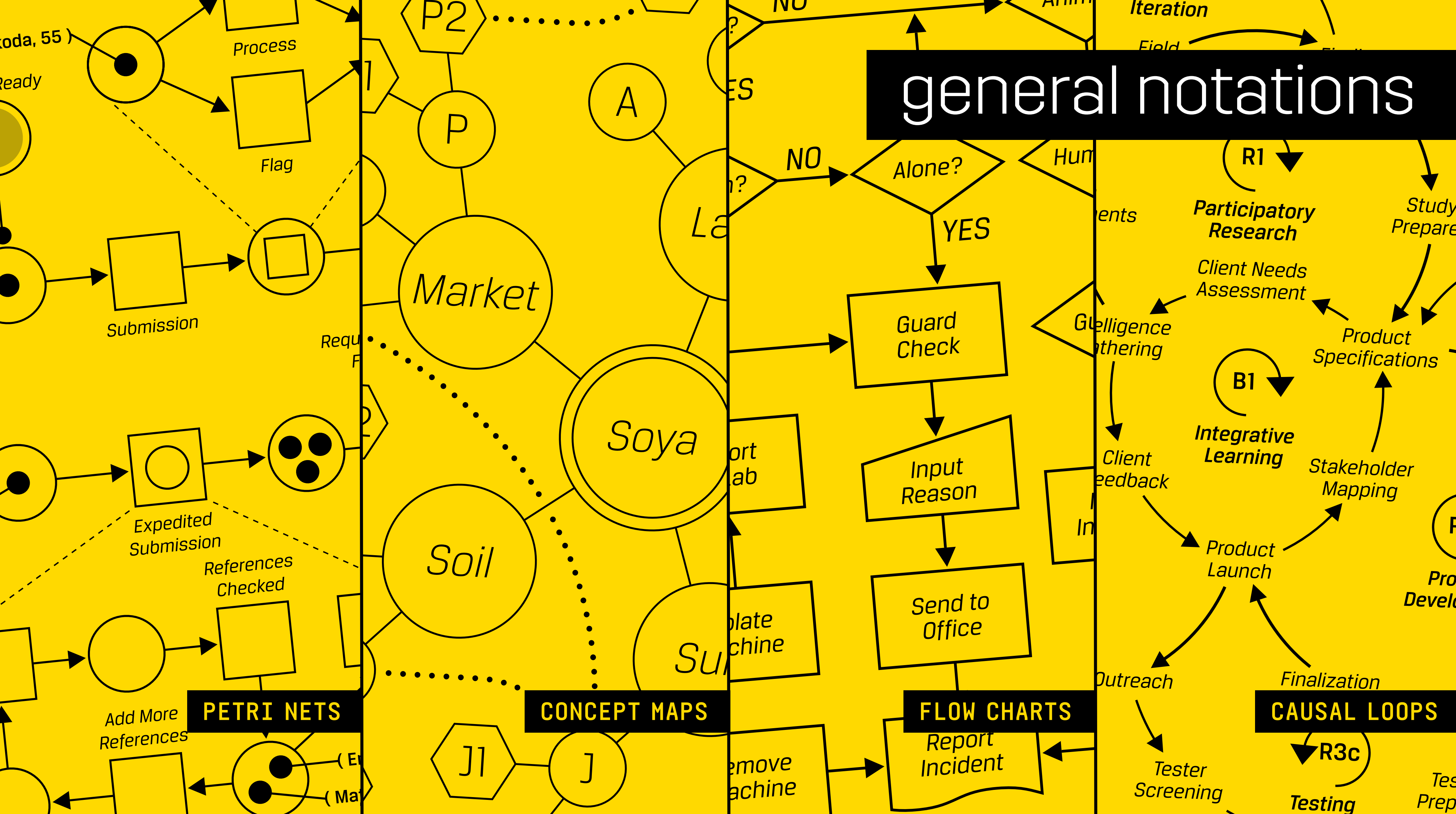
3 arrangements





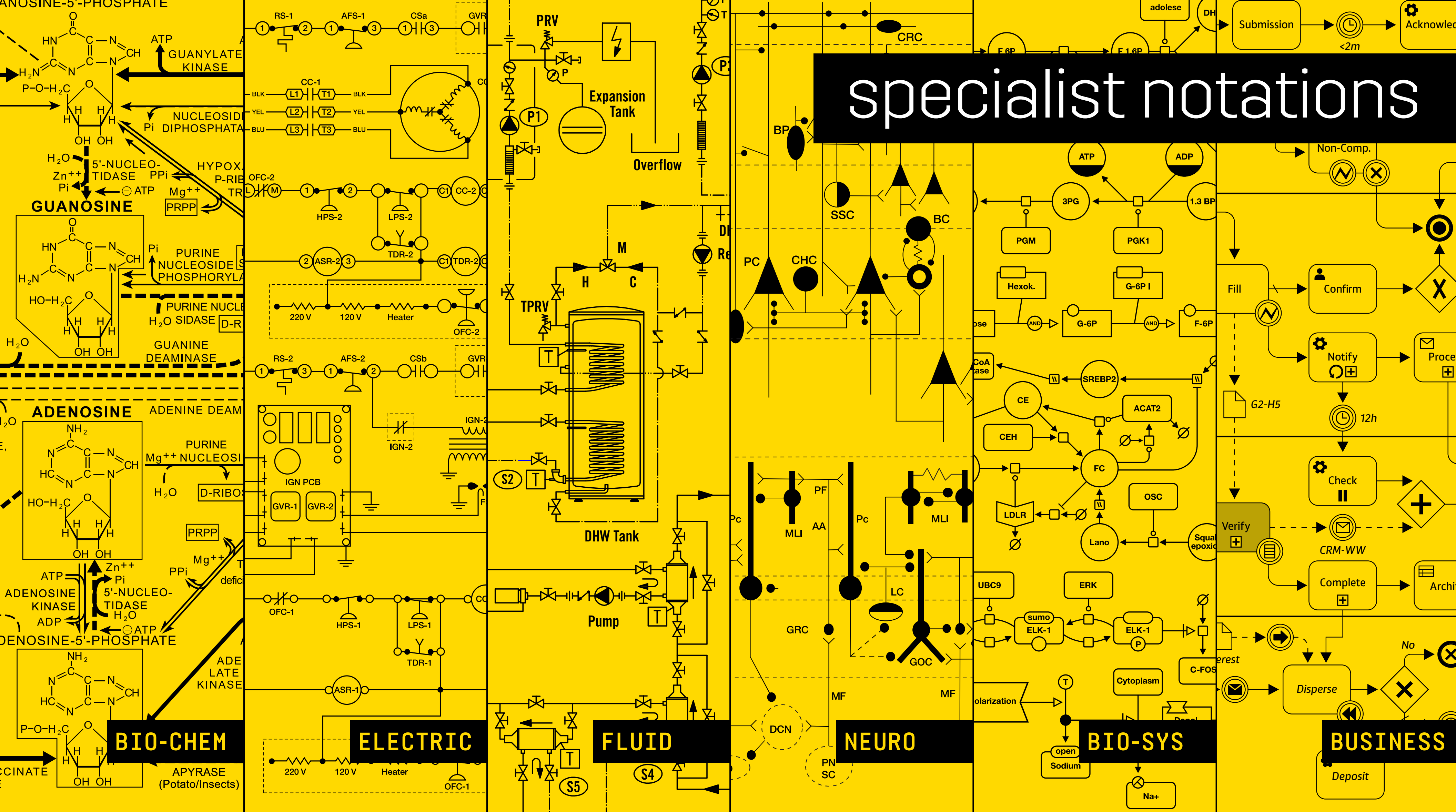
Consider two examples. One node may depict a person in clothing typically associated with farming. Overlaying an illustration of an "almond" further qualifies the farmer's role. Likewise with a beekeeper and bee. The juxtaposition of these two roles suggests a relation, which can be made explicit with arrowed lines. Bees provide pollination services to an almond grove, while pollen provides a food source for the bees: a mutually beneficial relationship anchoring part of a system. The simple spatial arrangement tells much of the story. Similarly, a simple chain arrangement shows a warehouse, courier truck, and retail store. The spatially arranged sequence (reinforced by an arrowed line) implies a meaning: the delivery of goods from storage to store shelf. Change the direction of the truck or move it to another position, suddenly the implied story changes.

general notations



Many **general notations** have emerge from the specification of these syntactical dimensions. For example, flow charts encode nodes with meaning using shapes: rectangle for process, diamond for decision, and so forth. Labels are placed inside the nodes to indicate specifics. Arrowed lines emanating from a diamond suggest "if … then" relations with other nodes. Labels indicating the "ifs" (such as "Yes," "No," "Maybe") are place next to each line. The overall arrangements indicates a cascading sequence of activities, with occasional branching and merging points along the way. General notations are generic enough to apply across a variety of different system types. That flexibility is a draw-back if greater specificity is required to communicate how a system works. Thus, a larger number of notations have emerged with more elaborate, subject-specific syntaxes.

specialist notations

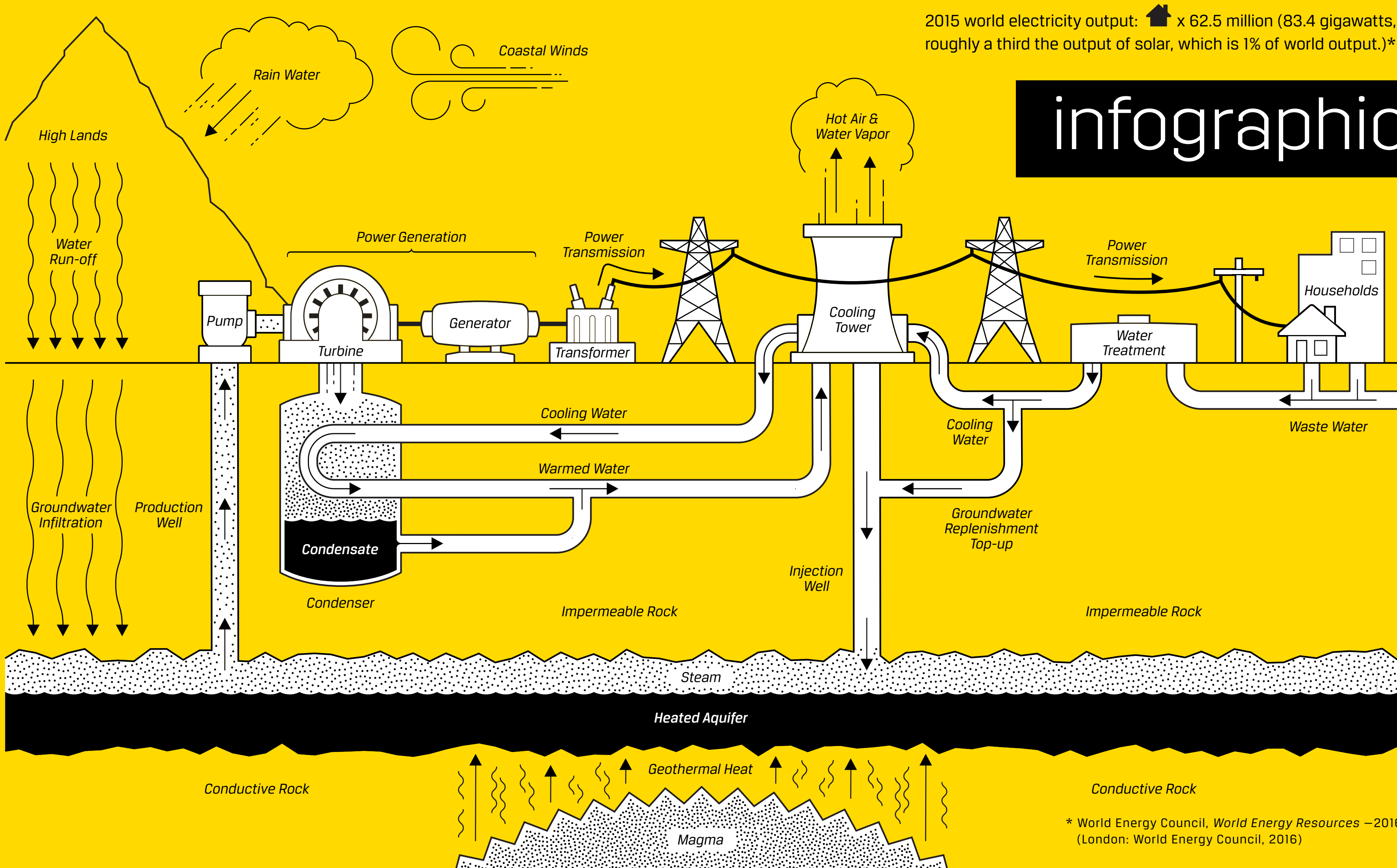


Specialist notations encode discipline-specific technical distinctions with symbols, acronyms, and other shorthands. Sometimes multiple notations compete. For example, *Systems Biology Graphical Notation* (SBGN) covers similar ground as *modified Edinburgh Pathway Notation* (mEPN) and *Molecular Biology Wire Diagram* (MBWD) notation. Each differs in their graphical philosophy, with SBGN being more visually oriented and mEPN being more textual. SBGN also emphasizes system dynamics, whereas MBWD stresses structural outcomes. Not that specialist notations are all that astute graphically; the vast majority are amateurish from a graphic- or information-design perspective. Few notations have been tested for efficacy. Even published versions are often drawn badly. Bad practices persist as "zombie conventions," perpetuated through uncritical mimicry.

silos

Professional and academic disciplines are **silos** that narrow attention. Specialist notations reinforce boundaries. For example, *Electrical Engineering Notation* does not readily integrate with *Energese* notation, describing energy flows more generally. Carving up of reality into thin slices for narrow purposes runs counter to the spirit of systems thinking. Systems are inherently open insofar as no single system operates in isolation. There are always outside dependencies and influences. Systems have side-effects on other systems. Interacting systems create new vulnerabilities. The unpredictability of our complex world has a lot to do with an inability to untangle the knotted relationships between systems. For those using specialist notations, outside influences simply get discounted as "exogenous," or beyond the scope of analysis. That is an entrenched blind-spot.

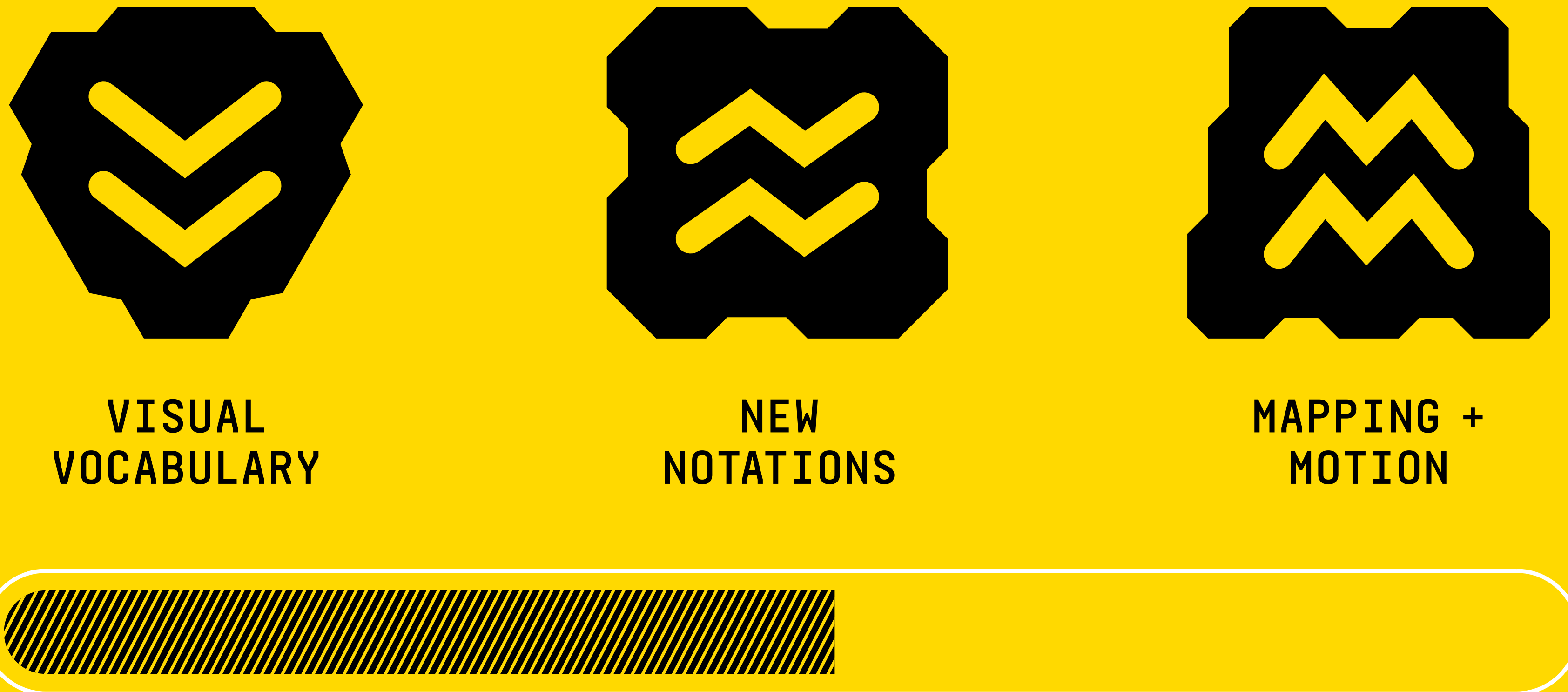
infographics



There are several well-established ways of overcoming these siloes and other limitations of notations. Information graphics, such as the ones published in science magazines, show systems pictorially and selectively using a variety of graphical devices. Giga maps and synthesis maps are assemblages of various forms of imagery (diagrams, graphs, charts, illustrations, and so forth) to describe systems.* These **bricolage** methods often incorporate notations for spot diagrams or use graphical devices popularized by notations. However, most of the visual invention is coming from these integrative alternatives. Perhaps there are lessons that can be applied to notations.

* See Peter Jones and Jeremy Bowes, "Rendering Systems Visible for Design: Synthesis Maps as Constructivist Design Narratives," *She Ji: The Journal of Design, Economics, and Innovation*, vol. 3, no 3 (2017), pp. 229-248; Birgir Sevaldson, *Designing Complexity: The Methodology and Practice of Systems Oriented Design* (Champaign, IL: Common Ground Research Networks, 2022).

project roadmap



The SystemViz Project began 12 years ago, aimed at improving how systems are visualized. There are three broad tranches of work. The first tranche is a stock-taking of the systems literature to figure out what topics could benefit from visualization, with a watchful eye for neglected topics. The intention is to build a nuanced language for talking about systems across disciplinary silos. Next came a stock-taking of the notations for visualizing systems, the codex I am presenting today. What works? What doesn't? What's missing? What's over-emphasized? Hopefully, answers to these questions will inform the creation of a new notation that is tidier, more comprehensive, and more intuitive to a general audience. Finally, such a notation should work using new media, especially those involving motion and interaction. Let's review some the key pieces created on this **roadmap**.

visual vocabulary

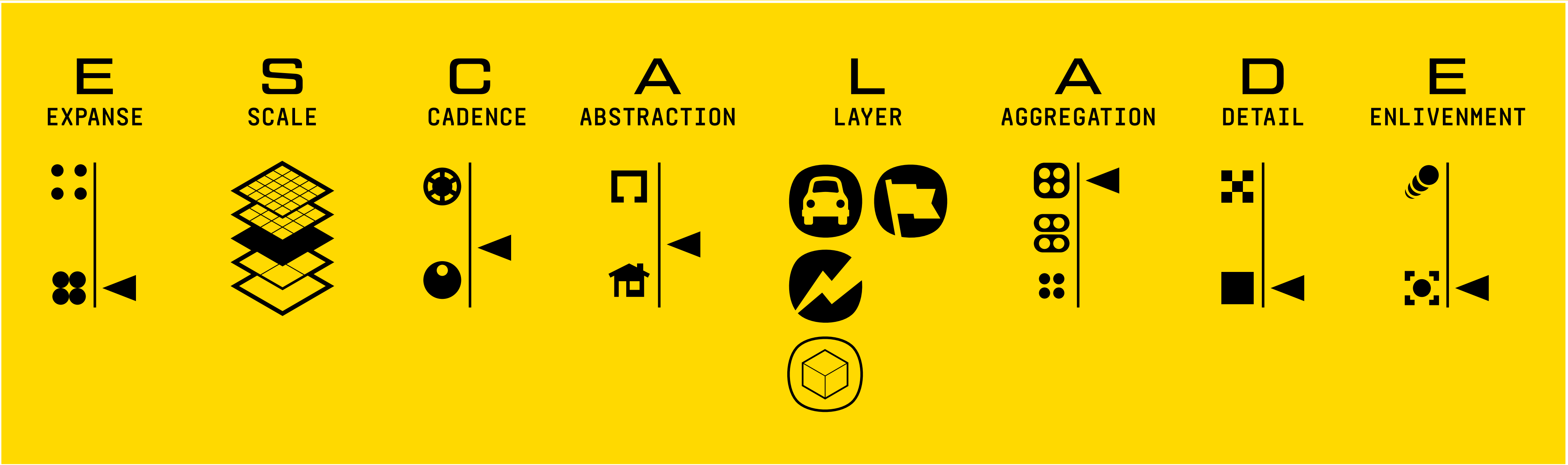
The *Visual Vocabulary of Systems* itemizes the elements and dynamics of systems. These were drawn from systems literatures found within the natural sciences, social sciences, managerial disciplines, and design disciplines. Elements and dynamics are arranged into six marquee categories which any system will have by definition: drivers, signals, states, boundaries, relations, and domains. They are then assigned an illustrative icon according to a unified visual syntax, one with notational qualities.* When the *Visual Vocabulary* is compared to the *Anatomy of System Notations*, a strong bias is seen among system notations in favour of drivers, signals, and relations over the other system subjects.

* Peter Stoyko, "Icon Design: Visual Vocabulary Syntax Specifications," SystemViz Project, Version 1.0.1, January 15, 2023. [systemviz.com]

At last year's RSD conference, I presented the *Pattern Atlas of System Vulnerabilities*. This codex plays with the theme of entanglement as its organizing metaphor. Some 30 forms of vulnerability in human-made systems are itemized across four levels of analysis. Each vulnerability is assigned an icon that analogizes using tangled (or otherwise compromised) threads. The set shows that many of the vulnerabilities emerge through the interaction of systems, sub-systems, or sub-routines. This is relevant because any notation should be able to show weak-nesses, failure states, and vulnerabilities, not just show systems in an idealized working order. Human-made systems are never flawless. Notations are expected to help diagnose risks and shortcomings. Systems cannot be made "bullet-proof" or "unbreakable" but they can fail minimally and recover elegantly.

design framework

DASHBOARD



What makes for a good system map, diagram, or notation? What evaluative criteria should be used to scrutinize the features? That is what the *ESCALADE* information-design framework is for. Most systems notations are cobbled together with little thought given to the overarching design principles. That is a major oversight considering that systems are a particularly challenging subject to visualize with fidelity. The ESCALADE framework helps visualizers manage the inherent tensions and trade-offs using eight criteria. Each criterion is actually a bundle of helpful considerations and design options. Taken as a whole, ESCALADE operates like a dashboard to be mindful of how a graphic handles summarization (or reductionism). Evaluation criteria is an important stepping stone towards designing a new notation. A lot of thought and care went into the framework.

project roadmap

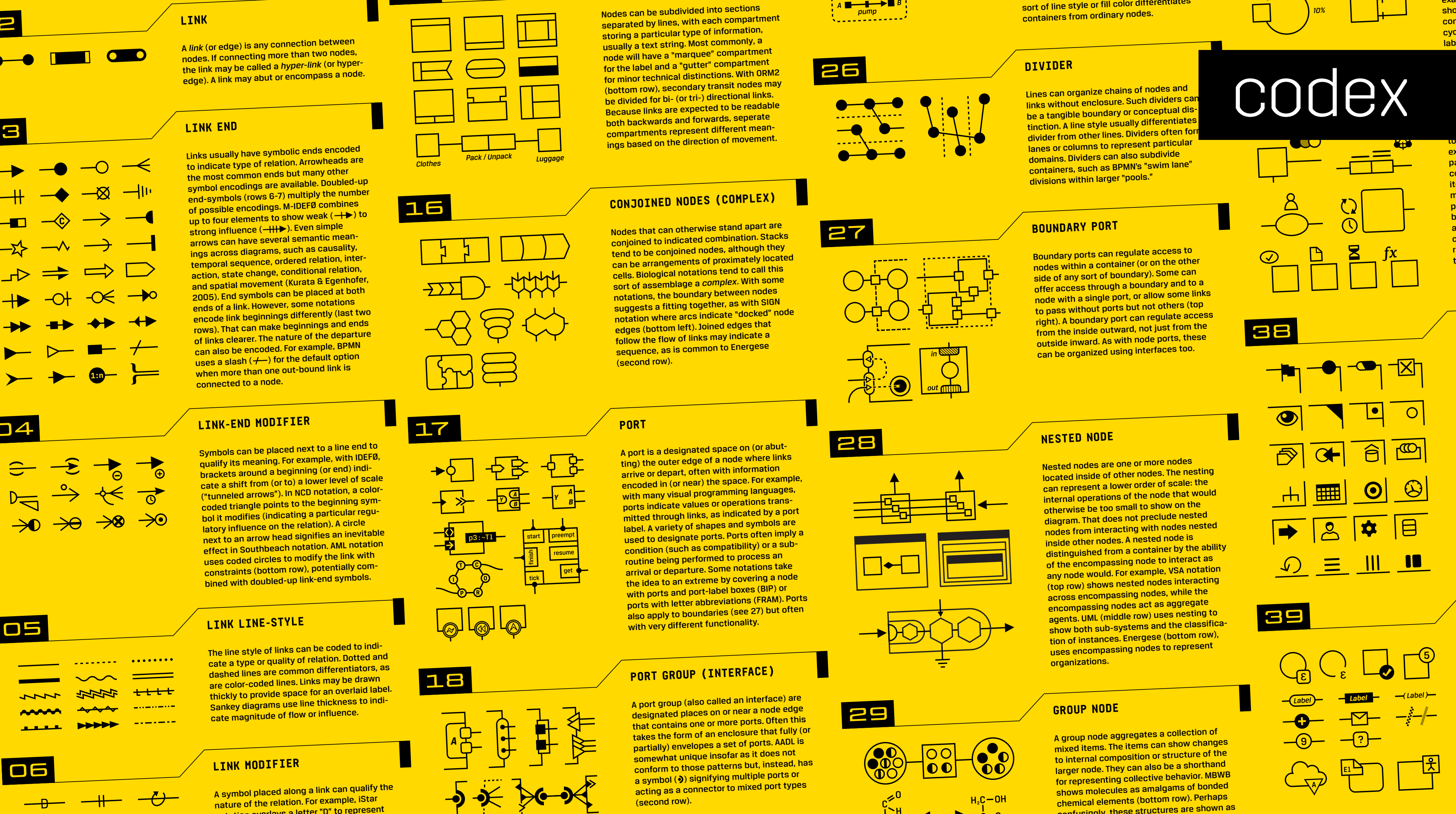
ELEMENTS + DYNAMICS

VULNERABILITIES

DESIGN FRAMEWORK

GRAPHICAL CONVENTIONS

Let's revisit the **roadmap**. A methodology is playing out. Each of these pieces represents progress towards a larger goal. First comes a compilation of the elements, dynamics, and vulnerabilities of systems. This is a survey of the territory, so to speak. Next comes a look at current mapping practices, in terms of both overall approach and prevailing techniques. This is how we judge the adequacy of our maps. The subsequent synthesis will undoubtedly reveal a lot of room for improvement. Hopefully, such lessons will be folded into a new notation that: (a.) can express the full range of system elements and dynamics (b.) using emerging forms of interactive media to (c.) make system analysis more accessible to everyone. That notation will be "opinionated" insofar as it manages trade-offs and tensions intelligently, rather than trying to be all things to all people.



So that is where the new **codex** fits into the grand scheme of things. The first incarnation of the codex is a poster that serves as a reference. Some 65 different graphical devices are listed, each with multiple examples. Citations are provided for those who wish to delve deeper into particular notations. If nothing else, the codex provides a list of graphical options that systems thinkers can draw from to develop their own maps and diagrams.

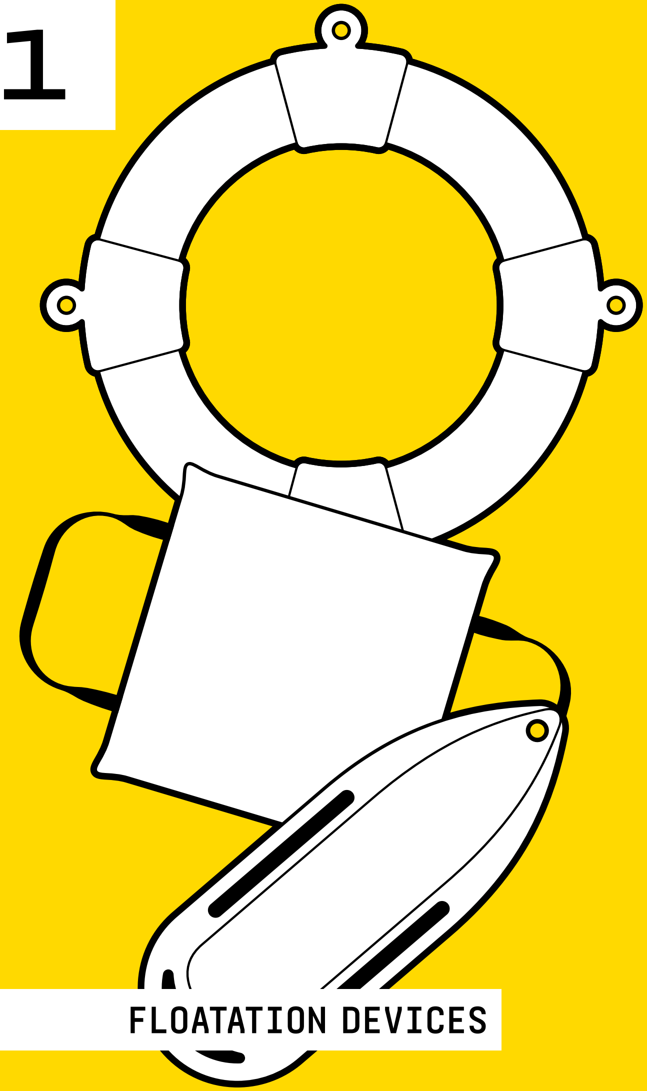
Indeed, all the pieces necessary to develop a new notation are there in one place so as to stimulate dialogue about the art of the possible. Although the codex is currently in the form of a poster, the hope is to create an online database so that all these graphical devices are organized in a way that is more extensible, flexible, and easier to cross-reference. Any suggestions along those lines are welcome.

mnemonic

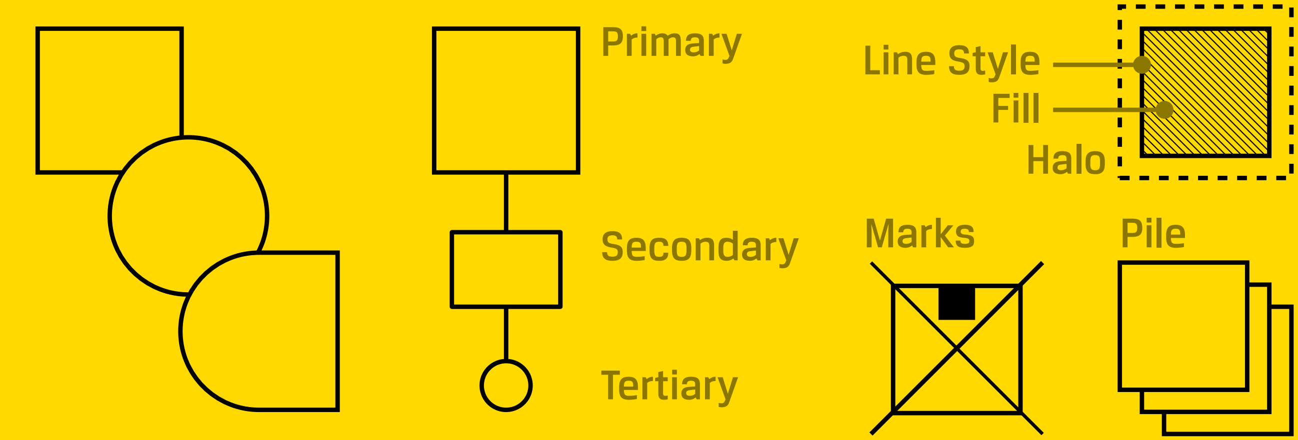
What is a good way to go through the substance of the codex? A rote itemization would not make for an exciting presentation nor is there enough time allotted for that. Instead, I thought it would be fun to group the various devices according to a playful metaphor: the sailing journey. The hope is that, by comparing groups of graphical devices to a set of familiar objects, they would become **easier to remember**. Not everything in the codex fits the analogy—much has been left out. Nevertheless, the main topics are covered in the seven general categories that follow. It is also hoped that some of the terms used will gain currency given the lack of a unified vocabulary for talking about features across notations. Indeed, care was taken in naming the different types of graphical device in the codex based on established usage or appropriateness given the range of applications.

1

node forms



FLOATATION DEVICES

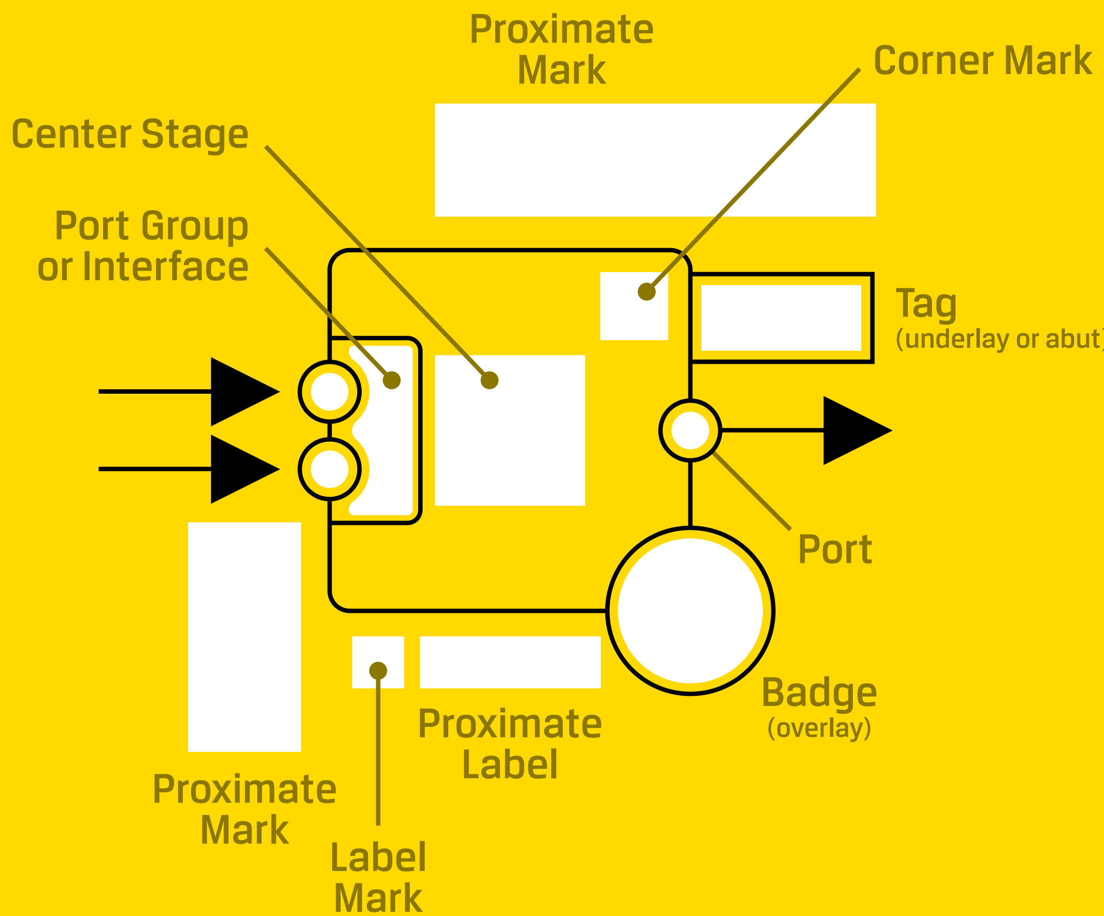


Consider the varying sizes and shapes of flotation devices found on sailing ships: ringed life preserver, floating seat cushion, torpedo-shaped rescue buoy, and so forth. Nodes come in different shapes, sizes, and styles too. These **node forms** are usually encoded with particular meanings. As mentioned, flowcharts use a large library of shapes: rectangle stands for "process step," rounded rectangle for "event," rectangle with one rounded side for "delay," and so on. Fills, lines styles, and various marks can be added to those shapes to multiply the encodings. Worth noting is the hierarchy of nodes, with nodes of different size and shape used for different classes of node. For example, a petri net usually has large round nodes for stations in a process and small rectangular "transit" nodes for transformations. That differentiation allows the eye to more easily parse a crowded diagram.

2

node complications

WATCH-FACE COMPLICATIONS

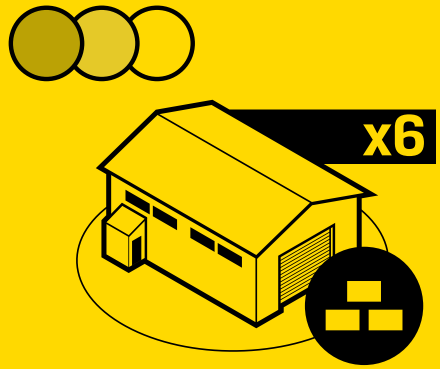


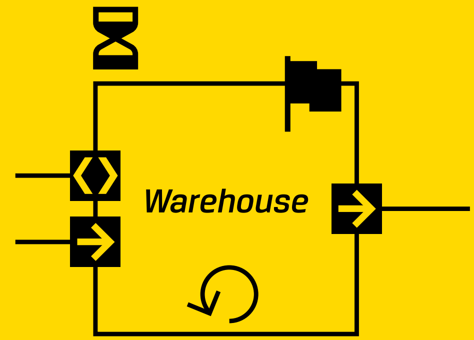
Consider the interface elements on an nautical watch face (a chronograph) that do not tell the basic time. Over the last century, watch designers have found ways of adding glanceable supplementary information to a highly constrained space using these *complications*. Notation creators have done likewise by attaching qualifiers to nodes, albeit without the same artisanal care. These **node complications** can modify the meaning of a node or attach information. The places where links "dock" with the node—*ports* and *interfaces*—can be given meanings, very useful if the node represents a complicated subsystem with functional Input/Output (IO) channels (as software modules, machines, and living cells do). Not all potential complications are used in a single notation, which would over-stuff the node past the point of distraction. These are simply available options.

2

node complications

WATCH-FACE COMPLICATIONS

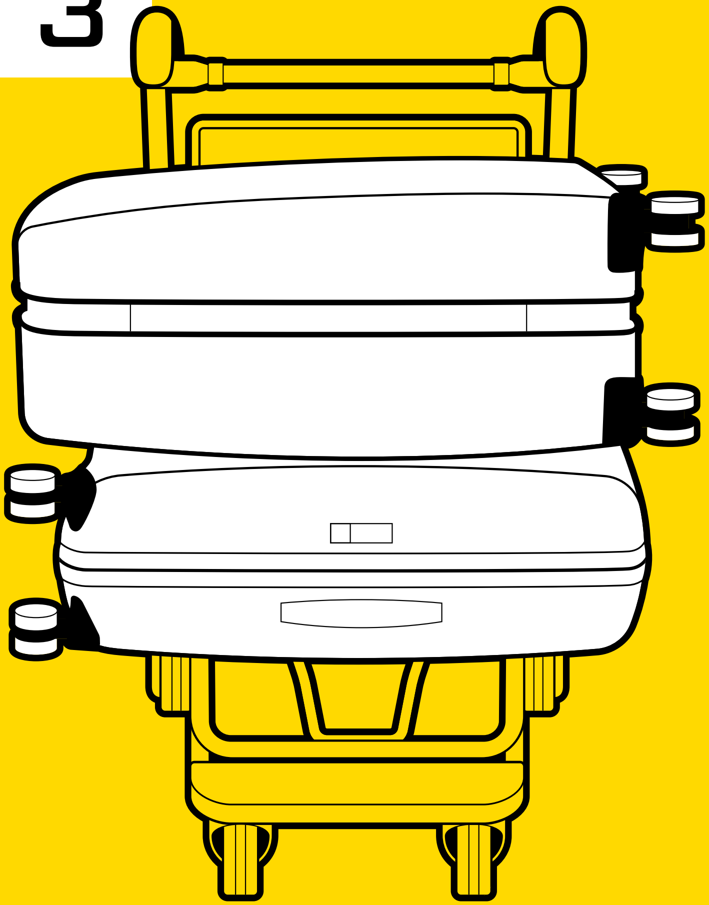




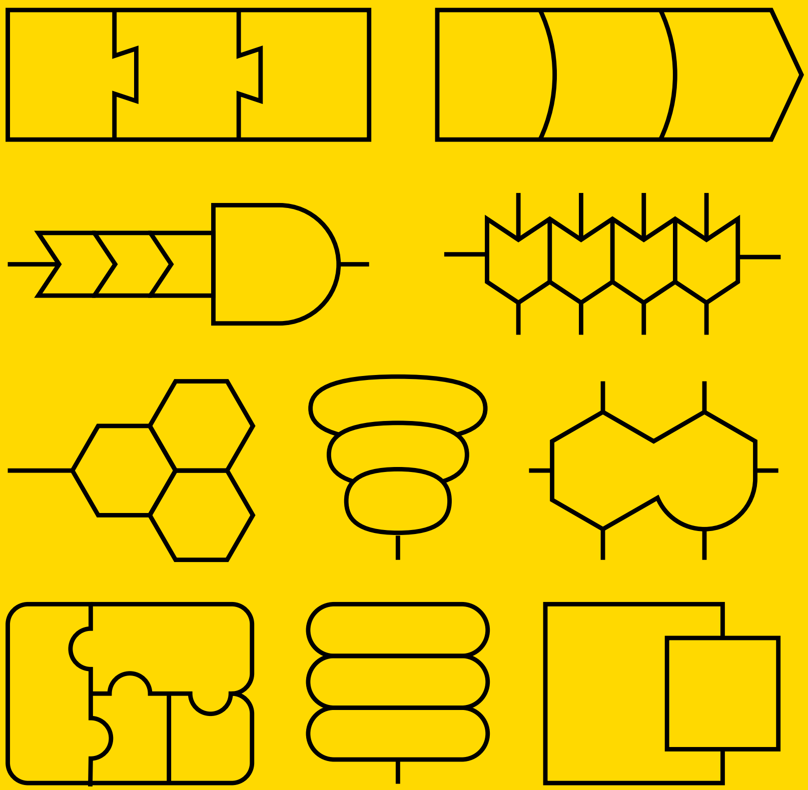
Let's look at some **examples** to show how complications tend to work in actual diagrams. If a node is a spot illustration (such as an isometric drawing of a warehouse), a badge could indicate what kind of warehouse it is (such as raw materials supplier). A tag could indicate how many warehouses there are in a supply chain. Color-coded circles could be used as proximate marks to represent roles taking part in warehouse-related tasks. In another example, node-margin marks can literally flag a node for special attention, or indicate something about its underlying processes. The centre stage of the node can be put to various uses, displaying a nested label, a symbol/label combination, or an inset graph. That latter device can qualify the nature of the interactions that the node takes part in if those interactions are not straight-forwardly linear but follow a complicated function.

3

node complexes



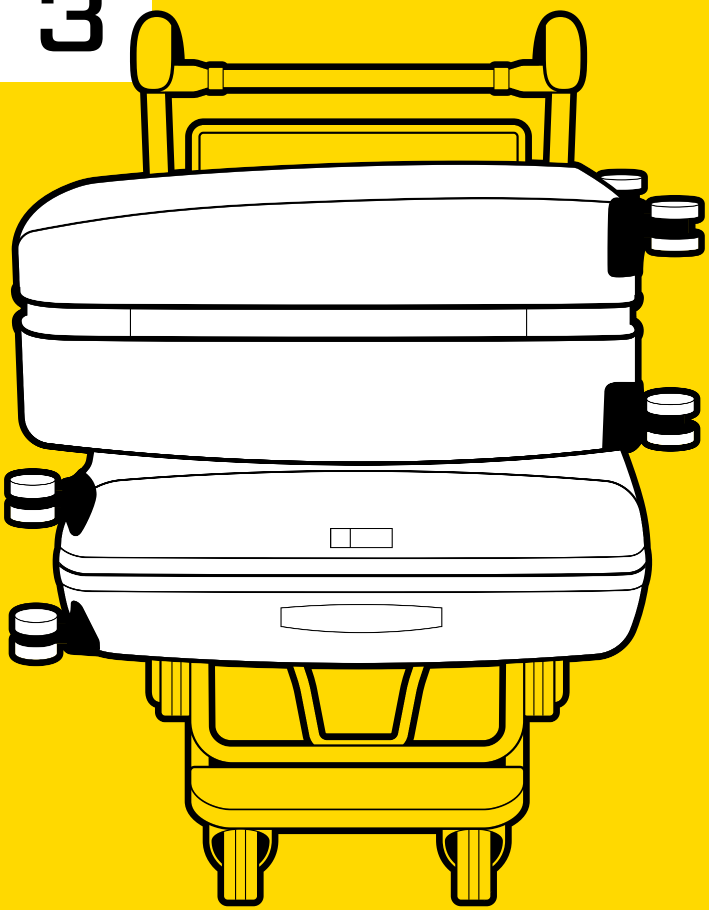
LUGGAGE TROLLEY



Next comes the luggage trolley as an analogy for nodes that are clustered together, much like a heap of suitcases. Those clusters are called **node complexes** within neuro- and microbiological notations, which seems apt for notations in general. Nodes take on the character of stacks, as different subsystems or components arranged together to loosely suggest internal dynamics. The way this is done varies wildly across notations. It is common to show hierarchies, sequences, and modular configurations. Directionality can be indicated with pointed or curved sides. Super- and sub-ordinate modules can be ordered horizontally. Puzzle-like tabs and slots can show points of interaction between modules. The larger point is that a node does not represent a simple actor or process, but a multi-part ensemble or routine respectively.

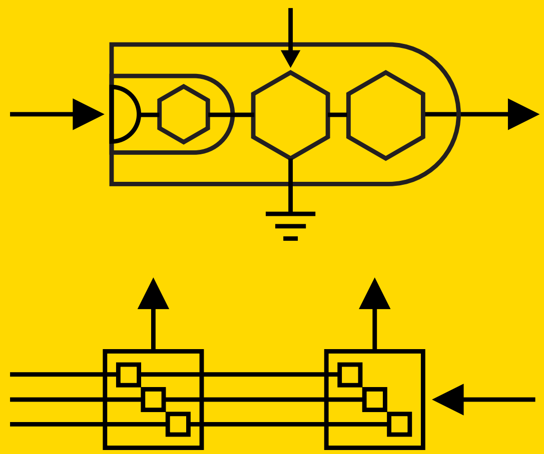
3

node complexes

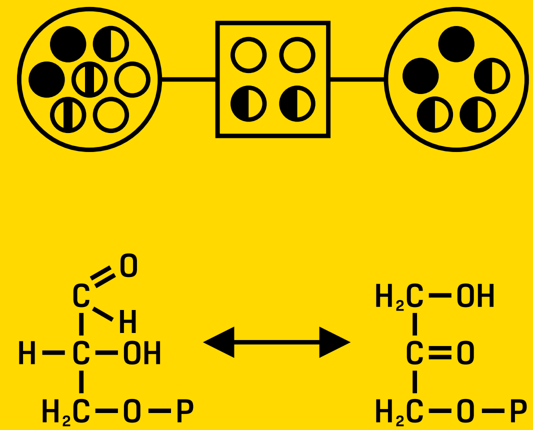


LUGGAGE TROLLEY

Nested

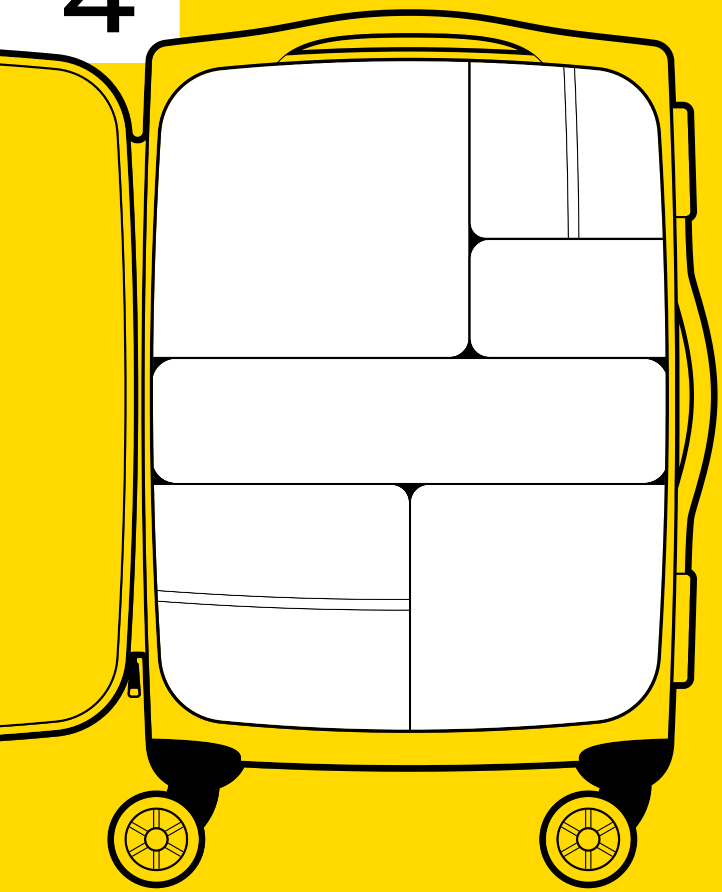


Group

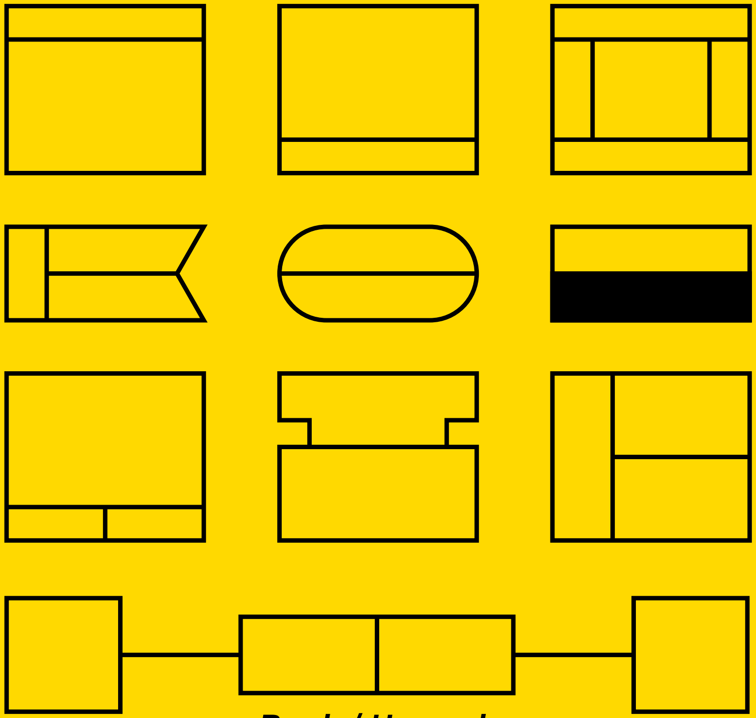


Node complexes can be more complicated than agglomerations of modules. Nodes can be nested within other nodes to suggest interaction at different levels of scale. For example, small scale nodes interact with each other through links, while larger containment nodes also interact with each other, suggesting aggregate effects. The Energese notation uses nested nodes to show organization (top example), with a containment node representing some sort of enterprise and internal nodes showing a sequential process. Nodes can also group internal items that show states. For example, nodes can show different compositions of items. With *Molecular Biology Wire Diagram* notation, chemical structures represent molecules which are then linked together. That increases the potential for confusion, given that both forms of organization are network diagrams.

4 node segmentation

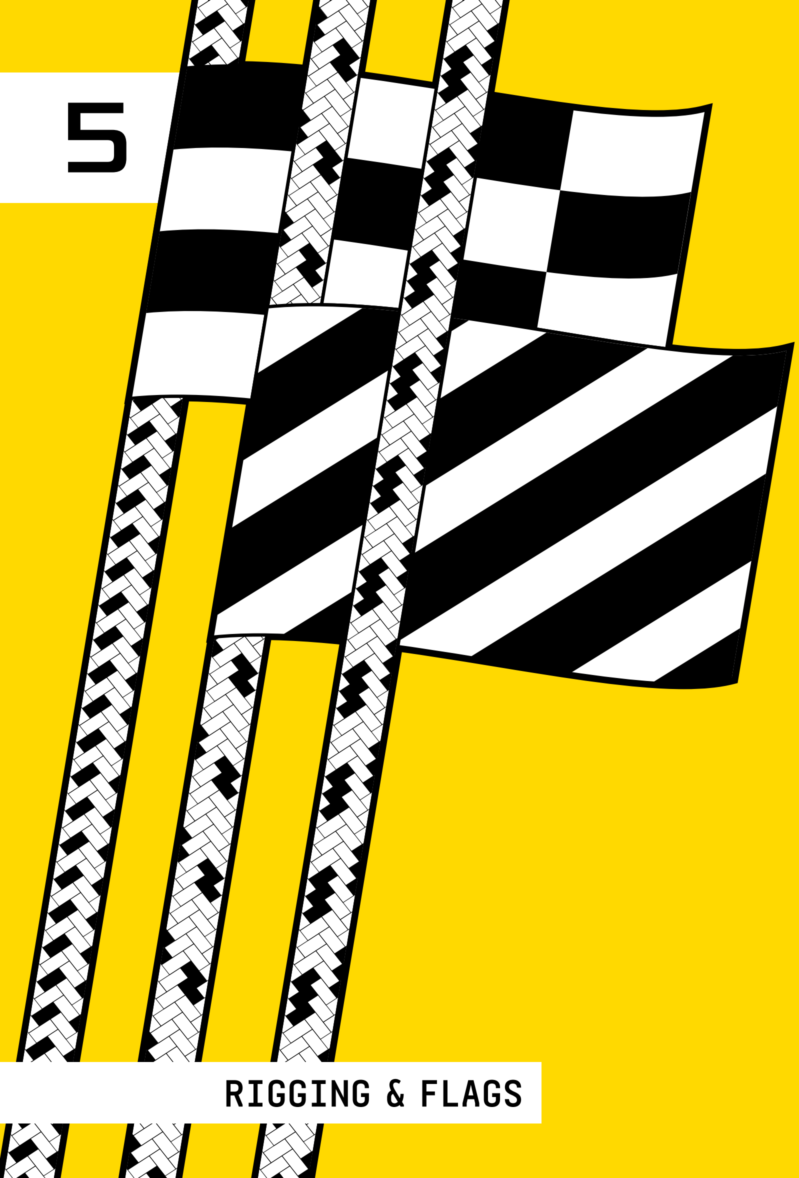


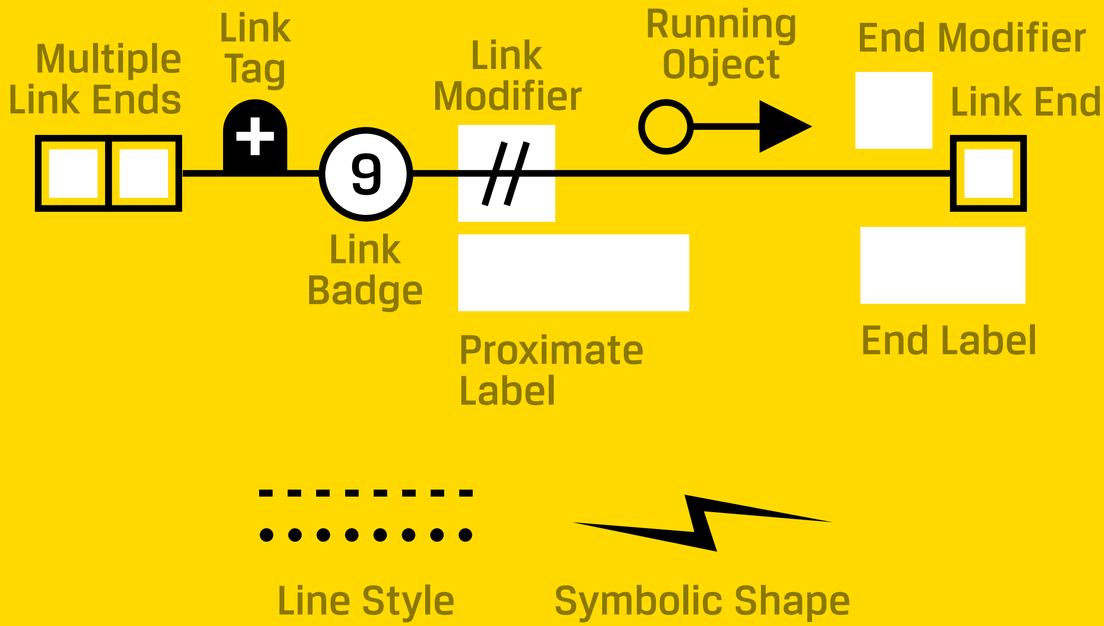
MODULAR LUGGAGE



Not only can nodes be merged together, they can be subdivided too. That does not imply different entities. Instead, the nodes are partitioned to show different types of information within designated spaces. That makes it easier for a viewer to parse a graphic and pick out the bits of information that are more relevant. Most commonly, a top bar is a space designated for placing a label. Conversely, bottom bar may represent a gutter for less important information. Nodes full of text often feature lists which can have their own designated space. Of particular interest is *Object-Role Modelling* notation, which subdivides transit nodes into two or three sections with labels assigned to each subsection. That allows the relation to be read forward or backward. For example, clothes that are packed become luggage, whereas luggage that is unpacked becomes clothes.

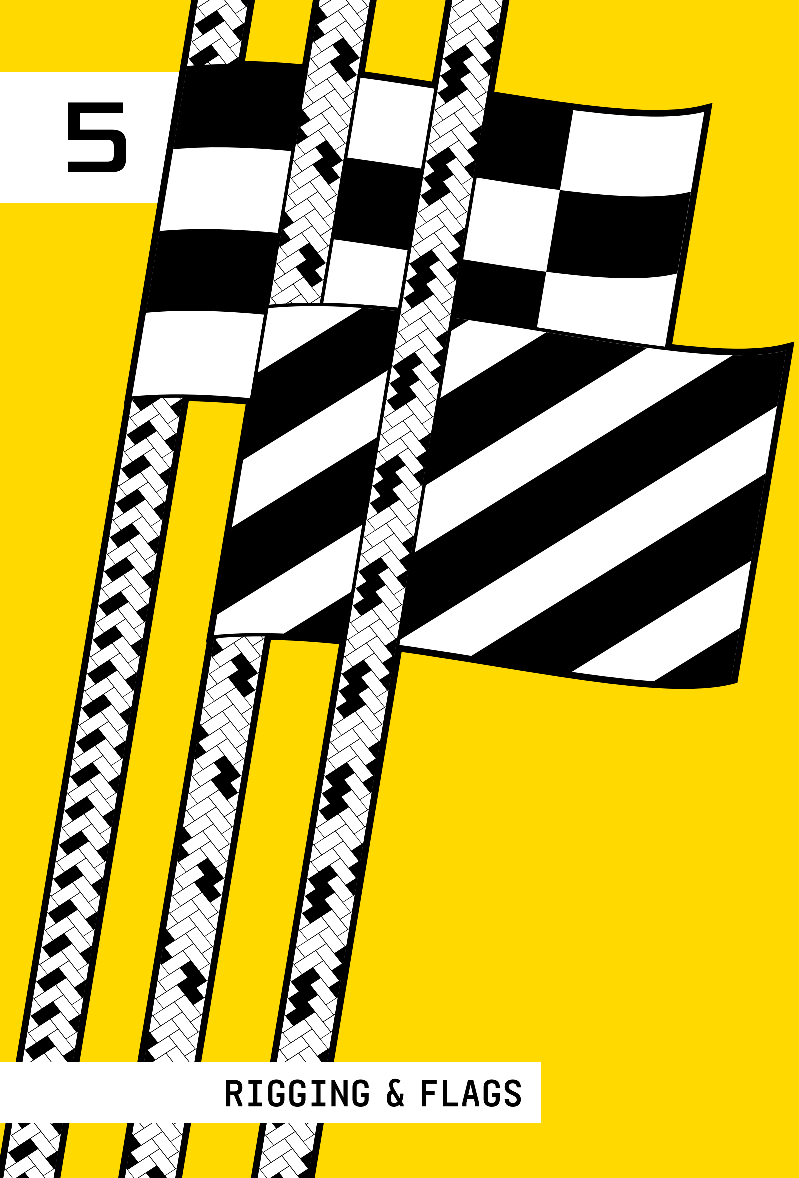
5 link complications



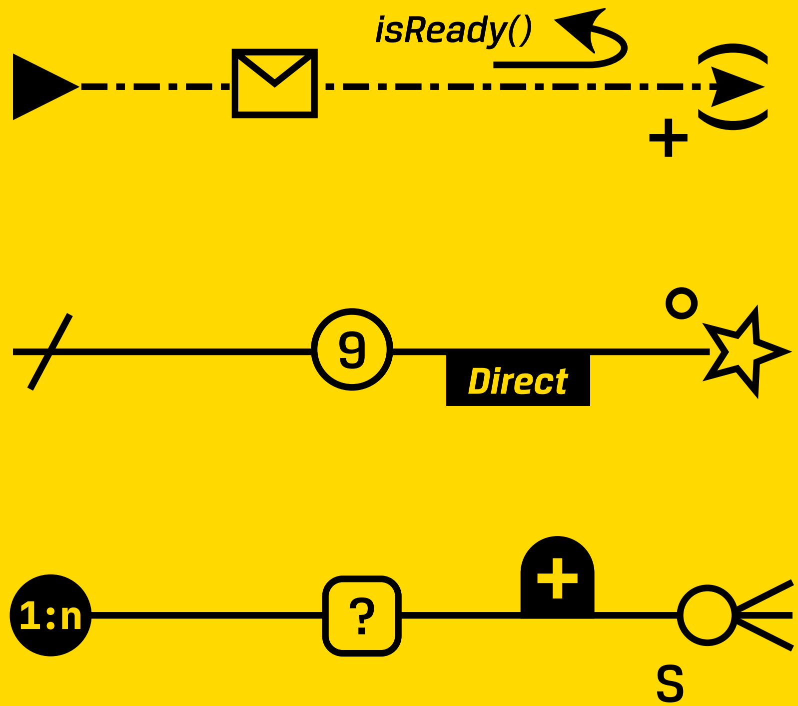


Much like nodes, links can have qualifying information in the form of complications too. Think of this as analogous to the way sailing ship ropes have different patterning, colored ties, and symbolic flags. A link can have an encoded end-point, such as an arrow head or circle. These can be encoded using fills, colors, shapes, or the like. Indeed, end encodings can be doubled- or tripled up. And these ends can be further modified with a nearby symbol. Link line-styles can show different types of relations. The shape of the line itself can have a symbolic meaning. Tags, badges, and intersecting marks can further modify the link. Labels can be placed along the line or near end-points. Moreover, objects can be shown running along the line, such as arrows or items attached to arrows. With animated diagrams, the running items would usually move along links in an overlapping way.

5 link complications



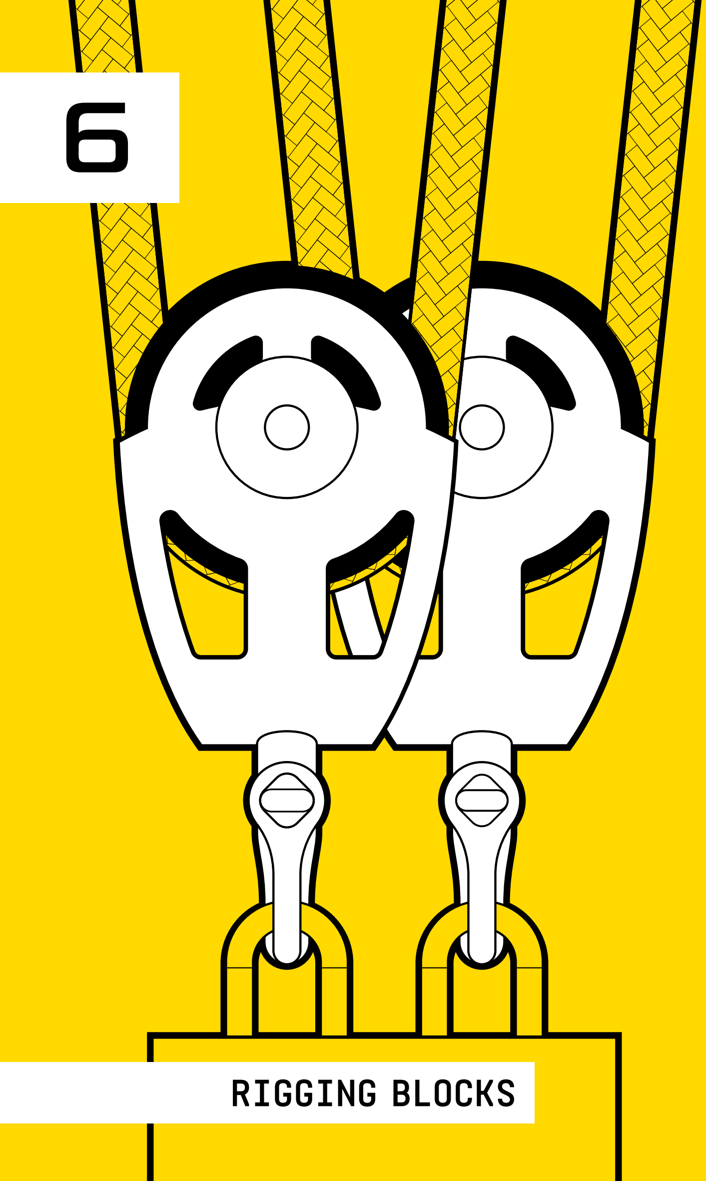
RIGGING & FLAGS

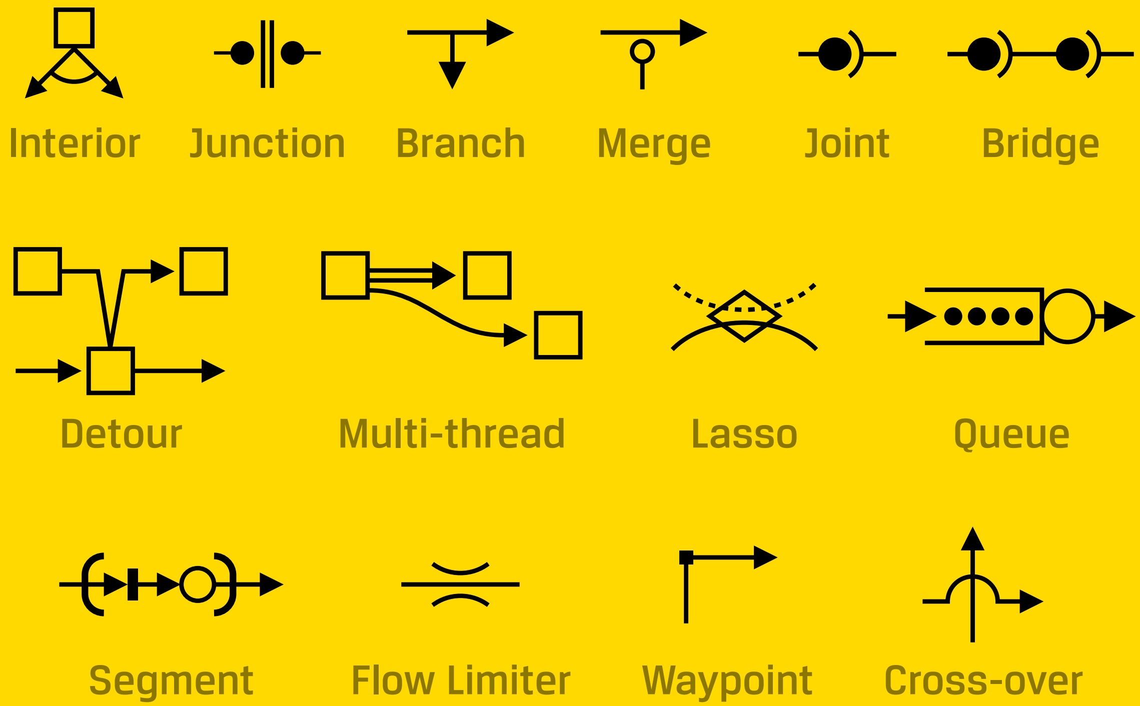


The variety of link complications is enormous. Consider some particularly interesting examples. Beginning-points can be coded differently than end-points. *Business Process Modelling Notation* adds a slanted mark at the beginning of a link to represent "default path" if a node has more than one out-bound link. Simple link modifiers (plus- and minus-signs) can indicate positive- or negative valence. IDEFø notation modifies end-point arrows using ordinary brackets to indicate that a link connects to a node at a lower level of scale. Within dense maps, it can be hard to associate a free-floating label with a particular link. Attaching tags and badges directly to the link prevents attribution errors and keeps the map tidy. Running arrows can be helpful if the link encodes direction but certain items are exceptional and can move in a different direction.

6

link paths

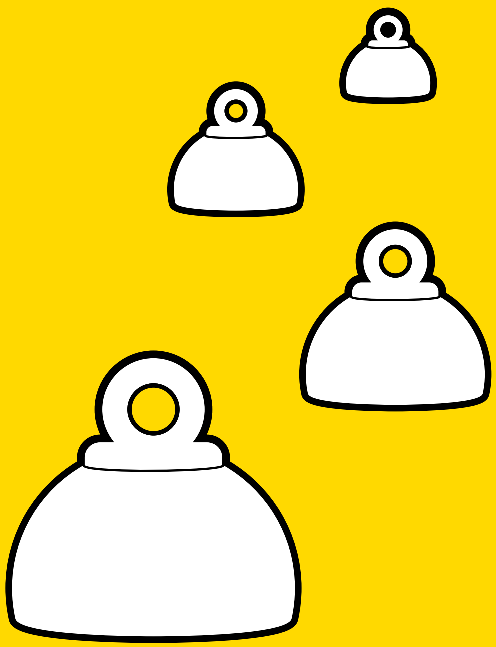


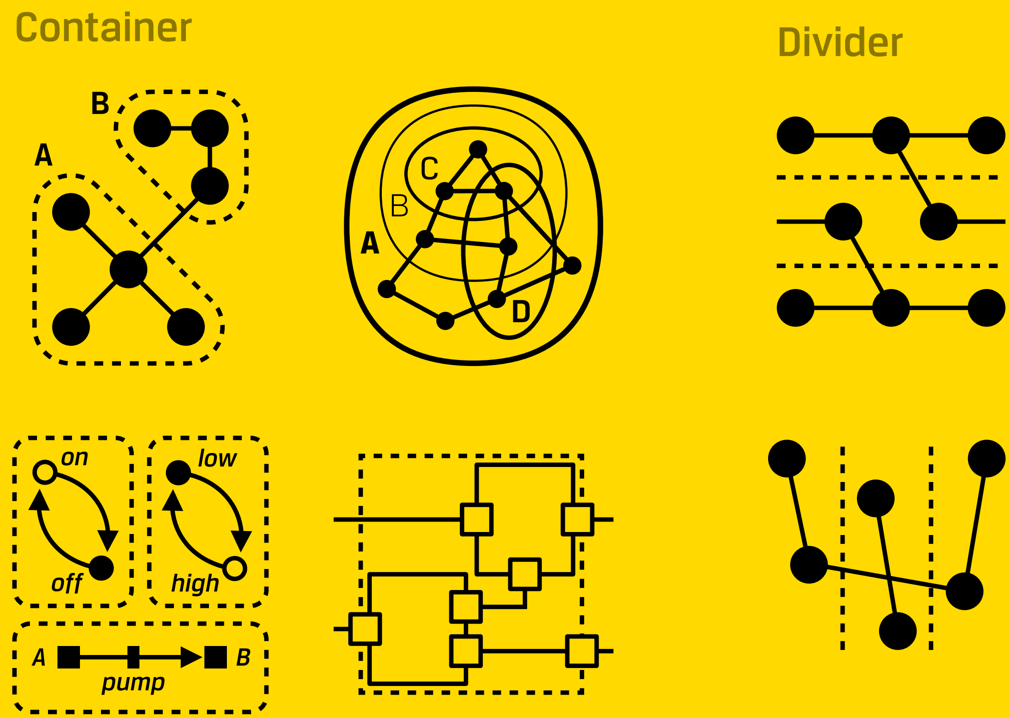


On a sailing ship, the rigging interconnects using rigging blocks, such as pulleys and fasteners. Likewise, there are a number of different ways that links connect together. Points where links join do not necessarily have to have nodes, yet some sort of encoding can indicate something about that connection. Sometimes links represent actual conduits that need to be joined but are not compatible, thus necessitating some sort of intermediating bridge. The interior angles of multiple out-bound links can be encoded, showing "either or," "both and," or "at least one" paths chosen. Detours can show side-effects or a link picking up some sort of taint. Queue symbols have both link and node qualities but, either way, show time delays. Flow limiters can encode rate information. Links can be broken down into segments in various ways to indicate stages, as one example.

7

contextual devices



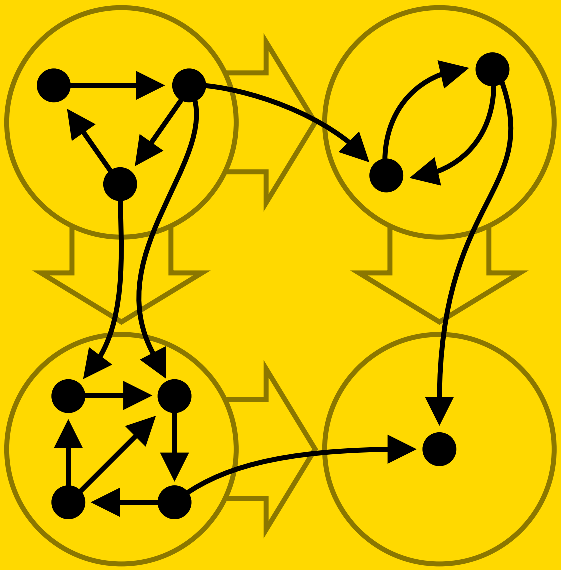


BOUNDARY MARKERS & BUOYS

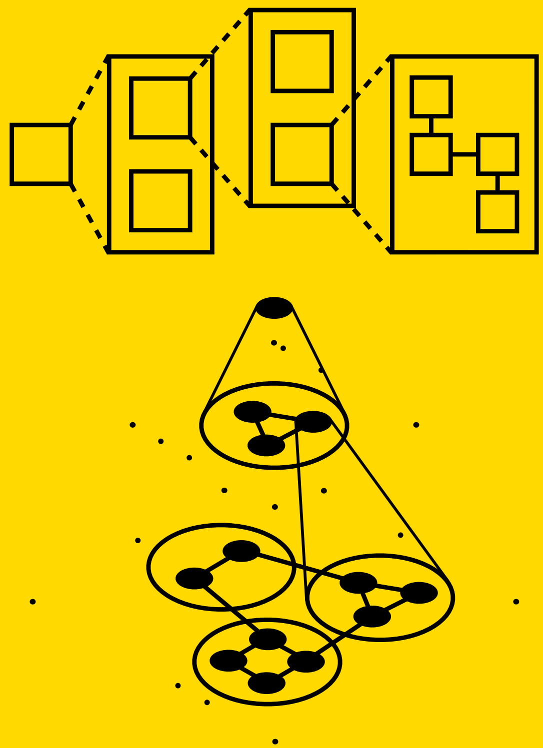
Finally, there are a number of graphical devices to indicate something about the domain or context a system operates in. Nodes can be grouped together using containers, which can then be encoded. Lines and background colors are the most common forms of container. Those containers can have ports, indicating boundary crossings. Open-ended boundaries (dividers) can also organize nodes and links, even if those boundaries are permeable. Of particular interest are containers that show subsystems that relate but are shown as discrete spaces. For example, a pump can be a transit node between two primary nodes. The operations of that pump can oscillate between "on" and "off" states, or "high" and "low" flow-states. This can be shown with proximate positioning. Or it the spaces can have nodes with the same labels.

contextual devices

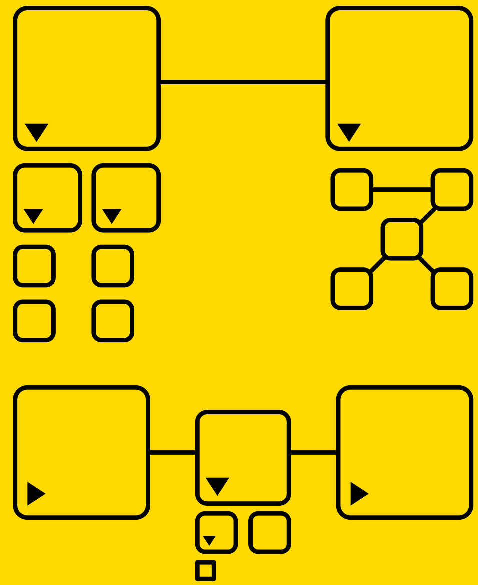
(Acyclic) Condensations



Decomposition Blocks

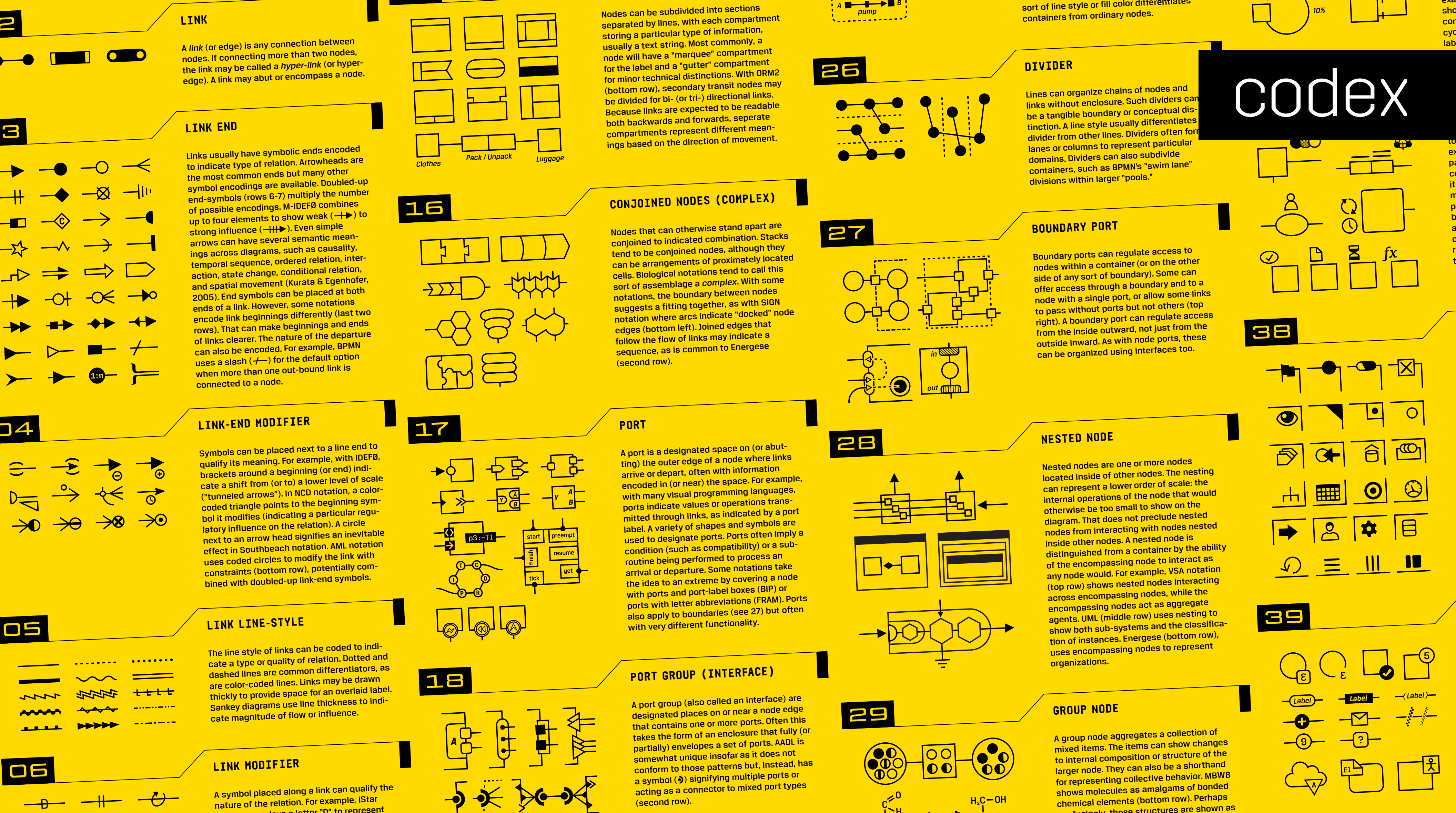


Disclosure Toggles

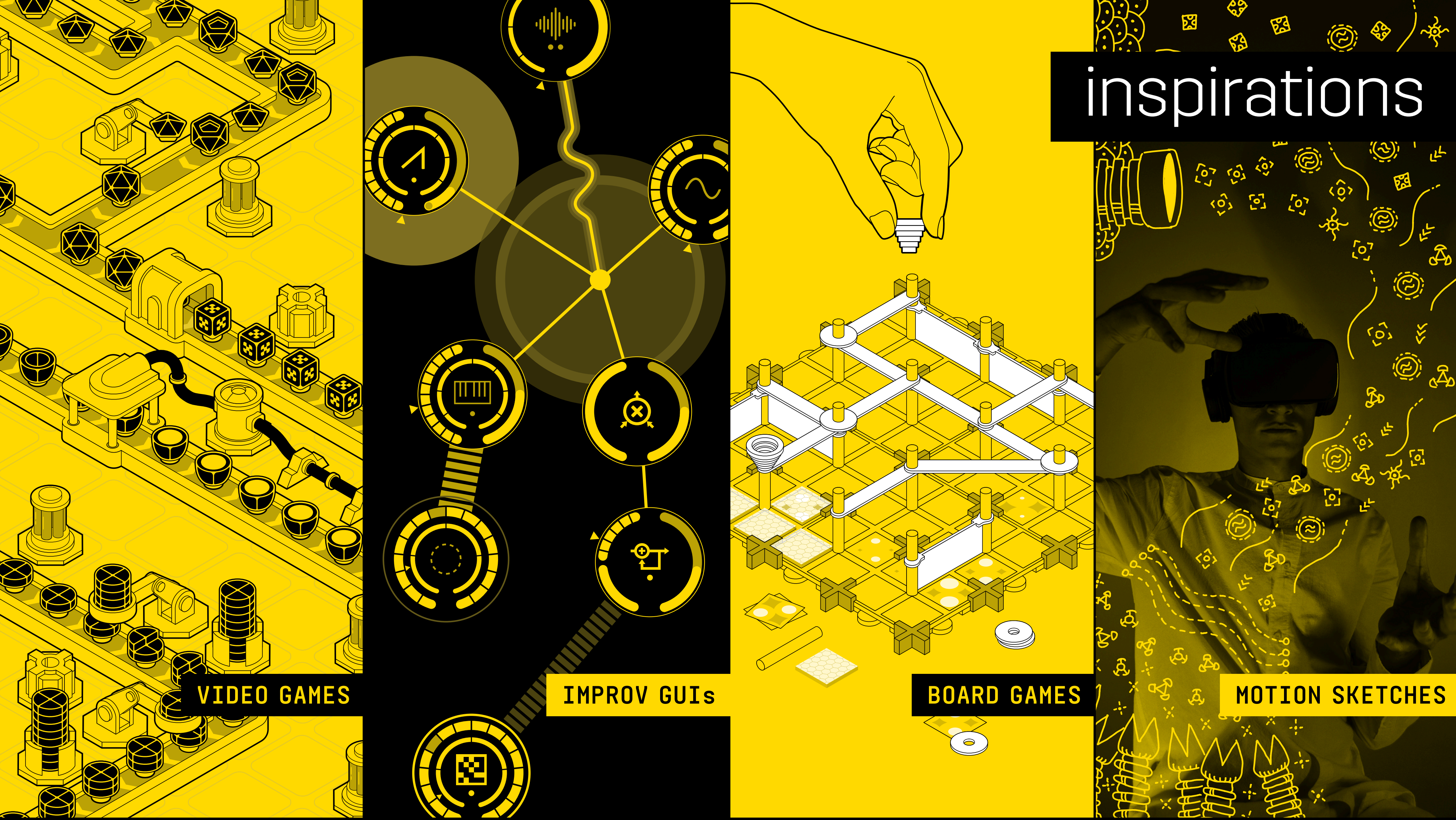


TREATMENTS OF SCALE

Of particular interest are contextual devices that show different levels of scale. Acyclic condensation can summarize circuits of links and nodes by superimposing meta-links and nodes. Another way of showing this is by having a simplified meta diagram appearing below the main diagram as a summary. Decomposition blocks show how wholes can be broken down into parts, which can in turn be broken down, and so on. This allows the diagram to "drill down" to lower levels of scale. The DSRP notation is unique in its use of disclosure toggles, which are more often associated with interactive computer interfaces. Clicking on a triangle pointing sideways causes it to point downward, at the same time revealing the internal contents of the larger node. That sort of graphical device is ideally suited to interactive diagrams but toggles work with static diagram too.



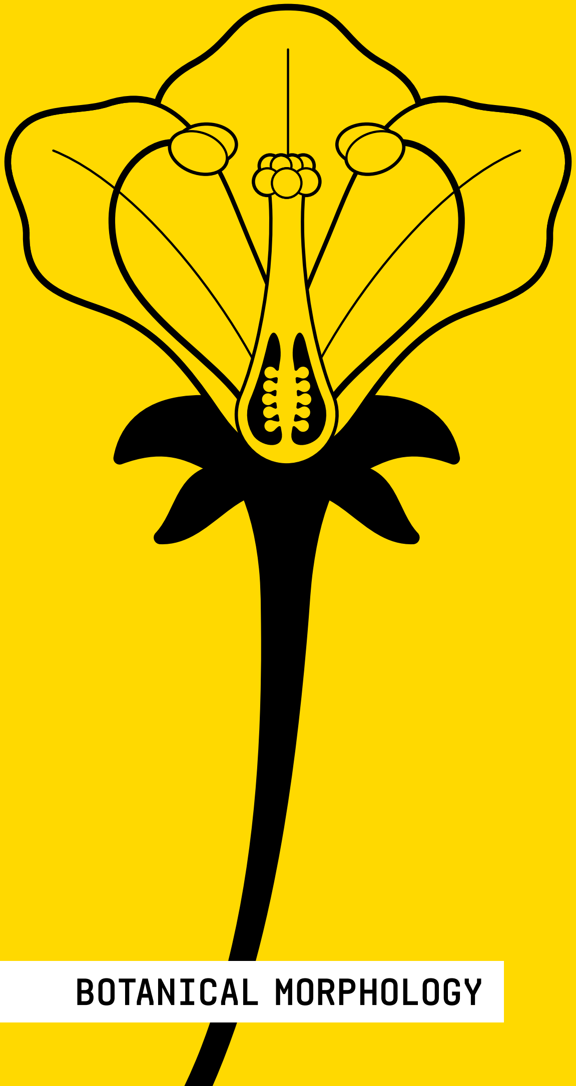
That is a quick tour of the **codex**. Not everything has been covered here, so I encourage you to peruse the poster and read the detailed explanations. There are plenty of examples to consider. And if you find a graphical device to be particularly interesting, the source materials are listed for further study. As an open-source tool, you are free to share the codex with others or use it in your work. I would like to end this presentation with a look to the future of the project. Several innovative notations did not quite make it into the codex because they are not systems notations per se. Perhaps they can be called "systems-adjacent" notations. They have noteworthy features. So I'd like to close-out this presentation by reviewing three such notations. Before that, I have a few words to say about other inspirations I have been collecting.



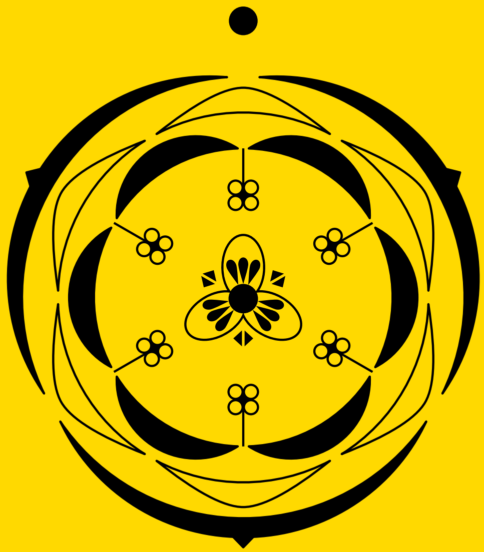
System notations have been losing ground to other modes of expression in terms of vividness, flexibility, and accessibility. Elsewhere* I have discussed how information graphics, video games, board games, film special-effects, visual programming languages, and so forth can be used to advance notations. There is a need for greater cross-fertilization of methods, not just across disciplinary siloes, but across media. Those **inspirations** extend to other types of notation as well. The blind-spots of existing system notations might be filled by techniques invented for very different purposes. It is in that spirit that I'd like to review a few "systems-adjacent" notations.

* Peter Stoyko, "Making Health Systems Transparent with the Help of the ESCALADE Information Design Framework," *IIID VisionPlus Conference*, Vienna, Austria, May 25th, 2003.

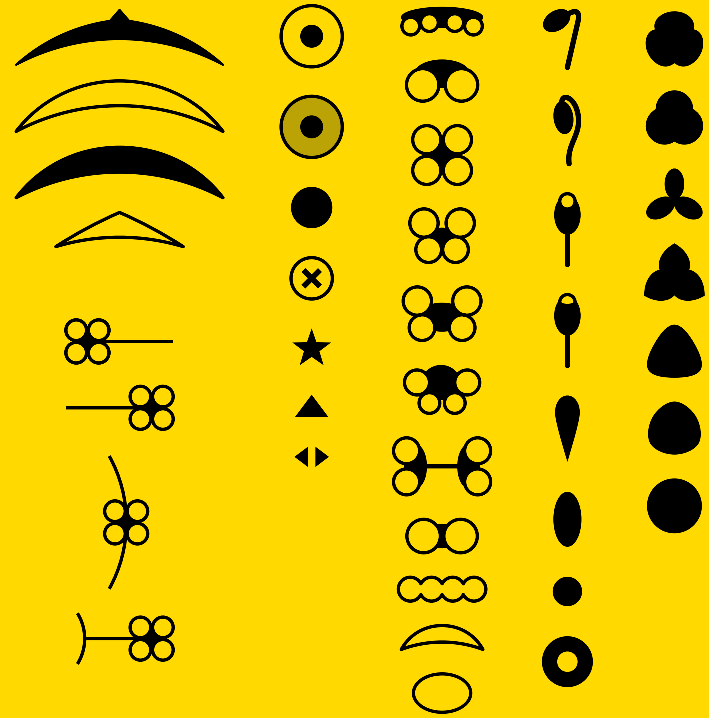
eichler flowers



BOTANICAL MORPHOLOGY



EXPRESSING FUNCTIONAL STRUCTURE (ORGANS)

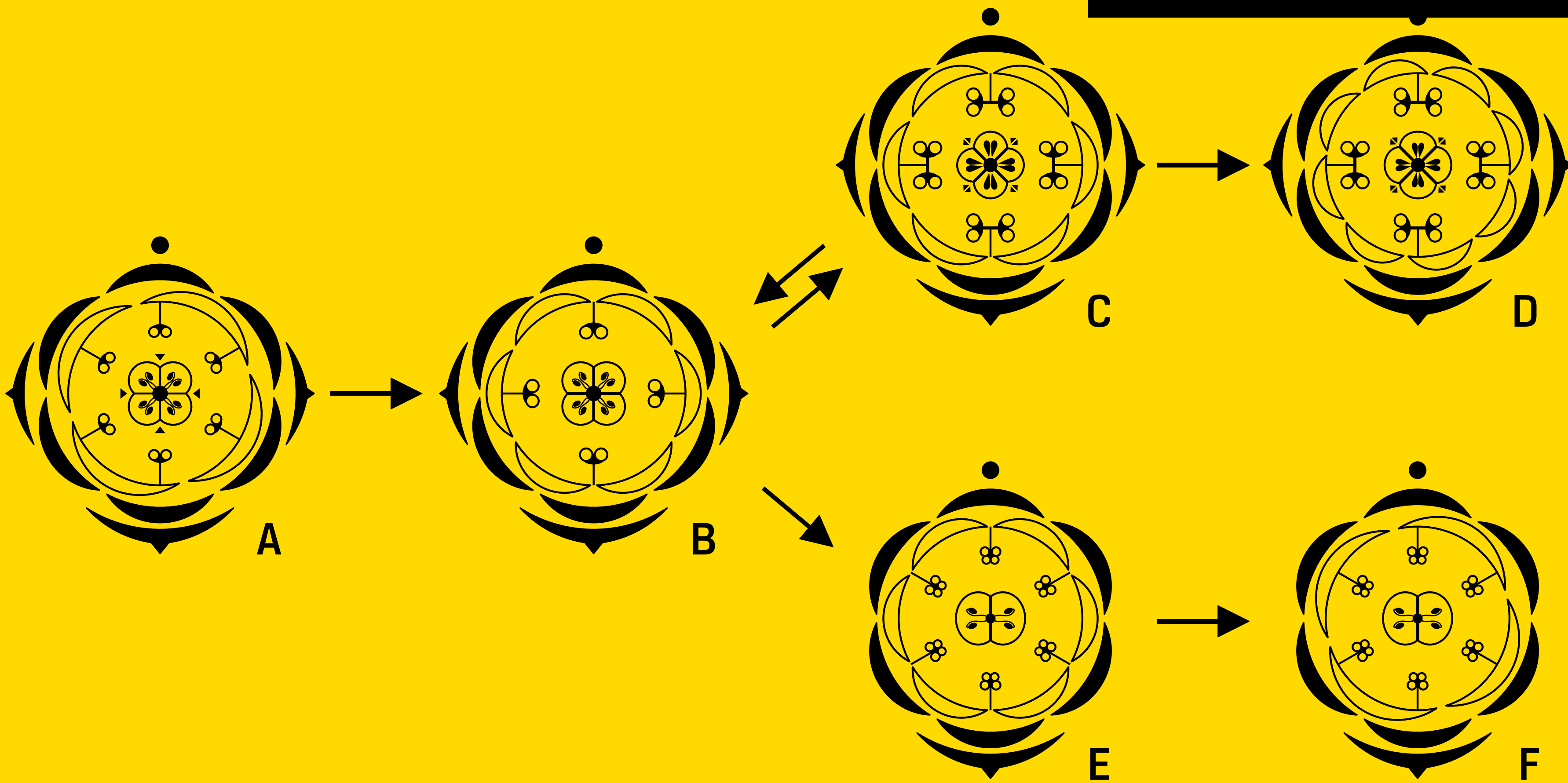


RONCE DE CRAENE NOTATION

The first inspiration is about making nodes better represent the real-world objects they are supposed to stand-in for. The 19th Century botanist August Eichler developed a notation to track the **morphology of flowers** due to evolution by natural selection within ecosystems, modernized by Ronse De Craene among others.* Morphology is structural change disregarding underlying causes, thus not concerned with systems per se but merely their outcomes. The notation shows how changes to object state (or configuration) can be portrayed over time in a relatable (low abstraction) picture using standardized pieces. Admittedly, this is a highly specialized notation but nonetheless offers inspiration for other subjects.

* Lewis P. Ronse De Craene, *Floral Diagrams: An Aid to Understanding Flower Morphology and Evolution* (Cambridge: Cambridge University Press, 2010).

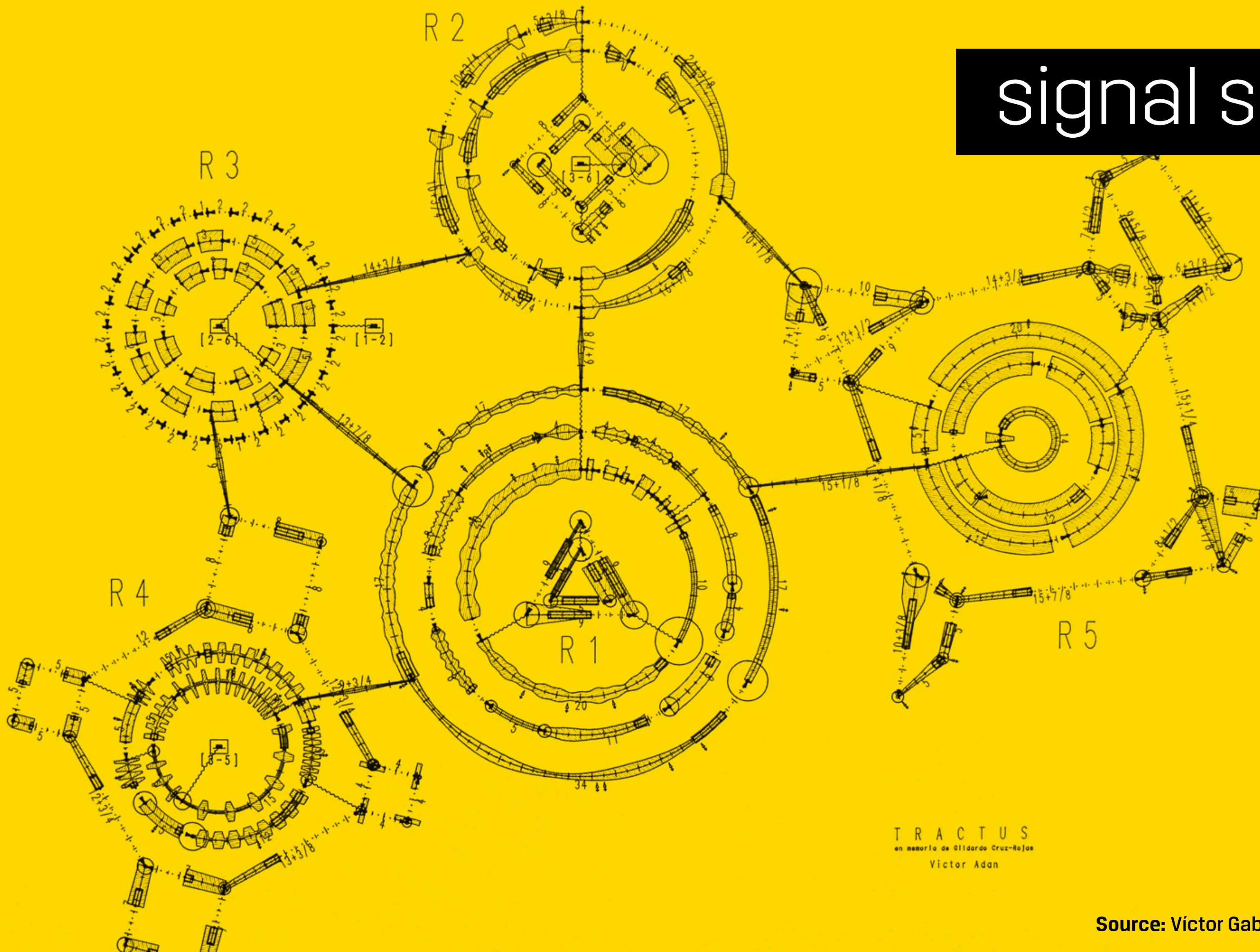
eichler flowers



The example shows the Ronce De Craene notation in action. Notice that these developmental pathways have a stronger relation to systems notation than the usual way morphology is diagrammed with cartograms (illustrated family trees). It is not difficult to imagine how explanatory links can be added to such a diagram. These nodes were generated automatically using an online control-panel.* Thus, rather than relying on large libraries of images that anticipate every possible variation, which is impractical, appropriate objects can be generated on the fly as they are needed. Moreover, much of the fuss of **assembling images** from ready-made pieces using graphic software has been done away with, thanks to automation.

* Tomáš Kebert, *Floral Diagram Generator* [online service], www.kvetnidiagram.8u.cz

signal spaces

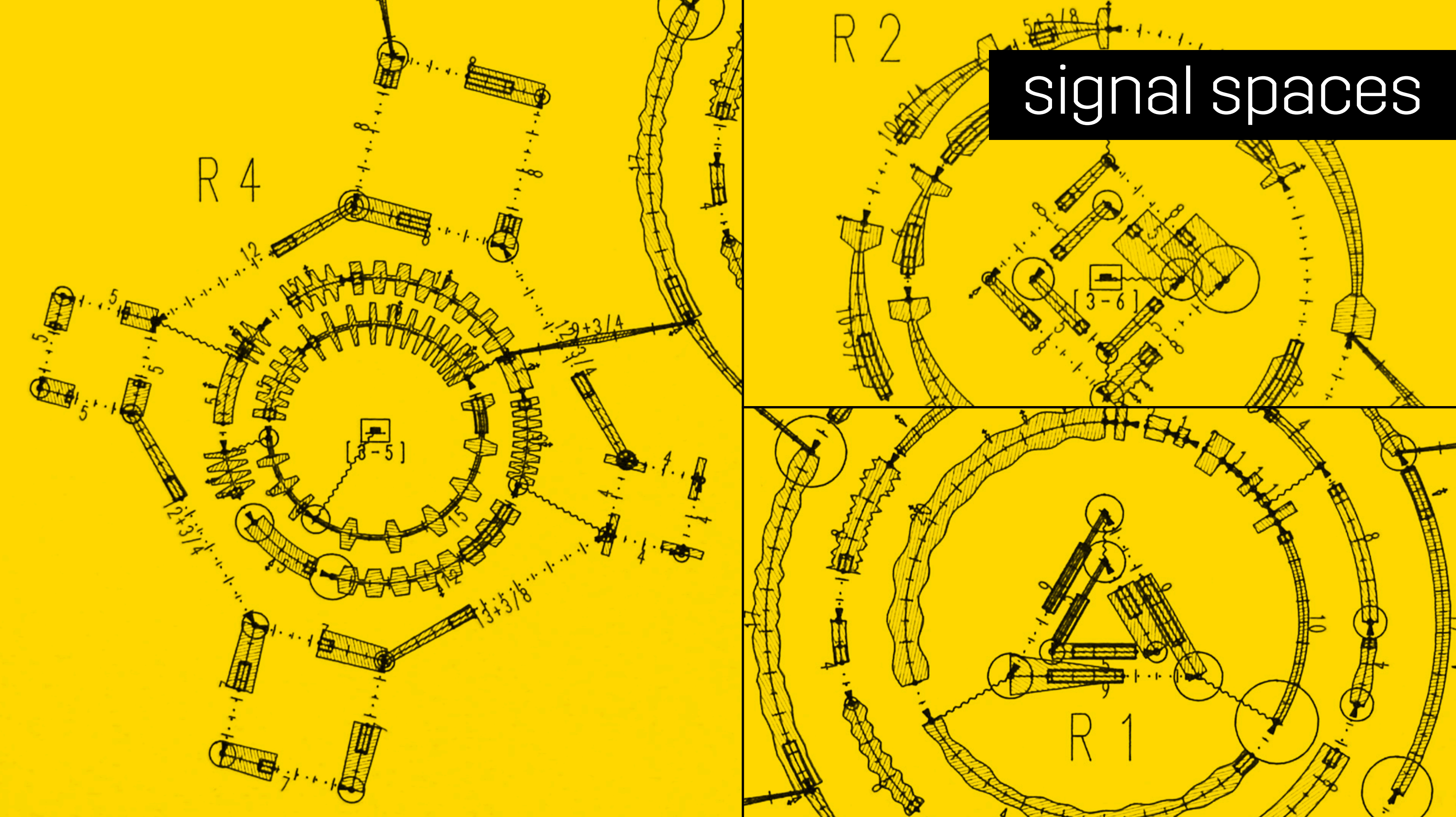


Source: Víctor Gabriel Adán, *Tractus* (2012)

The second inspiration is about music. Musical notations have existed for centuries but have struggled to keep up with avant-garde genres using novel instruments. There has been a lot of notational experimentation as a result. Of particular interest is the work of Víctor Adán, a composer and computer scientist with an interest in systems theory. He composed a piece for the Talea Ensemble called *Tractus* using a novel notation.* Music is a set of signals which Adán has decomposed into sequences of sounds, or instructional routines for particular instruments. They are arranged in parallel loops to create layered music. Musical movements are arranged into regions (R1-R5) like sub-systems. All that amounts to a mapping of temporal signals into space.

* Víctor Gabriel Adán, *Tractus*, performed by the Talea Ensemble, DiMenna Center, New York, April 20, 2012. Notation republished under a Creative Commons attribution license.

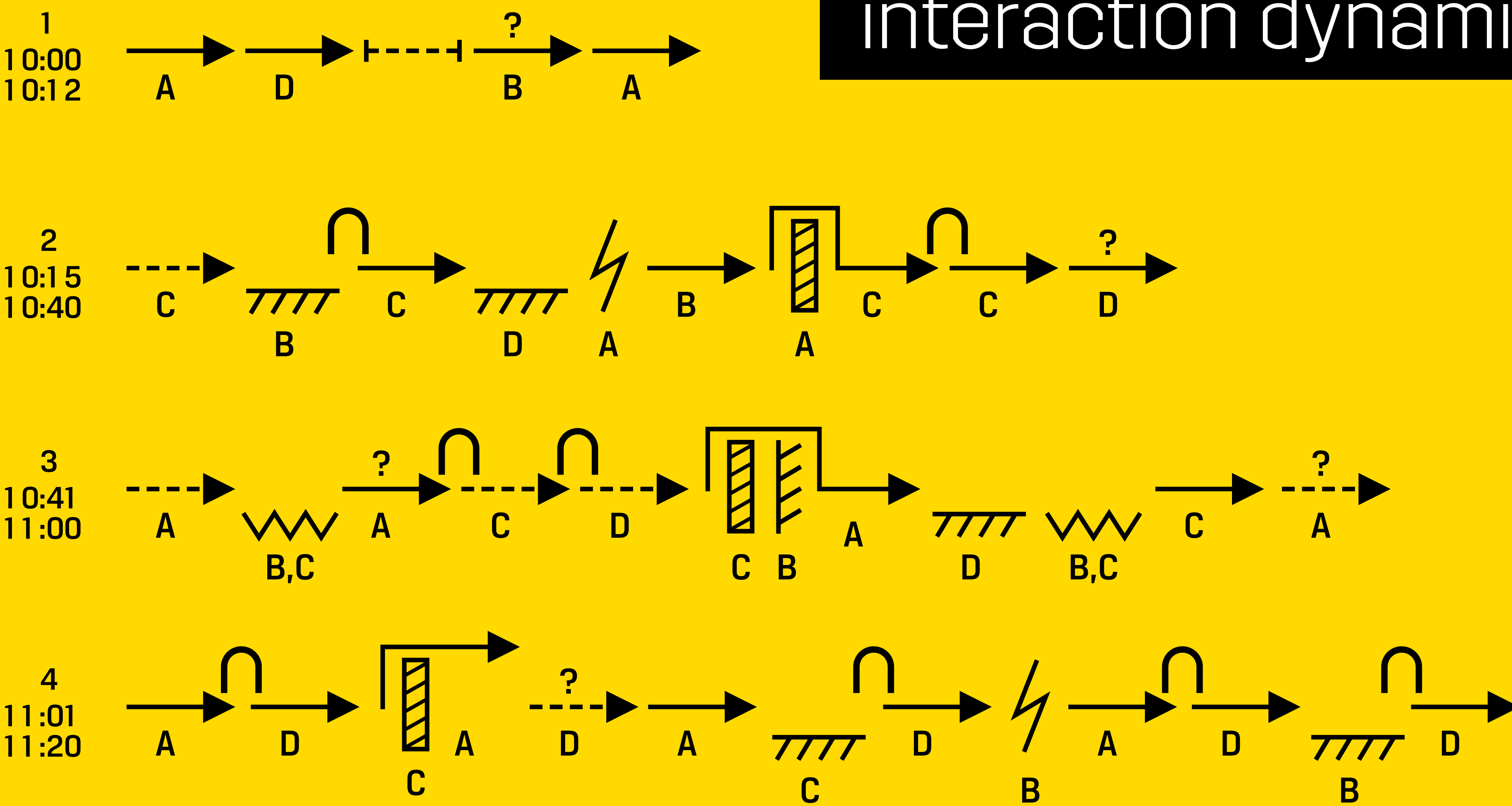
signal spaces



Zooming into the notation, notice the cross-walks between loops at particular junctures; the on- and off-ramps between sub-routines (or musical loops). Also notice how this notation does a good job of showing parallel processing. It is easy to see how certain sounds are expected to be played simultaneously through radial correspondence. What also makes Adán's notation of interest is the way it makes instructional signals a visual part of the diagram instead of encoding them into more compact forms. Several system theories focus on signals and boundaries, or signals and drivers. These theories might benefit from notations that show signals as detailed compositions with all sorts of nuances that parts of the system could react to.

* Víctor Gabriel Adán, *Hierarchical Music Structure Analysis, Modeling and Resynthesis: A Dynamical Systems and Signal Processing Approach*, MS Thesis, Massachusetts Institute of Technology, 2005.

interaction dynamics



The last inspirational example is about **social dynamics**. The *Interaction Dynamics Notation* (IDN) breaks down conversations within groups into a series of moves, counter-moves, and hesitations.* The resulting diagram shows the gambits and group dynamics that propel the conversation towards resolution (or breakdown). What makes this notation interesting is the way systemic frictions are shown on links as they progress through time. The conversation may get derailed by an interruption or humor. Or an argument may get blocked or pushed along a new direction. Or an assertion may gain support and be built upon. Thus, the tidy flow that is usually depicted by links is revealed to be full of messy frictions.

* See Neeraj Sonalkar et al., "Visualising Professional Vision Interactions in Design Reviews" *International Journal of CoCreation in Design and the Arts*, Vol. 13, nos. 1-2 (2016), pp. 73-92.

thank you

SYSTEMVIZ

www.systemviz.com

That completes the tour of the *Anatomy of System Notations*. Thank you for your attention. I encourage you to visit the SystemViz Project website to access the tools discussed in this presentation. The full poster can be downloaded there. Additional reference information can be found there too. If you have any questions, or feedback about the Anatomy, or about the project in general, please do not hesitate to contact me. I am particularly grateful for the systemic design community (the Systemic Design Association in particular) for its support of the project over the years.