Caresets

and the

Experimental Public Co-Design of Tomorrow

by

Deniz Aydemir

B.S. Computer Science and Philosophy
University of North Carolina at Chapel Hill, 2013

Submitted to the System Design and Management Program
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING AND MANAGEMENT

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2025

Authored by: Deniz Aydemir
System Design and Management
August 18, 2025

Certified by: Gioele Zardini
Rudge (1948) and Nancy Allen Assistant Professor, Thesis Supervisor

Certified by: Fábio Duarte de Araújo Silva
Principal Research Scientist, Thesis Supervisor

Accepted by: Joan Rubin
Executive Director
System Design and Management Program

Caresets

and the

Experimental Public Co-Design of Tomorrow

by

Deniz Aydemir

Submitted to the System Design and Management Program
on August 18, 2025 in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING AND MANAGEMENT

ABSTRACT

Participatory planning can help cities make better policy and planning decisions. An effective participatory planning framework must represent complex urban systems (compositional), compute Pareto-optimal solutions (computational), and incorporate residents into the decision-making process (collaborative).

We build a mathematical language using partially ordered sets to formally describe *care* in three forms: preference, ethics, and design. We then create a compositional, computational, and collaborative framework for participatory planning called the Experimental Public Co-Design of Tomorrow (EPCODOT). This framework adapts monotone co-design to work with our language of care and extends the approach with a collaborative interface. We demonstrate EPCODOT's capabilities first by modeling an MIT Senseable City Lab research project on trade-offs between data privacy and urban well-being, and then by modeling its potential application with a real public project in Durham, North Carolina. We provide a software prototype at epcodot.com.

This work develops two interconnected contributions: a mathematical language of care connecting applied category theory, decision theory, and ethics; and EPCODOT, a software tool enabling participatory planning for researchers, communities, and cities. Future work should pursue four directions: exploring additional category-theoretic structures within the mathematical language, formalizing connections to social choice and decision theory, testing EPCODOT directly with communities, and enhancing the computational capabilities of the framework's monotone co-design adaptation.

Thesis supervisor: Gioele Zardini
Title: Rudge (1948) and Nancy Allen Assistant Professor

Thesis supervisor: Fábio Duarte de Araújo Silva
Title: Principal Research Scientist

THESIS COMMITTEE

Supervisors

Gioele Zardini
Rudge (1948) and Nancy Allen Assistant Professor
Department of Civil and Environmental Engineering

Fábio Duarte de Araújo Silva
Principal Research Scientist
Department of Urban Studies and Planning

Reader

Bailey Flanigan
Assistant Professor
Department of Political Science
Department of Electrical Engineering and Computer Science

Contents

How can cities make better decisions?

Participatory Planning

Participatory planning is a form of community engagement in urban planning that shares decision-making power directly with residents (Forester, 1999; Afzalan and Muller, 2018). The promise of participatory urban planning, and of participatory governance more generally, is to create a public decision-making process that leads to higher quality and more equitable outcomes for residents (OECD, 2020).

But many attempts at participatory planning remain at basic engagement levels (U.S. EPA, 2024; Arnstein, 1969). Where deeper processes have been attempted, positive effects are observed only when participation is well designed and institutionally embedded (Mansuri and Rao, 2013). Successful participatory planning should allow residents to directly impact decisions and project plans, and ensure that the effects of those decisions are well understood (OECD, 2020). Participatory budgeting has produced notable outcomes when it is sustained and embedded, but this type of consequential, well-informed participation is still rare in more general planning contexts (Gonçalves, 2014).

This does not mean we should give up on participatory planning. When done well, collaborative decision-making can produce better policy outcomes (OECD, 2020). When decision-making is brought into the public sphere, it can also lead to more local attachment and civic participation (OECD, 2020). Residents themselves want to have a say in decisions that affect them, and when given that opportunity, trust in institutions improves (OECD, 2020).

Participatory Planning Today

There are many tools that local governments and planners use for community engagement and model-building, but none have provided a complete solution for participatory planning (Afzalan and Muller, 2018).

Decidim A set of open-source participatory tools for transparently accepting public input on proposals, budgeting, and projects. Decidim has been deployed at scale in multiple cities (Borge et al., 2023).

Pol.is A tool for discovering and aggregating public sentiment at scale. Pol.is has been used extensively in multiple cities (Small et al., 2021).

UrbanSim An open-source platform that can simulate policy impacts on multiple types of interrelated urban systems like real estate and land use. UrbanSim has been used by many cities and municipalities (Waddell, 2002).

MIT CityScope A simulation and design software that lets participants make proposals and see live feedback on their decisions. CityScope has been deployed across multiple cities (Alonso et al., 2018).

System Dynamics A framework that uses stocks, flows, and causal connections to model complex interactions, show trade-offs, and simulate outcomes. System Dynamics has seen limited deployments in public contexts (Veldhuis et al., 2024).

Decidim and Pol.is successfully solicit and surface public input, but they lack native representations of complex urban systems and provide no computational assistance for building consensus or selecting desirable proposals (Afzalan and Muller, 2018).

UrbanSim is a powerful tool for computing the impacts of potential policy decisions, but it is primarily used by experts due to its technical complexity (Waddell, 2002). The computational power cannot be easily leveraged by residents to understand impacts, and there is no native way to make it collaborative.

CityScope offers a compelling software prototype that allows participants to collaboratively simulate outcomes of their decisions. But it requires extensive customization for each urban environment, making it challenging for cities to deploy practically for their own projects (Alonso et al., 2018).

System Dynamics is novel because it can represent and simulate diverse types of systems to multiple degrees of detail. But it has been around for decades and has not broken through as a tool for use by public institutions (Veldhuis et al., 2024). Only in recent years has there been an effort to see how System Dynamics could be made collaborative (Hovmand, 2014).

What We Need

Based on what we see with the above tools, let's take a reasonable leap. Here are three requirements of a successful framework for participatory planning.

Composition The framework must be able to represent complex systems by breaking them down into components and do so at an appropriate level of detail for participants with varying expertise and background knowledge. The framework must be able to trace the impacts of proposals and preferences through the represented systems.

Computation The framework must be able to simulate outcomes derived from the inputs provided to the represented systems. The framework must be able to legibly surface trade-offs and discover Pareto-optimal solutions.

Collaboration The framework must provide residents an interface to submit proposals and state preferences. The framework must show how consensus is formed and how it impacts recommended solutions.

Each of the above tools misses at least one of these capabilities, so we aim to offer a novel framework that builds in all three natively.

Mathematical Formalisms for Participatory Planning

We know that mathematics is unreasonably effective in the natural sciences (Wigner, 1960). We also see mathematics applied to decision sciences (von Neumann and Morgenstern, 1944; Savage, 1954) and many other human phenomena through logical correspondences like the Curry-Howard-Lambek isomorphism. It is natural to ask if mathematical formalisms can be reasonably effective in helping us solve complex problems in urban planning.

Monotone co-design is an experimental framework for engineering design that uses applied category theory as its mathematical foundation. It represents system components as co-design problems that describe monotonic relationships between resources and functionalities, then composes these problems to represent more complex systems (Censi, 2015). Originally developed for engineering contexts, monotone co-design has the potential to be applied to diverse types of complex systems (Zardini, 2023; Zardini et al., 2023).

The term co-design is meant to represent a combination of multiple ideas: compositional design, computational design, and collaborative design (Zardini, 2023). This combination of capabilities makes monotone co-design a promising foundation for a participatory planning framework, but it remains untested in public and participatory scenarios.

There are two questions we need to answer to determine if monotone co-design has potential for use in participatory planning.

Can we express care using a formal language of ordered sets?

Monotone co-design expresses system functionalities and resources as partially ordered sets. This works well for engineering systems with technical components that have intuitively quantifiable outputs and requirements.

But planning is not about making decisions based only on what our urban systems *can* do. Urban planning is also concerned with what decisions and outcomes are *better*. We are concerned with helping cities make *better decisions*. This means making decisions and using a decision-making process that are aligned with our preferences and values.

We call this concern for making better decisions *care* and try to express *care* with a formal language of ordered sets.

Can we make monotone co-design participatory?

With its compositional structure, monotone co-design can represent the priorities of different stakeholders as different co-design problems.

But there is no way to take input and actively negotiate those differences between stakeholders. In order to be usable for participatory planning, a framework built on monotone co-design must add a collaborative layer that allows residents and stakeholders to provide input and priorities that contribute directly to the final decision-making process.

Towards a Public Co-Design

If we can show how to use ordered sets to express *care*, we hope to be able to rigorously express what communities care about. If we can then make monotone co-design participatory, we can build the composable, computable, and collaborative *Experimental Public Co-Design of Tomorrow* that governments, planners, communities, and organizations can use for participatory planning.

Our Plan

Question How can cities make better decisions?

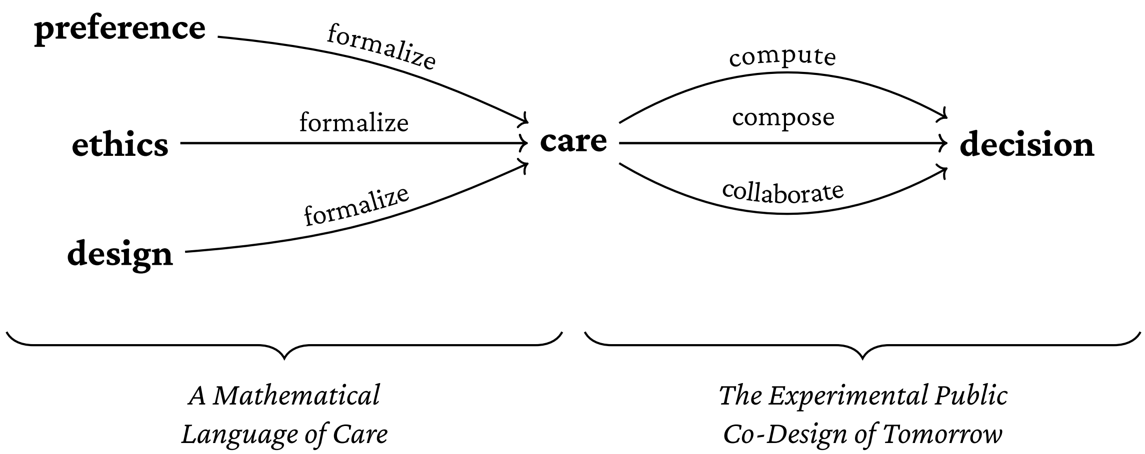
Scope We take our concern for making *better decisions* to mean *care*. We take *decisions made by cities* to mean urban planning projects, programs, and services.

Claim Ordered sets can be used to express *care* and monotone co-design can be adapted into an effective framework for participatory planning.

Method We develop a *mathematical language of care* using ordered sets and extend monotone co-design with collaborative interfaces to create the *Experimental Public Co-Design of Tomorrow*.

Test We present two case studies of this *Experimental Public Co-Design of Tomorrow*: one demonstrating the framework's basic capabilities, and another showing its application to a complex, real urban planning project.

Payoff A step toward expressing care using mathematical formalisms that touches various theories across social choice, decision science, and design. A framework and accompanying software prototype of a new compositional, computational, and collaborative framework for participatory planning that can be used in future research and community engagement efforts.


preference
formalize
ethics
formalize
design
formalize
care
compute
compose
collaborate
decision
A Mathematical
Language of Care
The Experimental Public
Co-Design of Tomorrow

The Plan

—

At this age she communicates more with movements than with words.

In fact, she communicates with no words at all. Just a look or a smile. Sometimes a loud sound. Ever since we learned that pointing is "prelinguistic", our gaze diligently follows her arms.

Today, she pointed.

It's a remarkable form of communication, pointing. You first have to be in the same space with someone else, then you have to understand where in that space the thing you care about is, then you have to draw an arrow to the thing you care about.

All that to tell us she's hungry. Or whatever it is she's trying to say, who could possibly know.

—

better

↑

worse

Part One

A Mathematical Language of Care

Chapter 1

Care

1.1 Claim

We care.

We care about many things or very few. We care about things that matter. Maybe all that matters is what we care about. We care needlessly. Maybe we should care more.

But we care.

Care can mean many things. We may care about something, or care for something. We may say that certain actions are more caring, or that care is a way of being. There are care ethics, and we can consider how to bring about more care in the world.

We initially used care to describe our concern for making better decisions, and this framing will guide how we build up this language of care.

But we must acknowledge the vastness of what care could mean, and acknowledge that we certainly will not be able to test our language against the whole vastness.

Still, we hope that our mathematical language will be flexible and robust enough to describe what goes on when we care – to express the physics of care.

1.2 Thoughts

If our initial framing of care is as a concern for better decisions, we can start with a thought.

Care requires imagining more than one possible world.

For this, and likely any conception of care, we must be able to see more than one possible world. If we cannot honestly imagine more than one possible world, then we cannot exhibit any form of care (Keltner and Stamkou, 2025). Even if we believe an outcome to be inevitable, if we care about the outcome, we must be able to imagine a world where the outcome is not realized.

The possible worlds we imagine can be outcomes, states, situations, actions, or anything else we may care about.

Possible World An imaginable potential form of the universe.

We can add a second thought.

Care requires a concern for the choice of possible worlds.

There may be infinite possible worlds (there are), but that does not require us to care. In any conception of care, we must have some feeling about those possible worlds. Even if we do not have anything to say about two given possible worlds, care requires us to have some capacity for concern.

We can describe our concern as a capacity for believing in the existence of *better possible worlds*.

Better A way to relate two *possible worlds* according to our care.

1.3 Forms

Our care is concerned with realizing *better possible worlds*. To reflect this, we will incrementally build up a mathematical language of care that is concerned with *possible worlds* that are aligned with our preferences (*preference*), with *possible worlds* that are aligned with our values (*ethics*), and with the decision-making process itself (*design*).

Preference

> *There is naturally an intimate connection between what a person cares about and what he will, generally or under certain conditions, think it best for himself to do.* — Harry Frankfurt

Caring organizes what we consider best to do (Frankfurt, 1988). Caring is a pattern of emotions and evaluations that makes things matter. Care is what creates the structures of our decision theory models (Frankfurt, 1988; Helm, 2001).

Ethics

> *On the most general level, we suggest that caring be viewed as a species activity that includes everything that we do to maintain, continue, and repair our 'world' so that we can live in it as well as possible. That world includes our bodies, our selves, and our environment, all of which we seek to interweave in a complex, life-sustaining web.* — Joan Tronto

Care is not only what we prefer, it is also an expression and realization of our belief in justice (Tronto, 2013). Care is what gives us reasons, obligations, and ethical values (Korsgaard, 1996).

Design

> *Everyone designs who devises courses of action aimed at changing existing situations into preferred ones.* — Herbert Simon

Design is the work of discovering and realizing better futures. Design takes "matters of fact" and converts them to "matters of concern", necessarily including an ethical dimension. Design is care in action (Simon, 1988; Latour, 2008).

1.4 Poetry

Any expressive language can be used for poetry (Alexander et al., 1978). As we explore and build this mathematical language, let's keep an eye out for the poetry hidden at the edges of the prosaic.

Chapter 2

Preference

2.1 Better

2.1.1 Basic Preferences

What does it mean to prefer a possible world?

I may prefer the world where I'm happy tonight to the one where I'm sad, or prefer the one where I'm full to the one where I'm hungry.

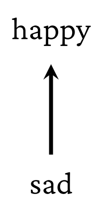


Figure 2.1: Emotions tonight

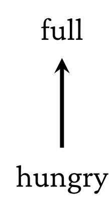


Figure 2.2: Hunger tonight

Note that we aren't speaking of actions here. We may hope a certain future comes true, and we may take an action to help realize that future. But we may not take any action at all.

I may prefer the world where I eat a cookie this evening to the one where I don't, and I may bake cookies to help realize that future. I may not be eligible to vote, but I may still care who wins.

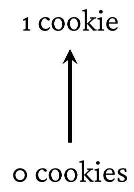


Figure 2.3: Cookies eaten tonight

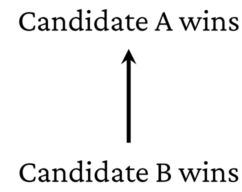


Figure 2.4: Election result

Thinking of preference in this way allows us to first understand what it is to care about something in the weakest sense. To care about something in the mind, whether or not it leads to action. This weak, mental care is what we call preference.

You can see the foundation of a language in these figures. We've drawn arrows that indicate when one possible world is preferable to another. We might say each drawing states a care.

2.1.2 Many Worlds

We may care about more than just two possible worlds. We may have a whole order of preferences. Our drawing may need more than two elements.

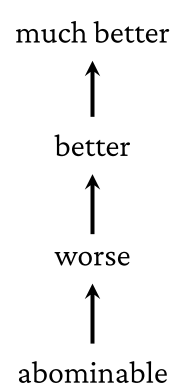


Figure 2.5: Much better and much worse

But the way we care is not always so clear. We may not know enough to have a preference. At lunch, we may know we prefer a salad to a soup or a sandwich, but between a soup and a sandwich, we are indifferent.

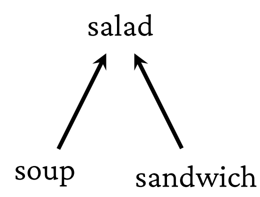


Figure 2.6: Lunch

2.1.3 Statements of *Better*

Let's formalize what it means to prefer a possible world.

When we state a preference, we state that we believe one world is better than another (Broome, 1991; Hare, 1963; Harsanyi, 1955). If we are capable of preference, then we are capable of believing that some worlds are better than others.

We use *better* over *prefer* because we want to be able to describe more than just preferences. For now, we are using *better* to mean *better according to our preferences*, but better can mean more.

We may say that some worlds are good and some are bad. We likely prefer good worlds to bad worlds. But this distinction is stronger than just preference and implies that there is an important line to draw between good worlds and bad worlds. We will allow ourselves to make stronger, normative statements in Chapter 3: Ethics.

For now, we can say that good worlds are better than bad worlds. And better worlds are better than worse worlds.

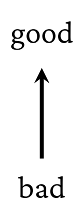


Figure 2.7: Good and bad

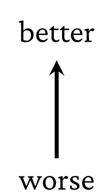


Figure 2.8: Better and worse

What are the different statements we can make with this idea of *better*?

Statements of Better

For any two possible worlds a and b, we must be able to express exactly one of the following three possibilities:

1. a is better than b.
2. b is better than a.
3. a is not better than b, and b is not better than a.

We can draw each of these forms of preference as well.

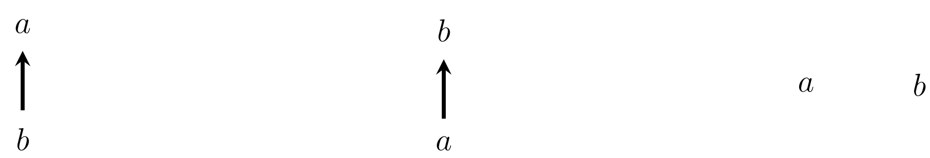


Figure 2.9: All statements of *Better*

These are the three types of statements our language can form about any pair of possible worlds. But we want to be able to speak about more than two possible worlds.

We have already seen how we can chain these statements to speak about more than just two possible worlds.

a is better than b and b is better than c

a is better than b and a is better than c

Figure 2.10: Two chained statements of *Better*

These statements are all ordered sets that describe our care.

2.2 *Caresets*

2.2.1 Picking an Ordered Set

In order to formalize our statements mathematically, we need to decide what type of ordered set to use.

Let's consider three candidates: preorders, weak partially ordered sets, and strong partially ordered sets.

Preorder

A set P equipped with a binary relation $\preceq$ that satisfies:

- **Reflexivity:** For all $x \in P$, $x \preceq x$.
- **Transitivity:** For all $x, y, z \in P$, if $x \preceq y$ and $y \preceq z$, then $x \preceq z$.

If neither $x \preceq y$ nor $y \preceq x$, we say x and y are *incomparable*.
If both $x \preceq y$ and $y \preceq x$, we say x and y are *equivalent*.

Weak Partially Ordered Set

A set P equipped with a binary relation $\preceq$ that satisfies:

- **Reflexivity:** For all $x \in P$, $x \preceq x$.
- **Transitivity:** For all $x, y, z \in P$, if $x \preceq y$ and $y \preceq z$, then $x \preceq z$.
- **Antisymmetry:** For all $x, y \in P$, if $x \preceq y$ and $y \preceq x$, then $x = y$.

If neither $x \preceq y$ nor $y \preceq x$, we say x and y are *incomparable*.

Strict Partially Ordered Set

A set P equipped with a binary relation $\prec$ that satisfies:

- **Irreflexivity:** For all $x \in P$, it is not the case that $x \prec x$.
- **Transitivity:** For all $x, y, z \in P$, if $x \prec y$ and $y \prec z$, then $x \prec z$.
- **Asymmetry:** For all $x, y \in P$, if $x \prec y$, then it is not the case that $y \prec x$.

If neither $x \prec y$ nor $y \prec x$, we say x and y are *incomparable*.

Let's pull out the differences to see what each structure offers (and requires) compared to the others.

Compared to a preorder, a weak partially ordered set (poset) requires that if two elements are both related by $\preceq$, then they must be $=$ (*antisymmetry*). This means that weak posets do not differentiate between equivalence and incomparability, where preorders do.

Compared to a weak poset, a strict poset uses a different type of binary relation: $\preceq$ vs $\prec$. $\prec$ does not relate an element to itself, while $\preceq$ does (*irreflexivity* vs *reflexivity*).

When we consider how these differences connect to our statements about *possible worlds*, two questions jump out.

1. Do we want to be able to differentiate between equivalence and incomparability?
2. Should *better* be $\prec$ or $\preceq$?

Do we want to be able to differentiate between equivalence and incomparability?

Another way to put this question would be to ask if these two structures should say different things.

Figure 2.11: Two chained statements of *better*

If we use a preorder, we can say precisely whether we believe *soup* and *sandwich* are equivalent or if they are incomparable.

There is a strong case to be made that preorders do a better job of describing the subtleties of what it means to care. We might have uncertainties that keep us from saying one possible world is better than another, while also encouraging us to avoid saying the two possible worlds are equivalent.

But the idea of *better* we have used so far doesn't need to differentiate between equivalence and incomparability. Either one world is better than another or it isn't.

We can think of our language as a way to make choices, which is what we hope it will eventually be useful for. If we have no *preference* between two worlds, labeling the two worlds as *equivalent* or *incomparable* makes no difference to our choice. All we need to know is if one world is *better* or not.

There is also an added cognitive cost to this differentiation. *Caresets* reduce the differences between possible worlds to a single relationship: is one possible world better than another? Choosing to represent our care with preorders requires us to ask a follow-up question for any two possible worlds where one is not better than the other: are these two possible worlds equivalently good or simply incomparable?

Answering this requires us to define a threshold for comparability. How much information do we need before we can say two worlds are equivalent rather than incomparable? Any such threshold risks being arbitrary.

We will not blame anyone for wanting to look further into the ramifications of using preorders to describe how we care, but for our purposes, we will prefer posets.

Should we use $\prec$ or $\preceq$ to represent *better*?

We can bring back the lightweight definition of *better* we proposed in Chapter 1: Care.

> **Better** A way to relate two *possible worlds* according to our care.

To create a binary relation from this idea of *better*, we need to be a little more specific. If we choose to use the standard usage of the word "better," that aligns more with $\prec$. A possible world cannot be better than itself.

But we can also define the relation to use a weakened form of *better* that aligns with $\preceq$. This would open our structures to more types of mathematical operations and theorems that are defined for weak posets, like representing our poset as a category in category theory.

Lucky for us, weak and strong posets are isomorphic to each other. This means that no matter which structure we use, we can convert between the two without losing any information.

This allows us to both define our *better* relation as a strict $\prec$ that aligns more naturally with the way we use the word in natural language, and also leverage the mathematical properties that come with weak posets.

From now on, we will not differentiate between weak and strong posets unless necessary. Our drawings will describe strong posets because that allows us to avoid drawing the reflexive arrows from every element back to itself and allows us to say without restraint that our arrows indicate one world is better than another.

2.2.2 Complete Set of Possible Worlds

We should also specify how to structure a complete poset of possible worlds, so we will require it to be both *mutually exclusive* and *jointly exhaustive*.

Mutual exclusivity means it is only possible for one of the possible worlds to be realized. In a complete set of possible worlds, there is no way for us to ever exist in more than one possible world simultaneously. This is required in order to be clear about what it means to prefer one world to another.

Joint exhaustiveness means that the set of possible worlds represents all possible worlds that can be realized within the given scope of care.

Let's take a look at our lunch preferences again.

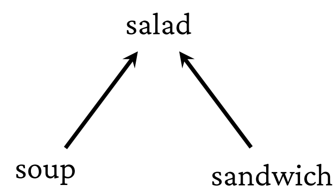


Figure 2.12: Lunch

In order to meet the requirements of a *complete set of possible worlds*, we have to assume the above drawing is equivalent to the following.

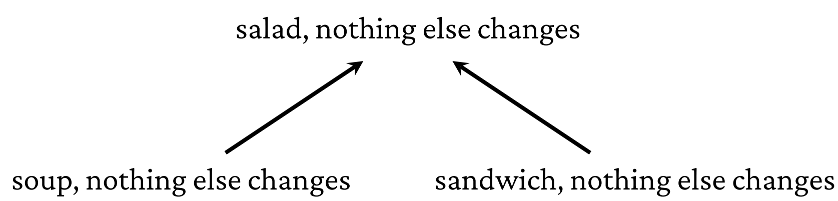


Figure 2.13: Lunch, explicitly complete

Whenever we create a drawing like the first, we will assume that it is *complete* and can also be written like the second.

It is worth noting that both these statements assume that one of *soup*, *salad*, and *sandwich* must come true to be jointly exhaustive. We know that this is not a reflection of reality – it is possible none of these three choices are realized.

To resolve this, we will assume that every statement of care describes all the possible worlds *that relate to our given scope of care*. This will satisfy our quality of joint exhaustiveness, and will later allow us to combine these statements and maintain joint exhaustiveness throughout those combinations.

2.2.3 Put the Pieces Together

Let's put the pieces together.

Careset

A ***careset*** is a set C of *possible worlds*, equipped with a binary relation $\succ$ such that:

- **Irreflexivity:** For all $x \in C$, it is not the case that $x \succ x$.
- **Transitivity:** For all $x, y, z \in C$, if $x \succ y$ and $y \succ z$, then $x \succ z$.
- **Asymmetry:** For all $x, y \in C$, if $x \succ y$, then it is not the case that $y \succ x$.

C is a *complete set of possible worlds* such that

- **Mutual Exclusivity:** For any $x, y \in C$, if $x \neq y$, then x and y cannot both be realized.
- **Joint Exhaustiveness:** C includes all *possible worlds* that could be realized within the given scope of care.

The relation $x \succ y$ encodes the idea that x is *better* than y.
The structure $(C, \succ)$ forms a strictly partially ordered complete set of *possible worlds*.

Let's call these posets of care *caresets*.

2.3 A Language of *Caresets*

Caresets will be the primitives of our language.

Imagine all the factors that go into choosing a job, or to picking a location to live. Because of the complexities, we may not always know which we prefer. Let's see what they look like as *caresets*.

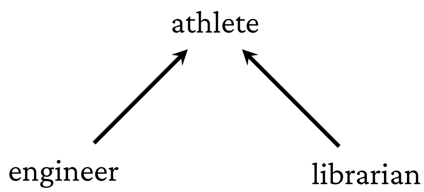


Figure 2.14: Jobs *careset*

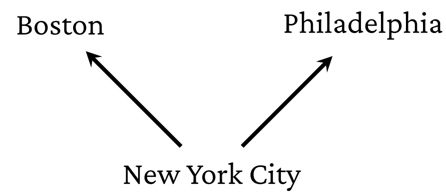


Figure 2.15: Locations *careset*

Using *caresets*, we can hint at potentially complex feelings with just a few arrows and words. Above, we might be describing someone who dreams of becoming a professional athlete. But between being an engineer and a librarian they are torn. Maybe they have interests in both, or are considering the trade-offs between the money they will make and the life they will lead. Maybe all they know is they want to be an athlete and nothing else matters. All we know is that by not having an arrow between *engineer* and *librarian* we know that right now they cannot say which of the two they prefer today.

This is also why we choose posets instead of totally ordered sets. All totally ordered sets are also posets, so we always have the ability to describe our care as a total order. But keeping posets as the foundation grants us a flexibility that better reflects the complexities of caring.

2.3.1 Combining *Caresets*

Emotions $\times$ Hunger

Let's revisit the very simple and very agreeable drawings we used to represent possible evenings. We can now recognize these as *caresets*.

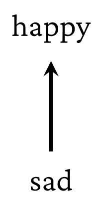


Figure 2.16: Emotions tonight

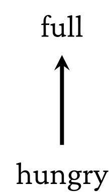


Figure 2.17: Hunger tonight

These *caresets* are drawn independently, but they describe related possible worlds that could be realized for us tonight. Because *caresets* are posets, we can take their product ($\times$). In our language of *caresets*, this will give us the full picture of possible worlds that are described by both of these distinct *caresets* as one combined *careset*.

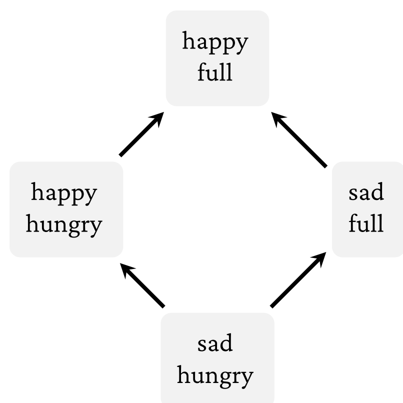


Figure 2.18: Emotions $\times$ Hunger tonight

This product *careset* is a Cartesian product of the possible worlds (all combinations of possible worlds from the two *caresets* are represented), and it maintains the ordering of the two *caresets* (the product order). *hungry* is never above *full* and *sad* is never above *happy*.

Product of Partially Ordered Sets

Let $(P, \preceq_P)$ and $(Q, \preceq_Q)$ be two partially ordered sets (posets). The **product poset** $(P \times Q, \preceq)$ is defined as follows:

- The underlying set is the Cartesian product $P \times Q = \{(p, q) \mid p \in P,\ q \in Q\}$.
- The order relation $\preceq$ on $P \times Q$ is given by:

$$(p_1, q_1) \preceq (p_2, q_2) \quad \text{if and only if} \quad p_1 \preceq_P p_2 \text{ and } q_1 \preceq_Q q_2.$$

For *caresets*, we will only allow taking the product of two *caresets* when all combinations of possible worlds are in fact possible. They do not have to necessarily be independent—my happiness might depend on whether I'm full. But it must be the case that the Cartesian product of the possible worlds of the two *caresets* still represents a mutually exclusive and jointly exhaustive set of possible worlds, like both the original *caresets* did. All four of these possible worlds should, at least logically, be possible.

Because our *caresets* are transitive, we also do not have to draw an arrow from *sad, hungry* to *happy, full*. Like in posets, only the arrows that are required to show all the relations are drawn.

The product *careset* has not only maintained the ordering of the two original *caresets* but also shown us where there is not enough information to decide if one possible world is better than another. If two possible worlds are not connected by arrows, either directly or transitively, then the *careset* cannot say if one is preferable to the other. The act of combining our two independent *caresets* has not been able to tell us which of the two is preferable between *happy, hungry* and *sad, full*. We only know that both are preferable to *sad, hungry*, and *happy, full* is preferable to both.

Caresets allow us to keep this uncertainty if it reflects our actual preferences. But maybe when we see this combined *careset*, we can resolve this uncertainty.

Tonight, we prefer to be *happy* and *hungry* to *sad* and *full*. Let's draw the line that describes this preference.

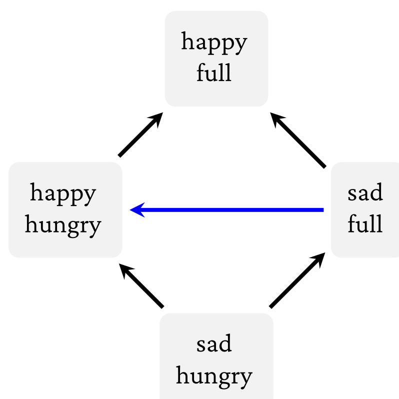


Figure 2.19: Emotions × Hunger tonight, with new assessment

Let's remove the extraneous arrows.

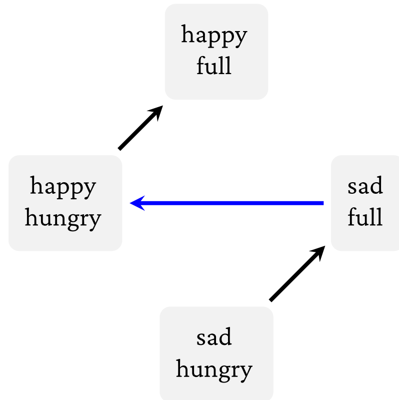


Figure 2.20: Emotions × Hunger tonight, with minimum arrows

Let's have all our arrows point upward. To be as legible as possible, we will have *caresets* always go from most preferable to least preferable, top to bottom.

Figure 2.21: Emotions × Hunger tonight, after assessment

Complexities of how we care can be revealed and addressed with product *caresets*.

Jobs × Locations

Let's consider our optimistic friend, the one who dreams of becoming an athlete and has a potentially contentious preference to live in either Boston or Philadelphia over New York City. Your profession and your location are related, and we should relate them.

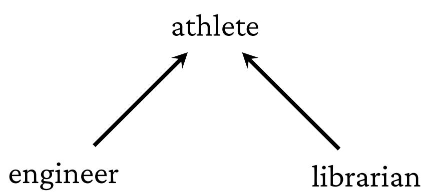


Figure 2.22: Jobs

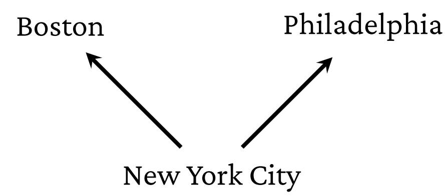


Figure 2.23: Locations

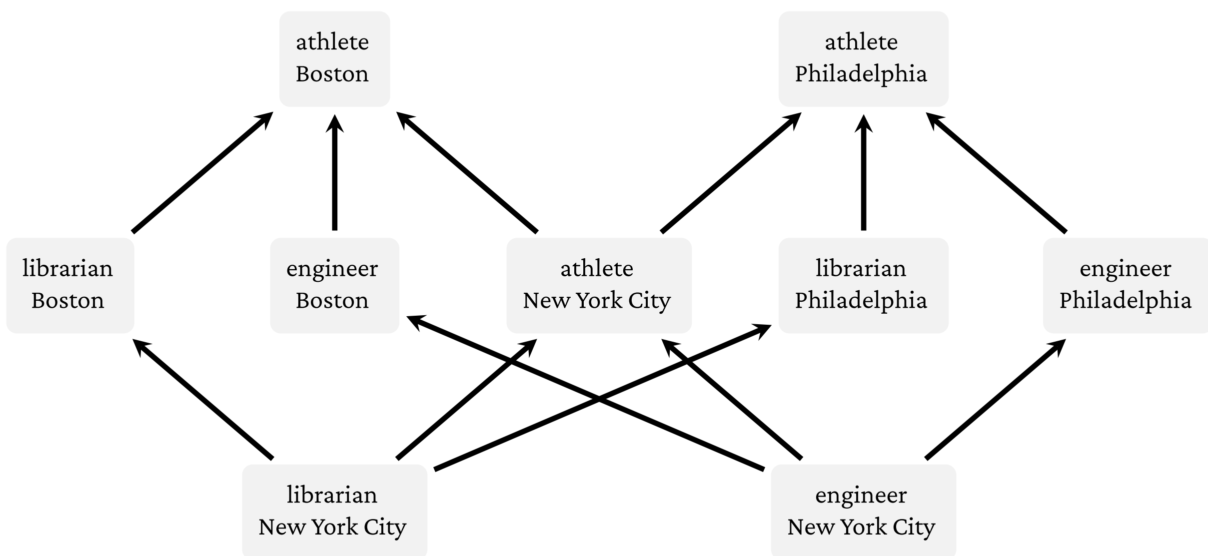


Figure 2.24: Jobs × Locations *careset*

This product *careset* is more complicated than the Emotions × Hunger *careset*, but it's the same rule being used to create the relationships. Every arrow (direct or transitive) in the product *careset* means that both the projection *caresets* had arrows in the same direction between their respective elements.

This is why *librarian* × *New York City* and *engineer* × *Boston* do not have an arrow between them. Even though *New York City* has an arrow to *Boston* in *locations*, there is no arrow from *librarian* to *engineer* in *jobs*.

Saying we prefer living in *Philadelphia* over living in *New York City* is a simple statement that implies many unstated changes. When we place elements in a *careset*, we naturally infer from context the effects on unspecified factors. The home where you live, the community you are surrounded by, and the local sports team will all be necessarily different. But some things might be possible to imagine holding equal, like income, proximity to a grocery store, and the existence of a local NBA team.

If we want to make a specific factor explicit rather than inferred, then it is the responsibility of the *careset* to include it in the description of the possible world. The *careset* needs to describe the objects of our *care*.

If there are multiple factors we need to make explicit, then they may be best stated as separate *caresets*. For example, we can imagine a companion *Incomes* careset that we can eventually combine with our *Jobs* $\times$ *Locations careset*.

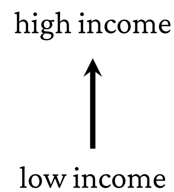


Figure 2.25: Incomes *careset*

That combined *careset* would remove the assumption that we hold income equal across the different locations, even if the likelihood of finding a *high income librarian* job is low.

By having the possible worlds in our *caresets* be mutually exclusive and jointly exhaustive, we avoid missing any possibilities. Not every possible world has to be equally likely, but our *caresets* are precise about what possible worlds are *better*.

2.3.2 Antichains

When we begin constructing *caresets* with multiple elements that do not have arrows between them, we may be interested in looking at these sets of uncompared elements.

Antichains are groups of elements in a poset that do not have any relations with each other, either directly or transitively through other elements. An antichain represents a group of elements where we cannot say any one of the elements is preferred to any of the others.

Antichain

Let $(P, \preceq)$ be a partially ordered set. An **antichain** in P is a subset $A \subseteq P$ such that for all $x, y \in A$ with $x \neq y$, neither $x \preceq y$ nor $y \preceq x$ holds.

A maximal antichain is an antichain where no more elements can be added to the antichain. In our Jobs × Locations *careset* we can see four different maximal antichains.

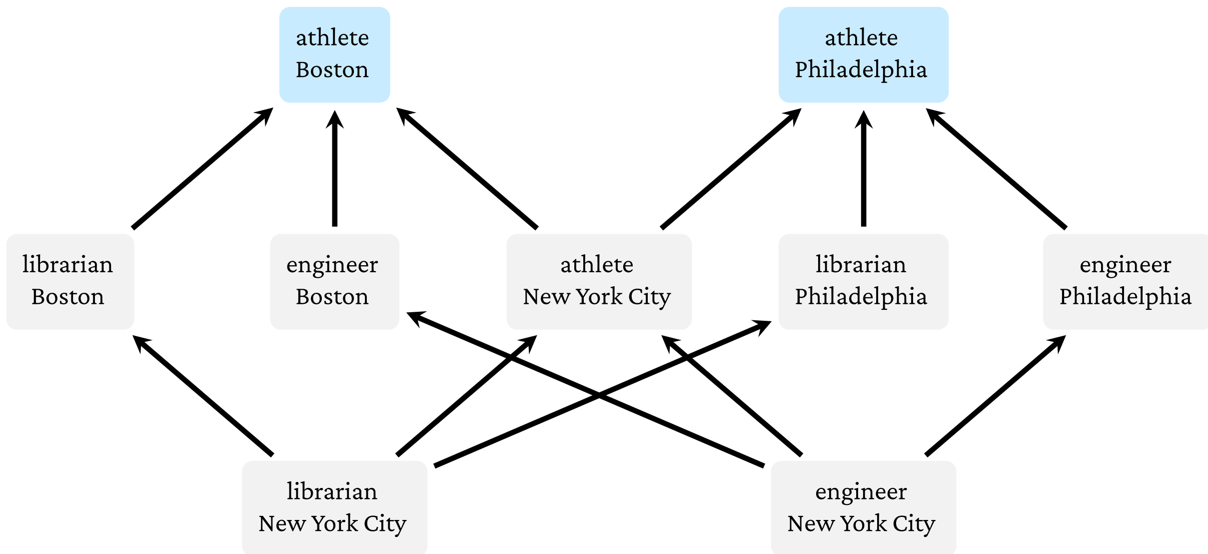


Figure 2.26: Jobs × Locations maximal antichain #1

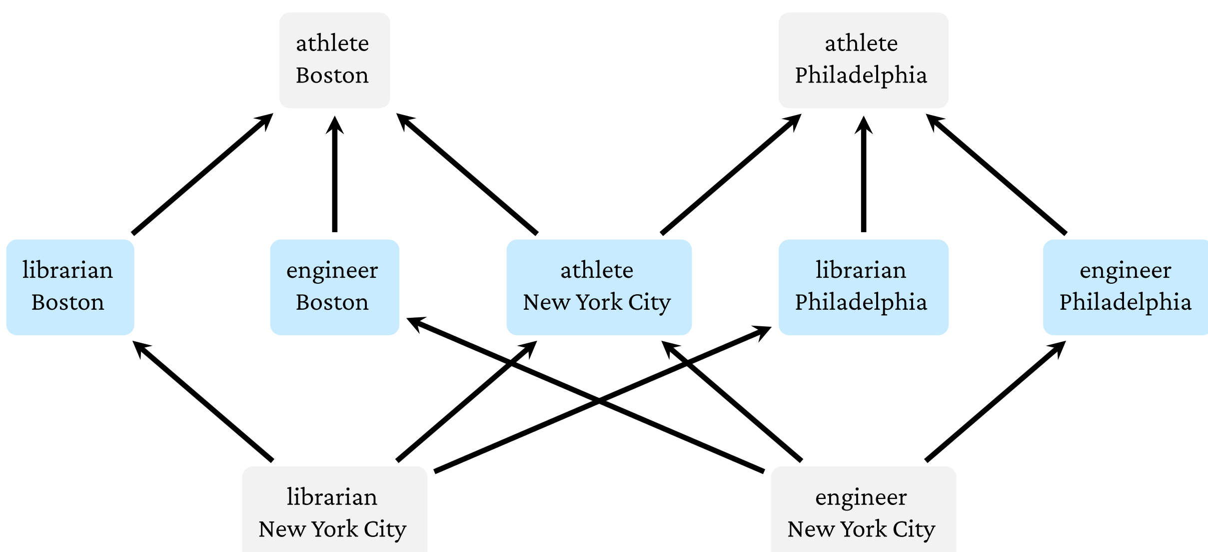


Figure 2.27: Jobs × Locations maximal antichain #2

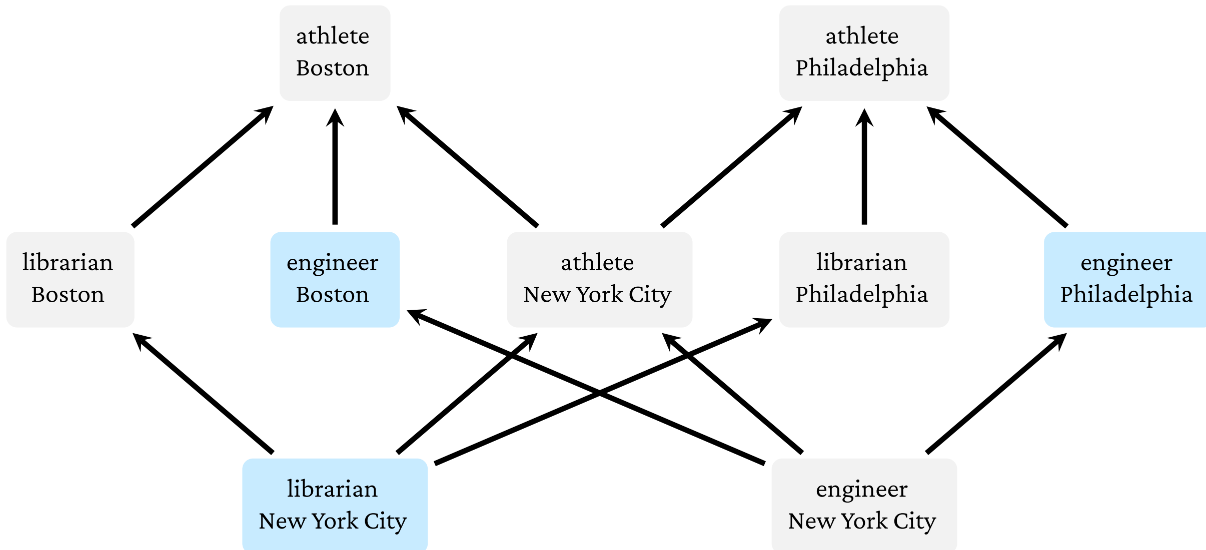


Figure 2.28: Jobs × Locations maximal antichain #3

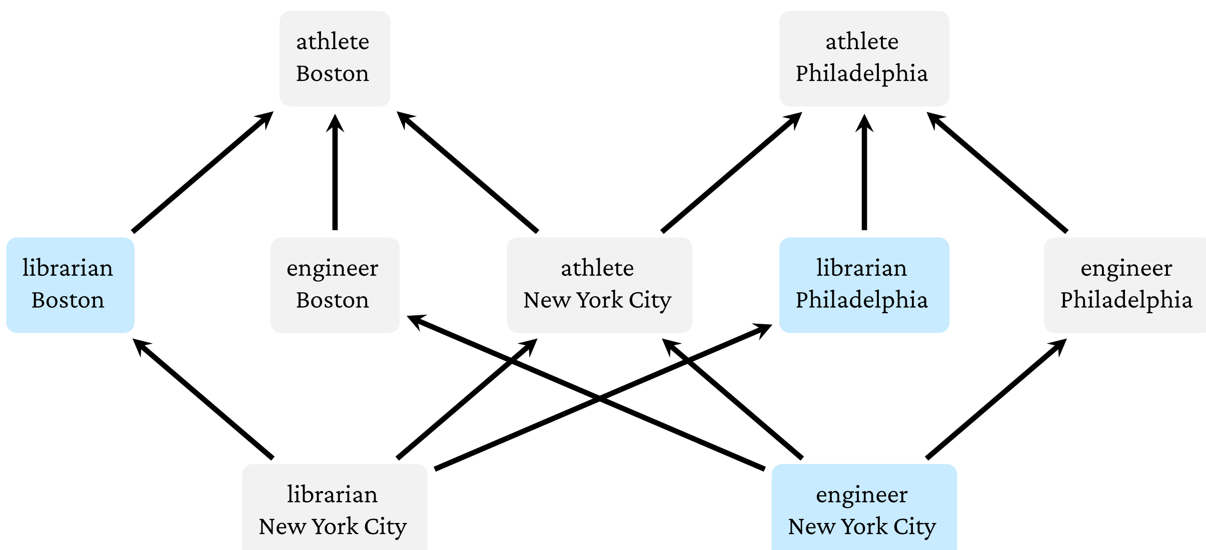


Figure 2.29: Jobs × Locations maximal antichain #4

In a *careset*, a maximal antichain can be thought of as a set of possible worlds where we don't know which is better. Because we are focused on maximal antichains, moving forward when we use the term antichain it can be assumed we are discussing a maximal antichain unless we specify otherwise.

If we have the option to be a librarian in Boston and an engineer in New York City, according to our initial Jobs and Locations *caresets*, we don't have enough information to know which one we will prefer.

The size of the largest possible antichain that can be constructed from a poset is called its width. We can consider the ratio of the width of the *careset* to the total number of elements in the *careset* to be a heuristic for how much uncertainty we have about what worlds we prefer. The second antichain above is the largest in the *careset* and has a size of 5, and there are a total of 9 elements.

These antichains are created naturally when we take the product of two *caresets*. But we may in fact have preferences within an antichain that have not been reflected in this new combined *careset*. Stating this preference could reduce our width and allow the *careset* to more accurately reflect our actual preferences.

We actually prefer the library in Boston to the library in Philadelphia, and we know in New York City we would rather be an engineer than a librarian. Let's update our *careset* accordingly.

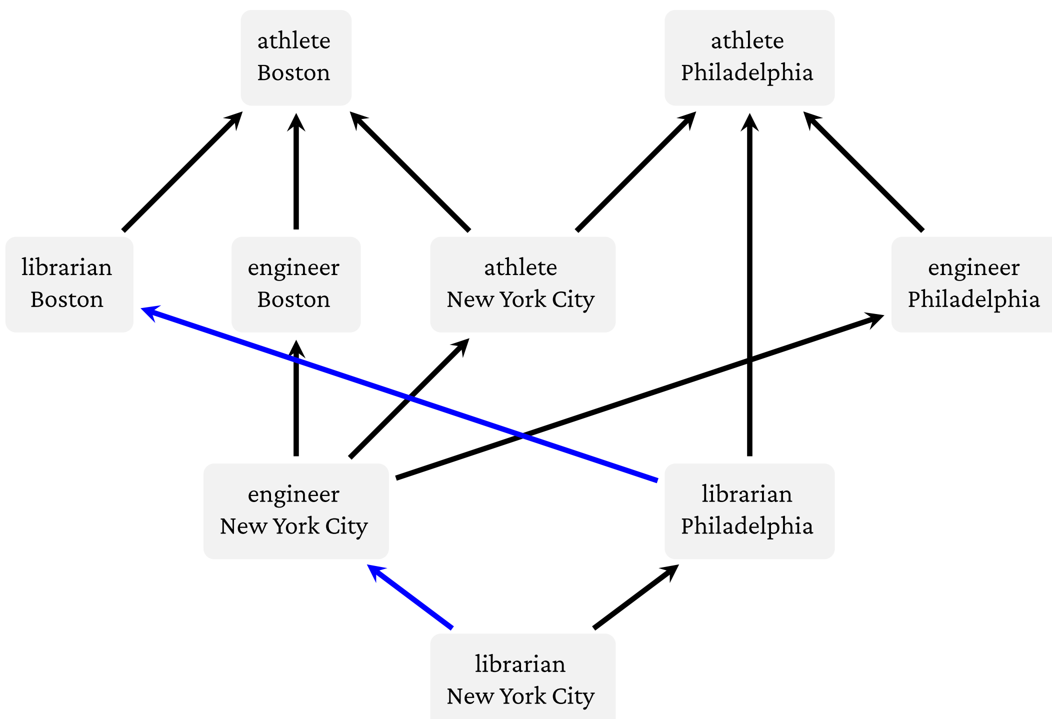


Figure 2.30: Jobs × Locations *careset* (new preferences in blue)

We have reduced the width of our *careset* to 4 and added a bit more resolution. Narrowing our product *caresets* is important so that our *caresets* represent our true preferences, and we reduce our uncertainty about which possible worlds are preferred.

2.3.3 Tradespaces

The first maximal antichain we looked at above is a non-dominated set of possible worlds, and is called a Pareto Frontier when visualized in a two-dimensional tradespace.

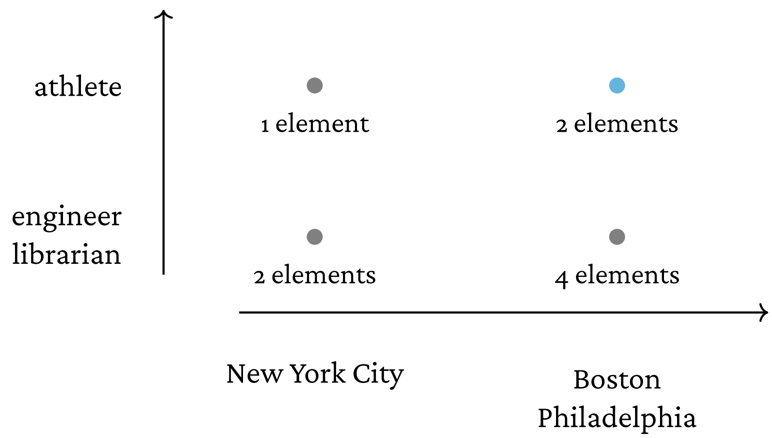


Figure 2.31: Jobs × Locations tradespace with Pareto frontier in blue

Tradespaces are used to show a set of options along two dimensions of criteria. The Pareto Frontier is the set of points that are not dominated by any other points. This means all options that lie on the Pareto Frontier are on an antichain.

Posets, and thus our *caresets*, can be made up of as many dimension as we want. We can create *caresets* that contain many different cares. A tradespace is a projection of a poset, and all our *caresets* can converted into a tradespace. If we want a two-dimensional tradespace, we can pick two cares from our combined *careset* and plot our antichain along those two dimensions.

2.3.4 Enriching *Caresets*

When we care about something, it is not always enough to say only that we prefer one world to another world. We may also need to specify just how strong that preference is. Consider a *careset* of possible dinners we may eat tonight.

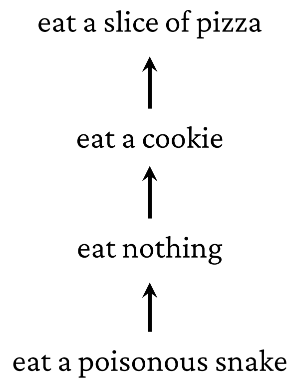


Figure 2.32: Dinners tonight

We may like cookies, but we may feel that one cookie is both not filling enough and not nutritious enough for dinner, so we may prefer a slice of pizza. A cookie is maybe better than nothing but just barely. But if we know anything, we know we definitely prefer eating nothing to having to try to eat a poisonous snake.

This *careset* does not communicate the magnitude of just how much we don't like the idea of eating the poisonous snake. We can enrich our *careset* to show this magnitude.

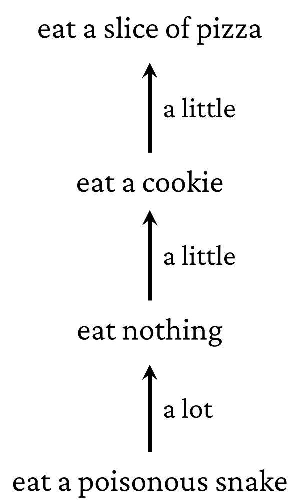


Figure 2.33: Dinners tonight, enriched with how much we prefer

Caresets can have the relationships enriched to be more precise about our care and preferences.

Mathematical enrichments must make sense as you move along the arrows. There must be a way to understand how to compose these enriched values.

To enforce this understanding, enrichment value must also come from posets, and those posets must be monoidal. We need to be able to compose elements of the poset with the monoidal operation and get back an element from the poset. This allows us to compose enriched relationships in our *careset* and still have the composed relationship be enriched by the same poset.

Enrichment of a Partially Ordered Set

Let $(P, \preceq)$ be a partially ordered set and let $(M, \otimes, e)$ be a monoid. An **enrichment** of P over M is a function

$$w : \{(x, y) \in P \times P \mid x \preceq y\} \to M$$

satisfying the following conditions:

1. **Identity:** For all $x \in P$, we have $w(x, x) = e$
2. **Composition:** For all $x, y, z \in P$ with $x \preceq y \preceq z$, we have

$$w(x, z) = w(x, y) \otimes w(y, z)$$

When we use *a lot* and *a little* to enrich our *dinners tonight* careset, we can declare the following operations to make sure our enrichment poset is explicitly monoidal.

$$\begin{aligned} \text{a little} \otimes \text{a little} &= \text{a lot} \\ \text{a little} \otimes \text{a lot} &= \text{a lot} \\ \text{a lot} \otimes \text{a lot} &= \text{a lot} \end{aligned}$$

This means that both *eat a cookie* and *eat a slice of pizza* are preferred to *eat a poisonous snake* by *a lot*.

We chose the phrases *a little* and *a lot* to allow us to hand-wave the magnitude of the difference of our preference. But we can see that caresets let us formalize that coarse quality with mathematical precision.

Quantitative posets can also be used for enrichment, with composition that is often more natural to reason about. Let's use pricing to indicate the magnitude of our preferences.

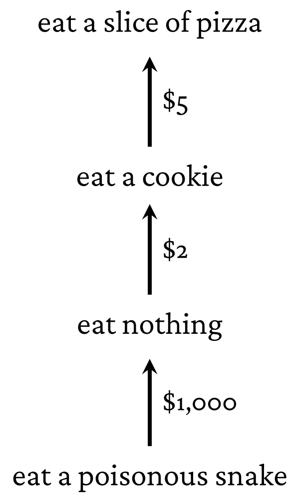


Figure 2.34: Dinners tonight, enriched with how much we would pay

According to the *caresets* above, we would be willing to pay $1,007 to move from the *eat a poisonous snake* to the *eat a slice of pizza* world for dinner tonight. *Eat nothing* is still a good deal.

These enrichments add detail while still being flexible enough to accurately represent the complexities of how we care. If we go back to a simpler meal decision from before, we can reevaluate our preferences for lunch by enriching our preferences with how much we would pay to move up the *careset*.

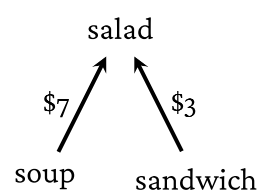


Figure 2.35: Lunch, enriched with how much we would pay

This *careset* adds more information about our lunch preferences, showing how much we would pay to move from one world to another. But there is no arrow between *soup* and *sandwich*. While the enrichment leads us to think that maybe we would prefer *sandwich* to *soup* because we are willing to pay more to move away from *soup* to *salad* than we are from *sandwich* to *salad*. But maybe we have not thought enough about whether we prefer a *soup* or a *salad*, or we would need more information to make a comparison. For whatever reason, we cannot say if we prefer *soup* or *sandwich*.

Our enrichments do not need to exhaustively order all our preferences – that would defeat the point of using a poset as our mathematical foundation. The *careset* allows us to speak with precision where it exists, and maintain uncertainty or incomparability where appropriate, even with enrichments.

2.3.5 Combining Enriched *Caresets*

We've changed our mind, a cookie is no longer a replacement for lunch, but a complement to it. Let's go back to our cost-enriched lunch *careset*, and also try enriching our cookie *careset* from before.

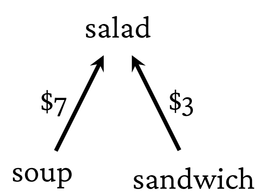


Figure 2.36: Lunch, enriched

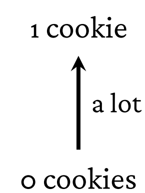


Figure 2.37: Cookie, enriched

Let's combine these two *caresets* to see how we feel about lunch with a cookie.

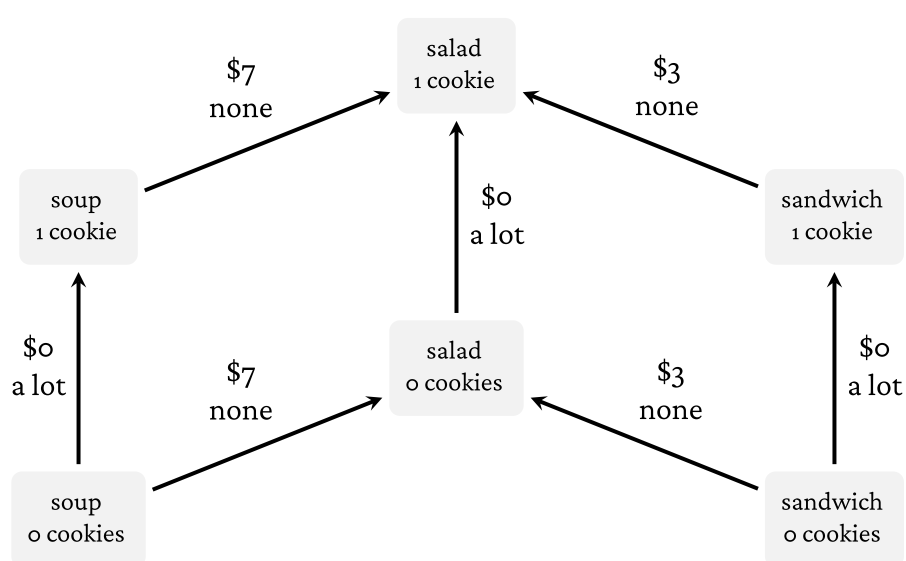


Figure 2.38: Lunch $\times$ cookie, enriched

We hint here at the power of posets, and the amount of precision and detail that is available to describe our preferences.

We could continue exploring if more forms of preference and decision theory can be represented as a *careset* in a satisfying way. This would be interesting and would require more mathematical analysis, questioning the monotonicity of product *caresets*, and searching for enrichments that would not be considered monoidal.

For now though, we will carry on.

Looking only at the tools we've built in our language so far, it will be useful to see if our language can support something stronger than preference, and maybe what we care about most: our ethical values.

2.4 A Few More Thoughts

We should address a few more thoughts before we go on to expand our language further.

2.4.1 Why not interval orders?

Interval orders treat all elements as acting like an interval on a number line.

Interval Order

A set P equipped with a binary relation $\preceq$ that can be represented by assigning to each element $x \in P$ a real interval $I_x = [a_x, b_x]$ such that:

- $x \preceq y$ if and only if $b_x \prec a_y$

That is, the interval of x lies entirely to the left of the interval of y.

It might be attractive to describe our preference for a possible world as existing within a range of preference, and then our relations as whether these ranges of preference overlap. This might allow for us to embed more uncertainty into how we view possible worlds.

But this also comes at a cognitive cost for minimal semantic gain. Posets already allow us to use incomparability as a way of describing uncertainty or "overlapping ranges". What would it mean to require that every possible world lives within a range of care?

Because interval orders require that elements live in intervals on some one dimensional line, they do not allow the $2 + 2$ formulation that posets do.

Figure 2.39: The $2 + 2$ Poset

It seems reasonable to want to be able to say that we prefer b to a, and we prefer d to c, but keep the two pairs incomparable.

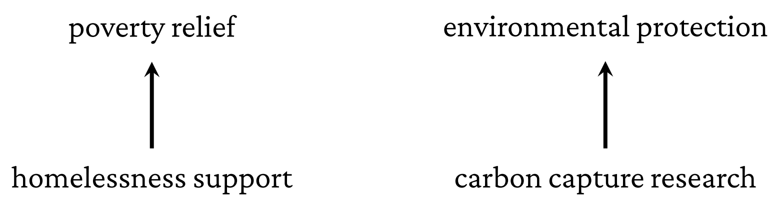


Figure 2.40: A $2 + 2$ charity donation *careset*

Imagine we can only donate to one of the four types of charities described in the *careset* above.

If this *careset* were an interval order, then we would have to resolve our preferences to a single dimension by drawing an arrow between the two related pairs.

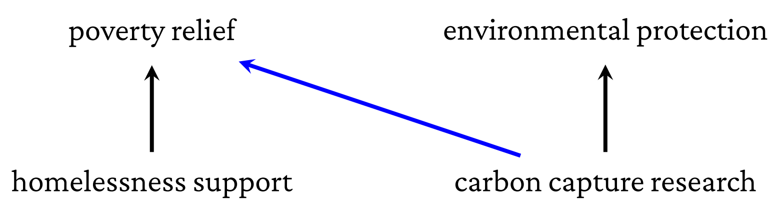


Figure 2.41: An interval order charity donation *careset* (new relation in blue)

This requirement that interval orders live in one dimension risks being too restrictive. We may not know how to think about our preferences between poverty-related charities and environment-related charities, and this fuzziness feels like an intuitive and natural way we might think about preference and care. We keep posets as the foundation for our *caresets* in order to natively support this kind of multidimensional care within a single *careset* and have our language of care support as many conceptions of care we can imagine.

There may be a case to be made that every individual *careset* should live in one dimension of care and the restriction provided by interval orders is useful. But we will leave that for a future exploration and take it as an excessive restriction for our current goals.

2.4.2 Why ordered sets at all?

One might also wonder why we should use ordered sets at all. We could consider graphs or other mathematical structures when considering building a mathematical language of care.

Ordered sets allows us to create structures that utilize transitivity, a fundamental concept in how we care. As we've already described, better and worse imply a transitive, relational idea of how we think of possible worlds. Graphs miss this crucial axiom, and when we make our directed acyclic graphs require transitivity, our graphs begin to look a lot like posets.

That being said, it would be an interesting exercise to examine what it would mean to describe care in the form of a graph rather than as a poset. But for our purposes, we will keep transitivity as a requirement and stick with orders.

This does not mean that all forms of care need to be relational or transitive, and posets allow us to say when we do not believe our care is transitive (see the $2 + 2$ example above). Ordered sets allows us to be expressive of these transitive relationships of preference alongside more complex ideas of care and preference, all while adding as little additional cognitive weight to our language as possible.

Using ordered sets, especially a structure like a poset, allows us to add significant mathematical expressiveness all on a very basic foundation. We will see later how we can take the product of two *caresets* because they are posets, and we can enrich them as well.

We have already mentioned that posets are also categories, and we will be able to open up our language of *caresets* to potential theorems from category theory. This has the potential to add even more expressiveness and discovery potential to our language that we will not be able to uncover today.

Posets strike a good balance between simplicity and expressiveness, and we will take them as the basic structures to describe our care. Let's see what we can do with our *caresets*.

2.4.3 On Social Choice and Decision Theory

Caresets will feel familiar to structures in social choice and decision theory, but we intentionally avoid committing to any specific relationships with those theories today.

We use posets and category-theoretic concepts like enrichment from the outset, and using our own terms gives us the freedom to build a language exactly for the goals we have in mind. This helps us maintain clarity and generality without having to specify how exactly to change existing social choice structures to work with the concepts we will build in.

That being said, our *caresets* are similar to preference orderings over alternatives and many concepts offered by Sen, Arrow, Fishburn, and Chang (Sen, 1970; Arrow, 1963; Fishburn, 1970; Chang, 2002). Likely, many of the concepts in our language will map neatly, maybe even isomorphically, to structures from social choice and decision theory.

But for now, we will avoid specifying how exactly our language would translate to these other theories, and leave that for future theorists.

Chapter 3

Ethics

3.1 Achieving

3.1.1 Actions

What we have not expressed yet in our language of *caresets* is the concept of an action. We have described changing between possible worlds, but this is not the same as an action. The possible worlds in our *caresets* are mutually exclusive and jointly exhaustive because they are meant to represent a set of options for how the world can be at a given moment. But when we take an action, we are not moving within a single *careset*. Instead, we are picking an element in the *careset* that represents the possible world that could result from our action.

An action can be thought of as a function from the "now" *careset* to the destination *careset*. The "now" *careset* is a *careset* we will use to represent the state of the world at the moment of the action. This will be a *careset* with only one element, since we only have one possible world that represents any given moment (as far as this language is concerned).

The action function then can be described as an arrow from the only element in the now *careset* to an element in the destination *careset*.

For lunch, we have three action functions available to us that we can visualize as three arrows.

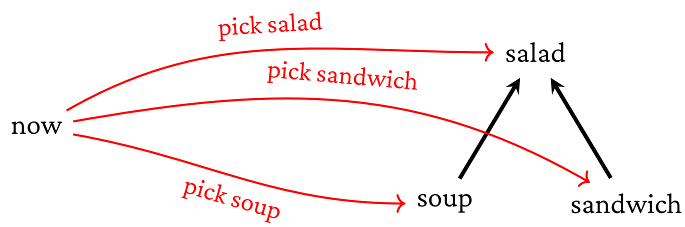


Figure 3.1: A lunch action

We can represent all possible functions from one *careset* to another using a different arrow.

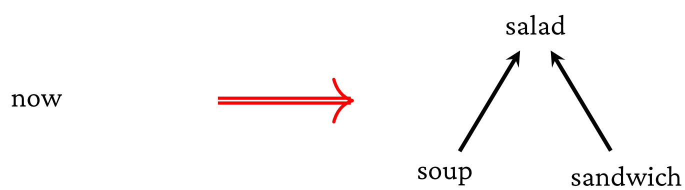


Figure 3.2: All possible lunch actions

We can also model the set of all actions that start from *caresets* other than the now *careset*.

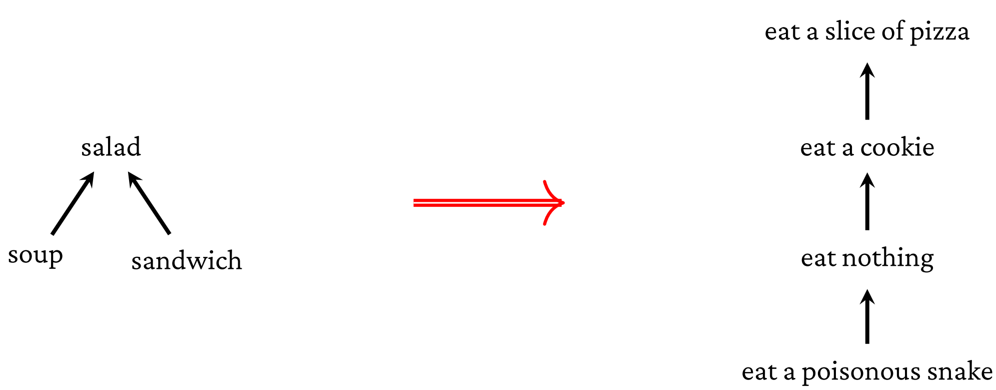


Figure 3.3: All possible lunch to dinner actions

Because our actions are functions, an action must have a destination for every possible world in the origin *careset*. When it's the *now careset*, this is simple because there is only one element. But when we model actions from *caresets* other than *now*, we are modeling an action taken for every given world we might be in in the origin *careset*.

3.1.2 Actions as *Caresets*

Representing actions as functions feels intuitive. We take an action in a given possible world and end in a possible world. But possible worlds are made up of everything in the world, including actions.

So what if we represented actions as a *careset*?

pick salad   pick sandwich   pick soup

Figure 3.4: Lunch actions *careset*

Because we are holding these actions independent from the worlds they realize, we will not make any preference judgments between them.

In fact, we can describe our lunch actions and our lunch together in a single *careset*.

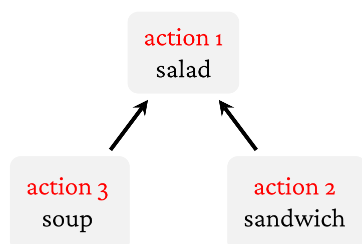


Figure 3.5: Lunch with lunch actions *careset*

Note that this is not the same as a product *careset*. Not all actions can be paired with all lunches, because an action picks a single lunch. This is simply a *careset* that represents the lunch action and the lunch together.

What this allows us to do is to ignore the *now careset* and look only at a single destination *careset* to decide which action is most preferred. Every *careset* that represents possible worlds also represents a set of equivalently preferred actions from the *now careset* to each possible world.

There may be multiple ways to get to a possible world, and they might not all be equivalent. In that scenario, we will need to examine the actions *careset* more closely.

Alternatively, certain possible worlds may be infeasible — there may be no actions that can get us to the possible world. We will consider this more in Chapter 4: Design.

For now, unless we specify otherwise, we will consider any *careset* to also represent its corresponding action *careset*. Namely, the *careset* that represents a mutually exclusive and jointly exhaustive set of equally preferable actions that realize each possible world in the *careset*.

So our lunch *careset* both represents the possible worlds that can be realized for lunch, and the actions we can take to realize those possible worlds.

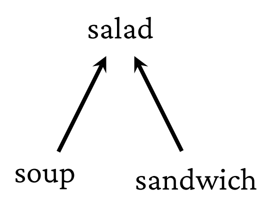


Figure 3.6: Lunch *careset*

3.1.3 Category of *Caresets* and Actions

Let's take a brief moment to propose a category *CARE*.

CARE

The category ***Care*** is defined as follows:

- **Objects:** *Caresets* $(C, \succ)$ where C is a finite set of *possible worlds* and $\succ$ is a partial order on C.
- **Morphisms:** Order-preserving functions $f : C \to D$ between *caresets*, called *actions*.
- **Composition:** Standard function composition $(g \circ f)(x) = g(f(x))$.
- **Identity:** The identity action (*do nothing*) $\mathrm{id}_C(x) = x$ for each *careset* C.

In ***Care***, the mutually exclusive and jointly exhaustive qualities of our *caresets* become important. By composing functions, we can, in theory, represent all possible chains of actions through a set of *caresets*.

There are many other aspects of this mathematical formulation of actions that could be explored in the future. How might probabilistic outcomes of actions be represented in ***Care***? Actions have only one possible world as outcomes, but actions are not taken with certainty about the outcomes. To reflect this, actions might map to probability distributions over possible worlds rather than single destinations. Could you enrich morphisms in ***Care*** with probabilities? Is ***Care*** a 2-category?

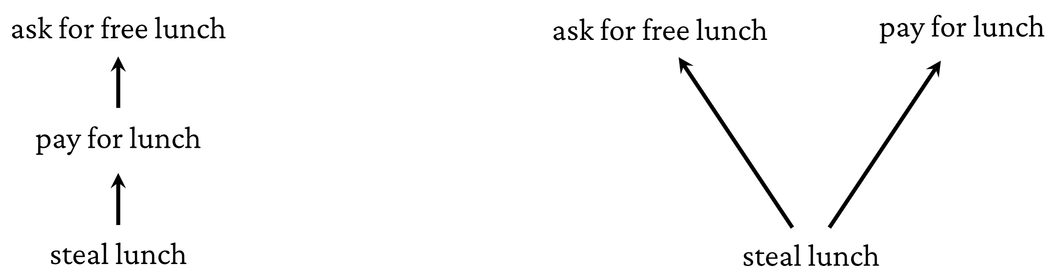
There may be theorems from category theory that can be applied to ***Care*** to see what they represent. We will not explore the ramifications of this category here, but we can share in the excitement nonetheless.

3.2 A Language of Ethics

3.2.1 Ethics as Better

It is very natural to use our ethical values to decide what possible worlds are better than others. We used *better* as the relationship in our *caresets* to prepare for exactly this kind of flexibility.

We now can think of two different kinds of *caresets*, ones that describe our preferences and ones that describe our ethical values. Much like two people can have different *caresets* with the same possible worlds, a preference *careset* and a value *careset* may also create different structures with the same set of possible worlds.



A possible preference *careset*　　　　A possible value *careset*

Figure 3.7: Two *caresets* for acquiring lunch

Alternatively, we could enrich our preference *caresets* with the *ought*. This could be a way to let our preferences and our values live side-by-side in the same *caresets*, allowing our language to make both *is* statements and *should* statements (Hume, 1739).

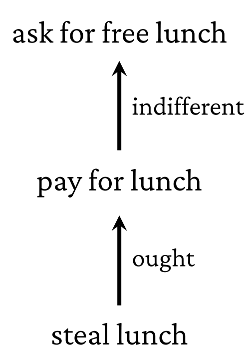


Figure 3.8: A preference *careset* enriched with *ought*

But our values are not always so obvious.

3.2.2 Trolley Problem

Let's consider this classical philosophical dilemma in the language of *caresets*.

> **The Trolley Problem**
>
> A runaway trolley is heading down a track towards five people who are tied up and unable to move. You are standing next to a lever that can divert the trolley onto another track, where there is one person tied up. You have two options:
>
> 1. Do nothing, in which case the trolley will continue on its current path and kill the five people.
> 2. Pull the lever, diverting the trolley onto the other track, where it will kill one person.

A *careset* of the possible outcomes would be quite simple.

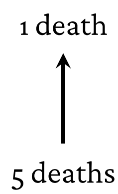


Figure 3.9: Trolley problem outcome *careset*

But the trolley problem gets at a deeper question. Possible worlds don't need to only be outcomes. We can represent the possible actions as possible worlds, and the actions we take to achieve these two worlds may not be morally equivalent.

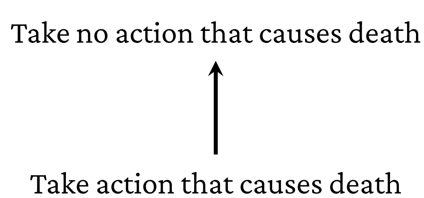


Figure 3.10: Trolley problem action *careset*

We can look at the wider space that is created by taking the product of these two *caresets*.

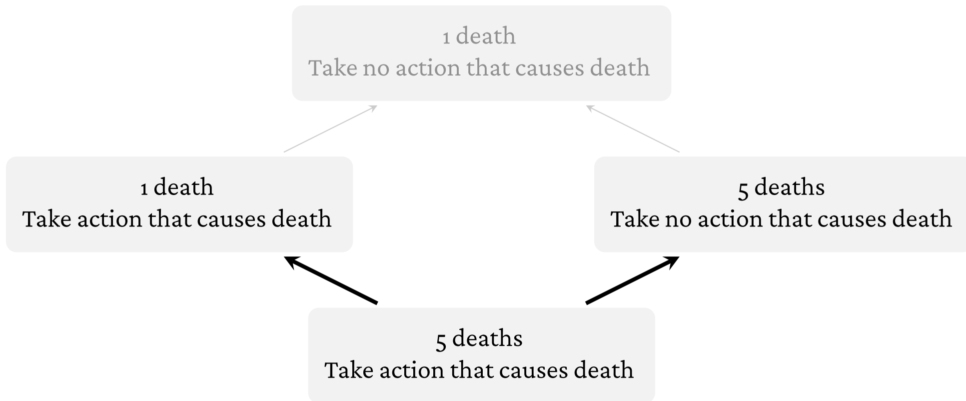


Figure 3.11: Trolley problem *careset* as product of two *caresets*

The best possible world is not feasible according to the formulation of the Trolley Problem, and herein lies our dilemma: our best options are in an antichain. When we put together the values as described in our *caresets* above, they do not provide a single best possible world for us to choose.

It is worth a moment to ask ourselves if we have successfully summarized the core Trolley Problem with two *caresets* and their product. That would be an interesting feat. We can at least say that we have produced a strong representation of why the problem creates a classical dilemma. But it is exactly the fact that there is disagreement about the structure of our values that the Trolley Problem houses a rich philosophical debate.

Maybe we do not believe the action *careset* accurately represents the situation. *take no action that causes death* is a controversial wording, are we really taking no action that causes death if we don't turn the lever? What is it to cause death? Possible worlds must be precise, because only through their precision can we draw *better* relations.

The pure act-consequentialist would throw out any concern for actions.

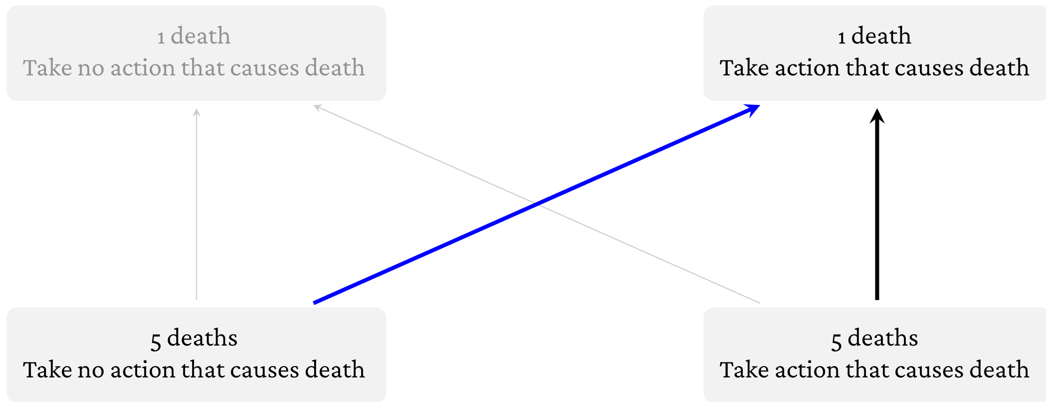


Figure 3.12: Pure consequentialist trolley problem *careset* (new relation in blue)

The consequentialist may generally value taking no action that causes death over taking action that causes death. But, given certainty of the outcomes, they will always prioritize the consequences over how those consequences were achieved (Smart and Williams, 1973).

A utilitarian would take the consequentialist *careset* and enrich it with some form of utility, say expected Quality-Adjusted Life Years (eQALYs) (Weinstein and Stason, 1977).

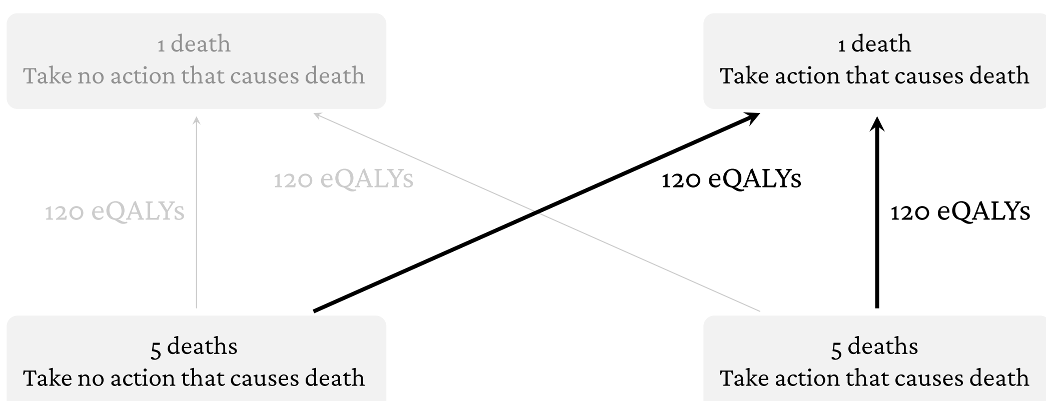


Figure 3.13: Utilitarian trolley problem *careset*

Meanwhile, a strict Kantian might restructure the whole *careset* according to the categorical imperative (Kant, 1785) and ignore the outcomes altogether.

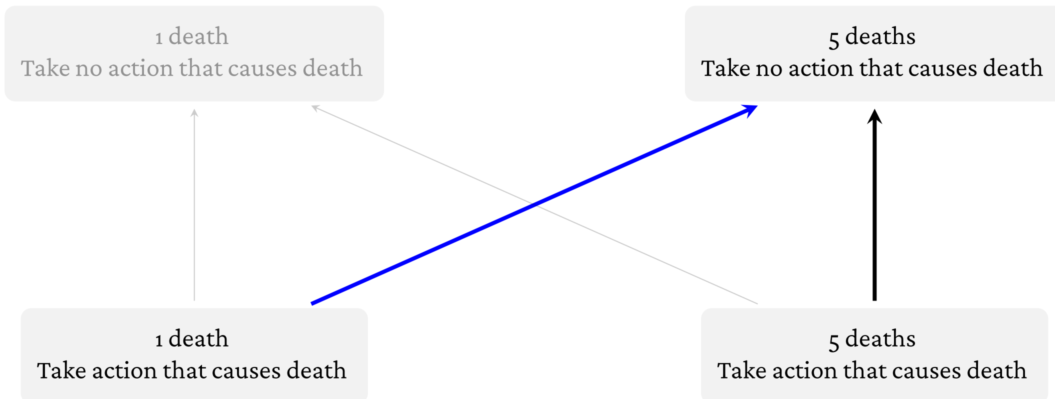


Figure 3.14: Categorical imperative trolley problem *careset* (new relation in blue)

A strict Kantian might in fact declare normativity – all actions are either good or bad.

3.2.3 Normativity

Not all ethics necessarily require a normative evaluation. Not all ethics declare a given action or a given possible world to either be good or bad. But we can very well represent those ethics that do.

The Kantian might declare that all actions either abide by the categorical imperative or they do not. And that alone makes them either permissible or not, either good or bad. We will not differentiate between ethically good and ethically permissible today.

In order to represent this binary of good and bad in a *careset*, we can imagine a function that maps all possible worlds in a *careset* to an old friend, the good-bad *careset*.

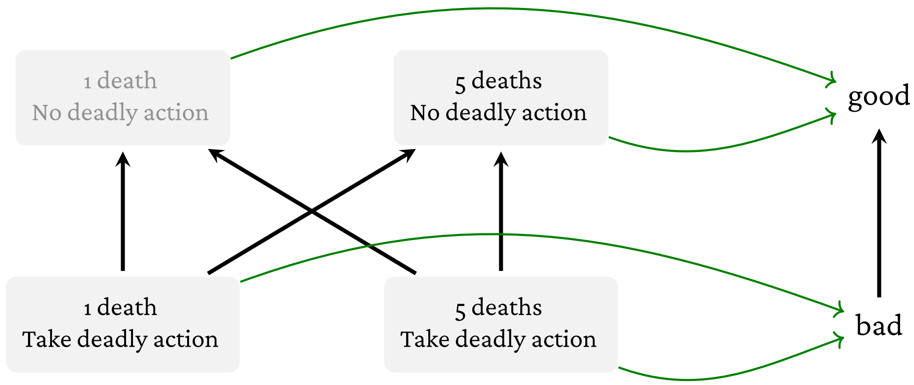


Figure 3.15: Categorical imperative normativity function on the trolley problem *careset*

Here we see that the two antichains of the Kantian trolley problem *careset* each map to a normative value. This is a quality of a *careset* fully defined by a strict, binary normative ethics – *caresets* created by that ethics will only ever have two antichains since there is no granularity other than good or bad.

Normative mappings do not need to only map to the good-bad *careset*. Any destination *careset* of a normative mapping can define a version of normativity. We can even support conceptions of normativity that are not total orders.

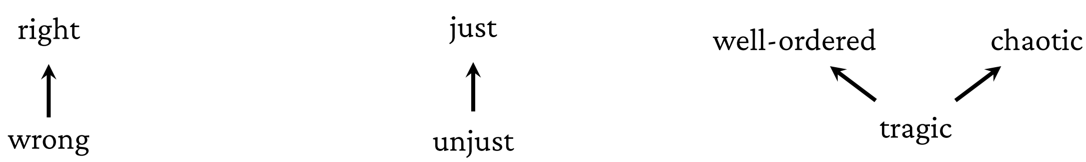


Figure 3.16: Candidate normative *caresets*

Normativity takes us back to the cliff of objectivity, and allows us to peer over the edge. Our *caresets* have so far seemed relational and subjective. But now we can create statements that make objective claims. *Caresets* do not prove objective facts, but they let you speak them.

With this flexibility, *caresets* seem capable of representing many kinds of ethical structure.

3.2.4 Ethics of Actions

We see in the Kantian *careset* that actions can matter significantly, and possible worlds that describe our actions can, at times, matter more than the possible worlds that describe outcomes.

Care ethics seems an appropriate example to consider. Care can be thought of as something that is enacted or realized, not only something that motivates or assesses our actions (Noddings, 1984).

If care is realized, then it should also be possible to describe with a *careset*. And if ethical actions are ones that bring about more care, then we should be able to point to the possible worlds with more or less care, and condone those actions that do better.

Care itself may be a form of normativity (Held, 2006).

Caresets give us a unifying structure to point to in discussions of ethics. There seems to be some hope that many ethics, value systems, and deontological theories can be represented as some structure or enrichment of possible worlds and actions. If this is so, then *caresets* may be a step towards a mathematical language of ethics.

Chapter 4

Design

4.1 What is design?

4.1.1 Design in *Caresets*

A design process takes into account multiple qualities, as many qualities as it can that are relevant for the system that is being designed. These qualities are the qualities that the designer cares about. Designing a building may include cares for function, form, cost, materials, and resilience. Designing a meeting agenda may include cares for length, engagement, productivity, and comfort.

Whether you are designing a building or designing a meeting, a non-trivial design considers multiple qualities of the system that is being designed.

If each quality of a design can be described as a care with a corresponding *careset*, then we can describe the possible outcomes of our design as a product *careset* of all those qualities.

Event Space
↑
Digital Media
↑
Books

Figure 4.1: Function *careset* for a library (assume each higher element contains all lower functionality)

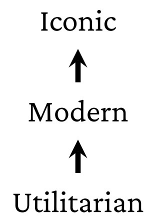


Figure 4.2: Form *careset* for a library

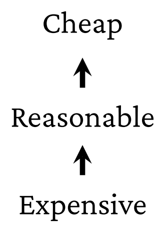


Figure 4.3: Cost *careset* for a library

When we put it all together, we can see the Library design space as a product *careset*.

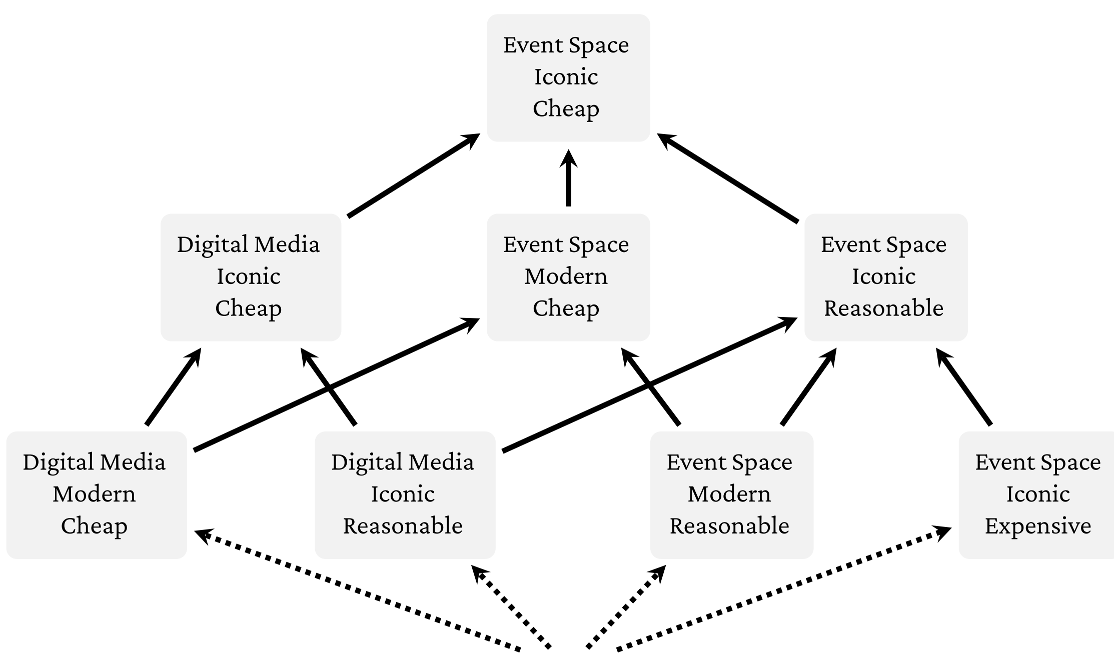


Figure 4.4: Top of the product *careset* for a library

This idea of design is a more general one than the term usually connotes in traditional design disciplines such as architecture, industrial design, and product design. Our idea of design does contain the more traditional concept of design – architecture is very much about negotiating between multiple qualities like those described in the *careset* above. But our more general conception misses the specific focus on aesthetics, style, psychology, innovation, and philosophy that more traditional design fields tend to carry with them. We respect the different uses of the term design and acknowledge our participation in the usurpation of the term from these traditional design fields. Let this be our semantic land acknowledgment.

For our purposes today, we will define design as devising "courses of action aimed at changing existing situations into preferred ones" (Simon, 1988). In our language, this translates to moving up antichains in *caresets*.

4.1.2 Moving Up Antichains

The purpose of design is to find the best antichain we can achieve in our *caresets*. As we know, we may not have a clear preference between some worlds, and that means those worlds will live in an antichain together. Design is the work of finding ways to achieve better worlds, and thus it is the work of discovering higher antichains that are feasible.

This conception of design fits very naturally with the idea that design is about negotiating different qualities. If a trivial design is only concerned about one quality, then we will not have a trade-off and our antichains will have only one possible world. We will only care about making one quality the best it can possibly be.

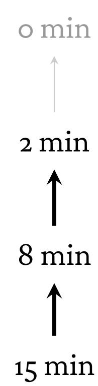


Figure 4.5: Transit frequency *careset* with infeasible possible worlds grayed out

A possible world may be infeasible either because it is impossible to achieve, or we do not know any way to achieve it. Zero-minute frequency is not possible thanks to a cantankerous spacetime.

If we have successfully included all the relevant cares of our system in our *careset*, and our *careset* is a total order, then all that is left to do is to go as high in our *careset* as we can. Every possible world is a one-element antichain, so we have no uncertainty about which worlds we prefer.

But this is a trivial version of a design space. A very coarse but more realistic version of this *careset* would look something like this.

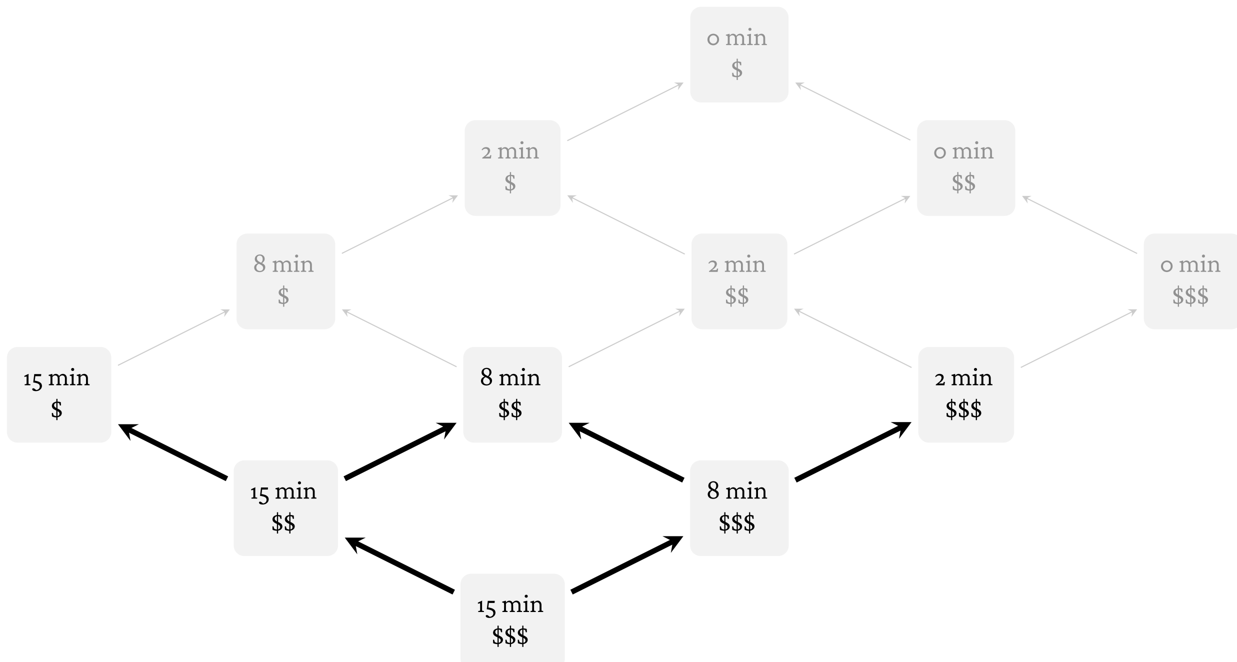


Figure 4.6: Transit frequency × cost design space *careset*

Real problems have wicked trade-offs (Rittel and Webber, 1973), so we know that there will be multiple qualities that we care about and it will be practically impossible to achieve everything we want from all of them. This is why we have any hope that posets can represent the things we care about, and why the antichains in the products of these posets are the goals of our design process. Design is discovering greater feasible antichains within a *careset*.

We can see this when we look at two antichains from our Library design space.

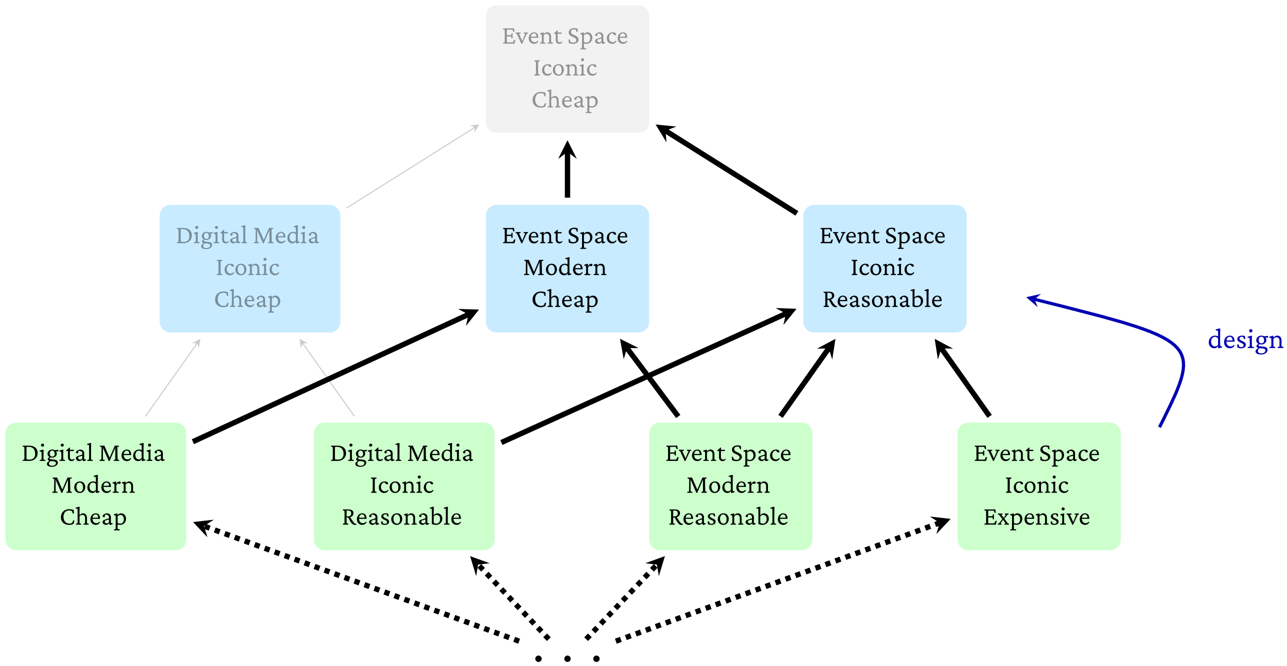


Figure 4.7: Library design space *careset* with two antichains highlighted

These are not the only two antichains that involve these elements, but you can see where design wants to go.

4.2 Designing with *Caresets*

4.2.1 Narrowing *Caresets*

Design might also want to make our *caresets* more narrow. It is usually not the case that everything we care about should be weighed equally. This means that a *careset* that is created by combining many different cares will be a wide *careset* with long antichains. A good design process should help us make decisions about what qualities matter more.

A good design process may choose to present an opinion or an ethic that dictates some worlds to be better than others.

If we know we care more about our library having an event space than we do about it being cheap or iconic, part of design's role is to reveal these preferences. When revealed, we can add new arrows that narrow our *careset*.

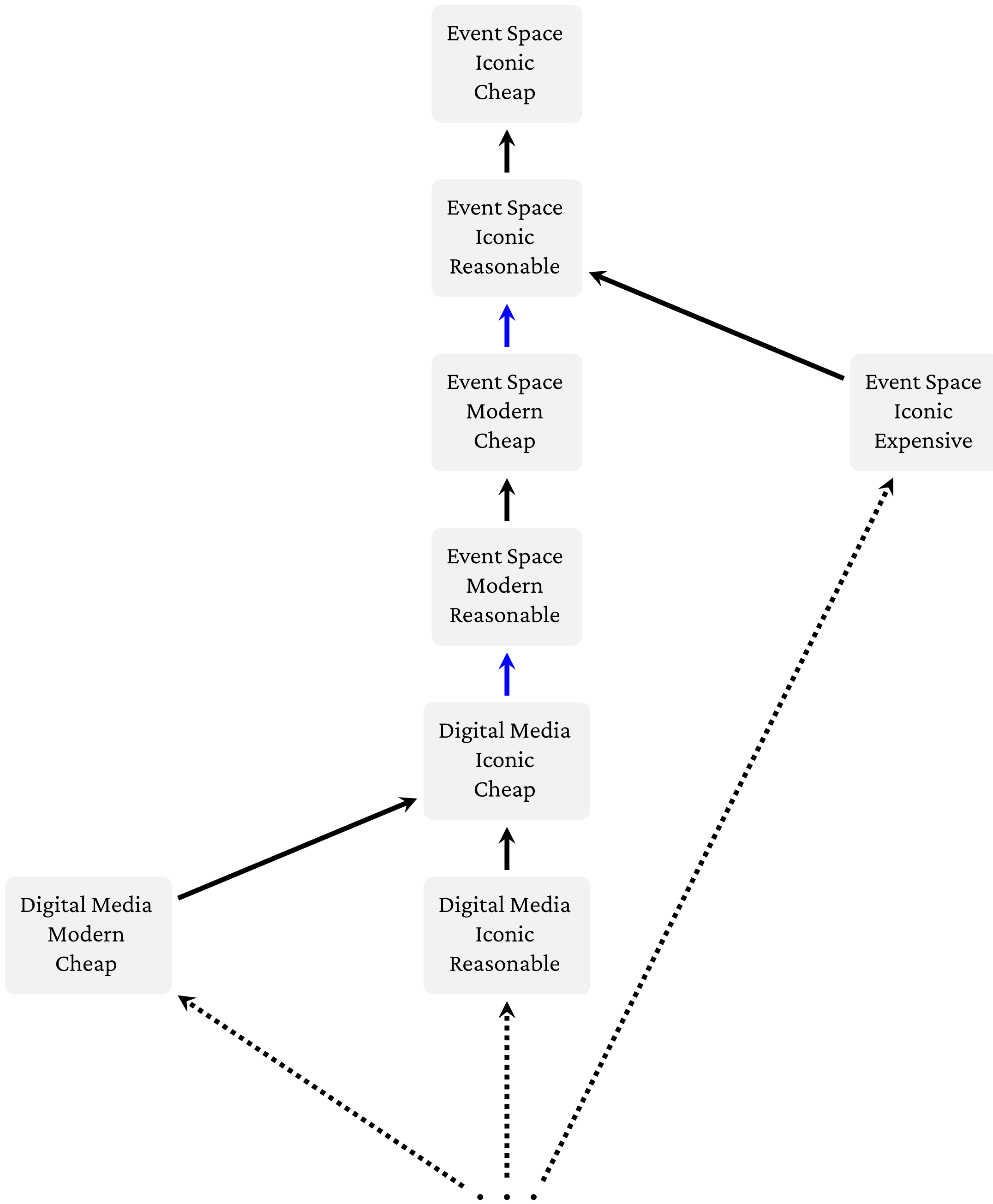


Figure 4.8: Top of library design *careset* (new preferences in blue)

Reflecting this revealed preference in our *careset* has significantly narrowed the top of the *careset*. The antichains have become shorter, meaning the designer has discerned which trade-offs are worth making.

4.2.2 Feasibility

As we hinted at above in our transit *caresets*, all design work is done with a respect for feasibility. Before, we have considered *caresets* in the abstract, imagining all the possible worlds to be worlds that are possible. With the goal of realizing these worlds, design must now consider feasibility within *caresets*.

If a global maximum possible world is feasible, then the careset contains no real trade-off – all preferences can be satisfied and the designer's goal should be to achieve that global maximum.

More likely, the tops of our *caresets* will not be feasible, and the tops of our *caresets* will be out of reach. This is the very nature of *caresets* that represent real trade-offs.

4.2.3 Imagining Better Worlds

But I hope you are not happy that we chopped off the tops of our *caresets*. We claim to care but we did quickly amputate our best possible worlds.

Caresets make explicit how high we believe we can go, and they make us acknowledge how far we are from where we wish we could be.

It is not particularly novel to ask a design process to admit that it cannot achieve a perfect solution, or to acknowledge that trade-offs need to be made. But when we are imagining all possible worlds as a *careset*, we are made to visualize the entire design space holistically and consider improvements we may not have considered.

Unless you are at a local maximum, a *careset* will always show you better possible worlds. Because the *careset* is all possible worlds involving the cares in our scope, any justified solution must show why it cannot be moved to realize the nearest best possible world. Tell me why I can't keep taking arrows upward.

If *caresets* can be an intuitive mechanism for communicating about nearest possible better worlds, they may also be a tool for inspiration and imagination.

There is always a risk that a design process will become mired or lose sight of the bigger picture. Very often a design process will start from an engineering solution and move to determine what possible worlds this solution can help realize. Starting from the *careset*, and using it as a north star that guides any design process, may lead to more creative and value-driven design.

The scope of a *careset* may be large, spanning what would normally be covered by many different disciplines and types of expertise. Decomposing *caresets* can be valuable, but wide-scoped *caresets* provide a value of their own. A designer may seek ways to improve a given sub-scope of a *careset*, and that scope may contain the standard trade-offs of the solutions the designer is considering. But when the designer is exposed to a larger scope, new possibilities for movements towards better worlds may be revealed.

The hope is that *caresets* can inspire us to imagine upward movements that we might not have considered before leading us to better worlds than we had originally thought feasible.

tomorrow

↑

today

Part Two

The Experimental Public
Co-Design of Tomorrow

Chapter 5

From Language to Framework

We now have some reason to believe that preference, ethics, and design can be formalized through partially ordered sets. This mathematical language of care gives us hope that monotone co-design can represent problems of care as well as it does engineering design problems.

Can we use our formalism of care to help make monotone co-design more expressive of community cares? Can we make monotone co-design participatory?

We present the Experimental Public Co-Design of Tomorrow (EPCODOT) as an attempt to create a collaborative interface that leverages the compositional and computational capabilities of monotone co-design.

The EPCODOT framework has a companion software prototype, available at epcodot.com. But the diagrams used to describe the user interactions are not necessarily exact representations of how epcodot.com, or any other implementation of this framework, must choose to build out these interactions. Here we attempt to provide a canonical description of the framework that is agnostic to any specific software implementation.

We build up this new framework over the coming chapters, showing how we can adapt monotone co-design and how our framework lets users compose, compute, and collaborate. We will conclude with two case studies that show how EPCODOT can be used to model trade-offs in urban contexts and be a tool for participatory planning.

Chapter 6

Co-Design and Care

6.1 Monotone Co-Design

Monotone co-design is a framework for engineering design that breaks down systems into co-design problems that each represent a monotonic relationship between a poset of resources and a poset of functionalities, and composes those co-design problems to represent more complex systems (Censi, 2015; Zardini et al., 2023).

Co-design problems are represented as a box with one or more resource wires and one or more functionality wires.

Figure 6.1: The basic elements of a co-design problem

When we provide co-design problems with implementations and compose them together, they represent complex systems. These systems have complex interactions, like feedback loops, making it difficult to determine which implementations maximize functionality and minimize resources. The monotone co-design problem (mcdp) solver uses these implementations to compute Pareto-optimal solutions for these complex, composed systems.

6.1.1 Implementing Co-Design Problems

Any system that uses a resource to provide a functionality monotonically can be represented as a co-design problem. Any implementation for that co-design problem must describe a functionality provided for a resource required. This can be done either with a formula or with a set of implementations that connect a specific element in the resources poset to a specific element in the functionality poset.

Consider a co-design problem that represents the process of purchasing materials to build a warehouse.

Figure 6.2: Purchase materials co-design problem

We could implement the *purchase materials* co-design problem with a simple formula that includes the cost of the materials.

$$(\text{steel} \times \$800/\text{ton}) + (\text{concrete} \times \$150/\text{ton}) \leq \text{cost}$$

Or, consider a related co-design problem that uses those materials to build a warehouse.

Figure 6.3: Warehouse co-design problem

Then imagine we have four warehouse designs that can be used as implementations for the *build warehouse* co-design problem.

Name	Floor area (ft^2)	Height (ft)	Steel (tons)	Concrete (tons)
Compact Storage Warehouse	5,000	15	20	30
Standard Distribution Warehouse	15,000	25	45	80
Wide-Span Warehouse	20,000	20	40	100
High-Capacity Logistics Center	40,000	40	95	200

Table 6.1: Warehouse designs

These are feasible implementations for the co-design problem. Given a certain amount of steel and concrete provided to our *build warehouse* co-design problem, we can use these designs to determine what floor area and height are feasible. In monotone co-design, the relationship between the resources and the functionalities must be monotonic.

Monotonic means that if an implementation uses a certain amount of resources to provide a certain amount of functionality, then it can always use more resources to provide the same functionality or provide less functionality using the same resources.

Take the Wide-Span Warehouse implementation. To be feasible, it needs at least 40 tons of steel and 100 tons of concrete, and in turn it provides 20,000 ft^2 of floor area and 20 ft of height. If we provide more steel, say 50 tons, we can still build the Wide-Span Warehouse and this co-design problem can still provide 20,000 ft^2 of floor area and 20 ft of height. If we need less floor area, say 18,000 ft^2, we can still build the Wide-Span Warehouse to provide that, and this co-design problem can still require 40 tons of steel and 100 tons of concrete.

This monotonicity is how functionalities and resources are defined in co-design. A functionality is something where requiring less never reduces the feasible implementations, and a resource is something where providing more never reduces the feasible implementations.

6.1.2 Composing Co-Design Problems

When co-design problems are composed, and resources and functionalities are wired together, we can begin to model complex systems.

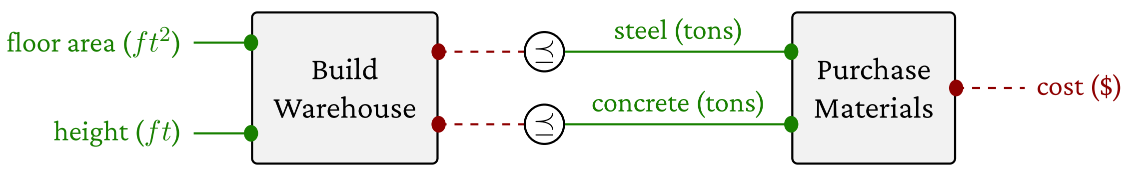


Figure 6.4: Composing the warehouse co-design problem

Every co-design problem can only use resources that are $\leq$ to the functionality provided by a connected co-design problem.

As you connect co-design problems, you create new, composed co-design problems that describe a higher level of abstraction.

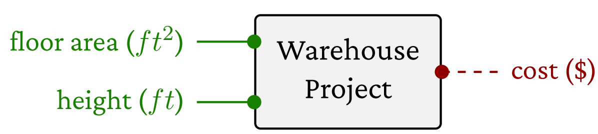


Figure 6.5: Warehouse co-design problem

The composable nature of co-design lets you navigate up and down layers of abstraction within the same framework, with each co-design problem solving a discretely scoped relationship of functionalities and resources. We can compose up to represent larger and more complex systems or decompose down to solve smaller and more detailed problems.

6.1.3 Posets

Posets are the primitives used in monotone co-design – they describe what a system provides and what it requires. We can think of a functionality like speed, where the poset would be real numbers ≥ 0 representing kilometers traveled per hour, or a resource like fuel where the poset would be real numbers ≥ 0 representing liters.

But we've seen that posets can represent more than just numerical values, consumable resources, and quantifiable functionalities. The *caresets* we described in Chapter 1: Care can express our preferences, our values, and our design goals.

Does monotone co-design work with *caresets*? Is there a difference between the posets in monotone co-design and *caresets*?

If we can use monotone co-design to build systems with *caresets*, can we use it to represent and run computations over more than just engineering systems? Could monotone co-design be used to represent and compute the impact of systems on the public good?

6.2 Co-Design Problems Live in *Caresets*

We can now examine how well the poset structures in monotone co-design align with the poset structures in our *caresets* from Part One: A Mathematical Language of Care.

Let's take another look at our warehouse from before.

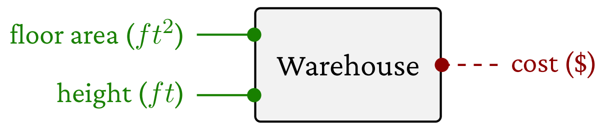


Figure 6.6: Warehouse co-design problem

Floor area, height, and total cost are the relevant posets for this co-design problem. But do they have anything to do with care? Do we care about them?

At first, it might be difficult to say if care is the right word to use here. If I am the business owner, I might not care about these values in particular. Maybe I care more about what a warehouse like this unlocks for our business, and what the opportunity cost and feasibility are for building this warehouse.

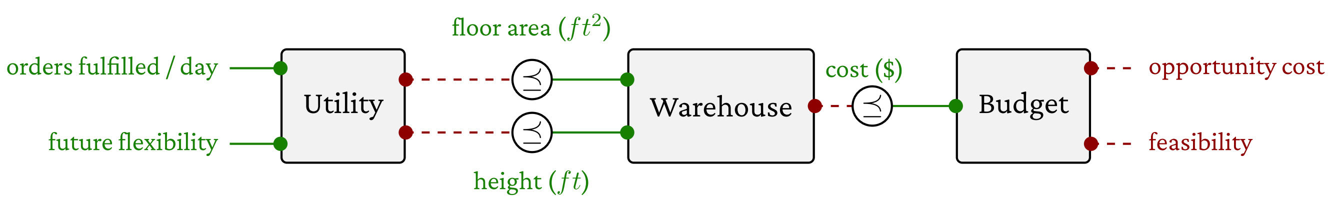


Figure 6.7: Connecting the warehouse co-design problem to business priorities

Now we can wrap this in a new co-design problem.

Figure 6.8: Co-design problem for overall warehouse business opportunity

Maybe these posets are closer to things the business owner cares about. We can imagine continuing this composition up. Orders fulfilled per day only matters in so much as it helps please customers, bring in revenue, satisfy shareholders, grow the business, make the business owner proud, etc.

If we cannot get to something we care about by composing our co-design problems up layers of abstraction, then we must say that we do not care about the co-design problem. If we are able to get to something we care about by composing our co-design problem, we must say that, at least in some instrumental way, we do care about the co-design problem.

If we care about a co-design problem, that means we believe certain states of functionalities and resources are better than others. Can the elements of functionalities and resources be represented as mutually exclusive and jointly exhaustive sets of possible worlds?

If we look at the above examples of functionalities and resources, no matter the level of abstraction we can always understand each functionality and resource in terms of possible worlds. The 20,000 ft^2 element in the floor area poset represents a possible world where the warehouse has a floor area of 20,000 ft^2. The 5,000 element in the orders fulfilled per day poset represents a possible world where the business opportunity provides capacity to fulfill 5,000 orders per day.

We can imagine elements for non-numeric functionalities and resources: a possible world where the business has great future flexibility, or where the opportunity cost is reduced marketing spend. We can then require that every functionality or resource include all mutually exclusive and jointly exhaustive possible worlds relevant to the scope of the functionality or resource.

Mutual exclusivity comes naturally for functionalities and resources – the provided functionality or utilized resource must be a discrete value from the respective poset.

For joint exhaustiveness, we have to make sure the functionality or resource covers all possible scenarios. For example, if we are describing the opportunity cost, we must have elements that cover all possible opportunity costs that are within the scope of the business opportunity. At worst, we have to add a catch-all fallback element to cover joint exhaustiveness.

We have used our own term *possible world* to give ourselves the freedom to describe states of the world as generally as possible. Any functionality or resource can be described as possible worlds, and they are already structured like posets. Now once we require mutual exclusivity and joint exhaustiveness, every functionality and resource poset becomes equivalent to a *careset*.

The posets that connect co-design problems are *caresets*. When you take the product of all the functionality and resource *caresets*, you get the *careset* that represents the overall co-design problem.

We can see a snippet of the *careset* that makes up the warehouse co-design problem.

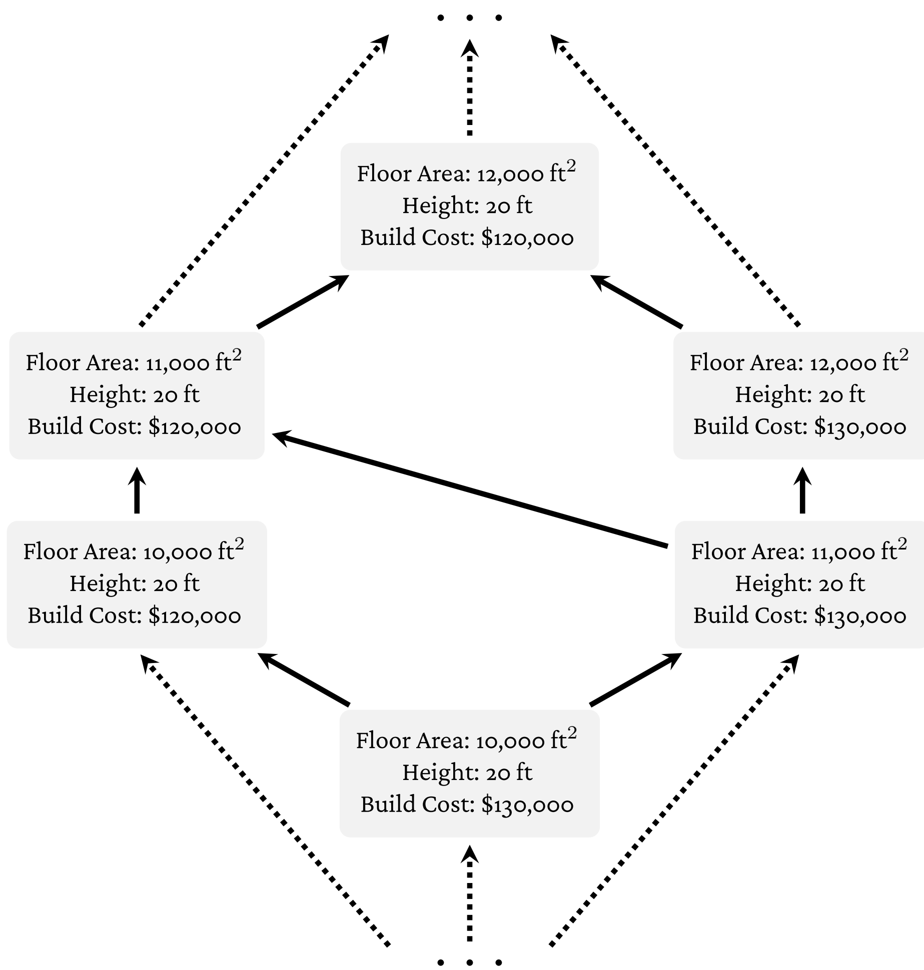


Figure 6.9: Warehouse *careset* showing Floor Area × Height × Total Cost

6.3 Co-Design Moves Up Antichains

We can think of functionalities as *caresets* we want to maximize in the context of the co-design problem, and resources as *caresets* we want to minimize in the context of the co-design problem. In the context of the co-design problem, the best possible worlds are the ones with the most functionality and the least resources.

This is not to say that we care about maximizing every functionality and minimizing every resource intrinsically. In fact, the functionality of one co-design problem often becomes the resource of another, and the care we wanted to maximize as a functionality in one context becomes the care we want to minimize in another context. This makes sense as long as those cares are instrumental to other cares that we actually value intrinsically. The highest level co-design problem we draw essentially determines the intrinsic cares we want to maximize or minimize globally.

We can see this in the case of the business owner. They do not inherently prefer having 100 tons of steel to having 50 tons of steel in and of itself. But in the scope of the build warehouse co-design problem, having more steel can only increase the options for what you can build. So for that scope, more steel is better because it might help achieve a better warehouse, which may in turn help achieve a better world for the customers, the business, the owner, or whatever is intrinsic to the scope.

But we usually will not be able to achieve the worlds where we maximize functionalities and minimize resources – the implementations of the co-design problems will determine which possible worlds are actually feasible.

When we discussed design in Chapter 4: Design, we saw that design moves up antichains in *caresets*. This concept translates directly to monotone co-design. If a co-design problem lives in a *careset*, then the mcdp solver is working to find the highest antichain in the representative *careset*.

Let's reconvene at the library.

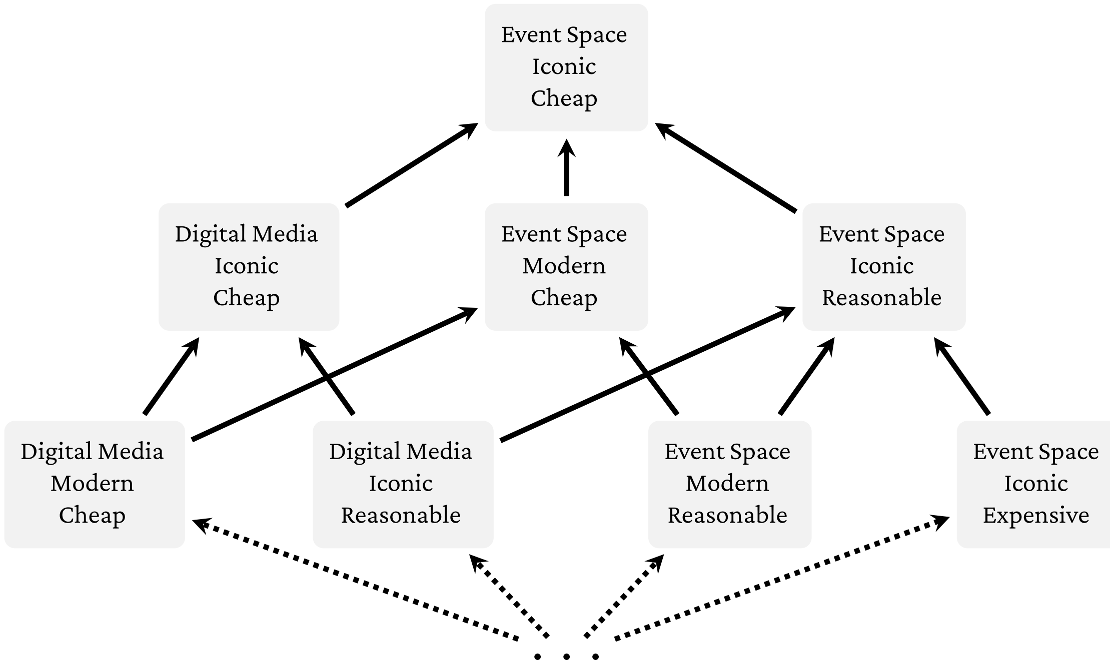


Figure 6.10: Top of the product *careset* for a library

We can now map this to a co-design problem. We want function and form to be maximized so we make them functionalities, and we know that more cost can get us more of both functionalities, so we make it a resource.

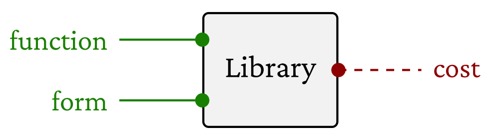


Figure 6.11: Library co-design problem

Let's say we've had a number of proposals submitted that all seem equally feasible. When those proposals are placed in the co-design problem as implementations, we can see our *careset* updated for which possible worlds are actually feasible from the given proposals.

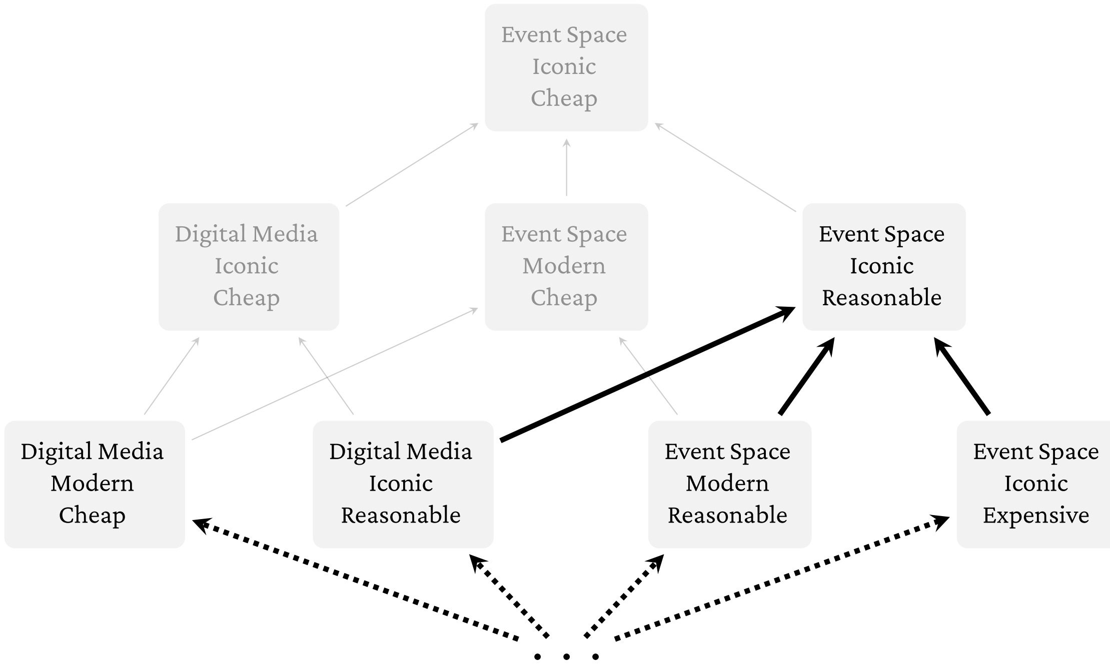


Figure 6.12: Library *careset* with feasibility

The mcdp solver would then be asked to find the best antichain in this *careset* that is still feasible. We fix the cost to reasonable (or less) and aim to maximize functionality. There may be more, but we know there are at least two outcomes that are feasible and belong in the best feasible antichain.

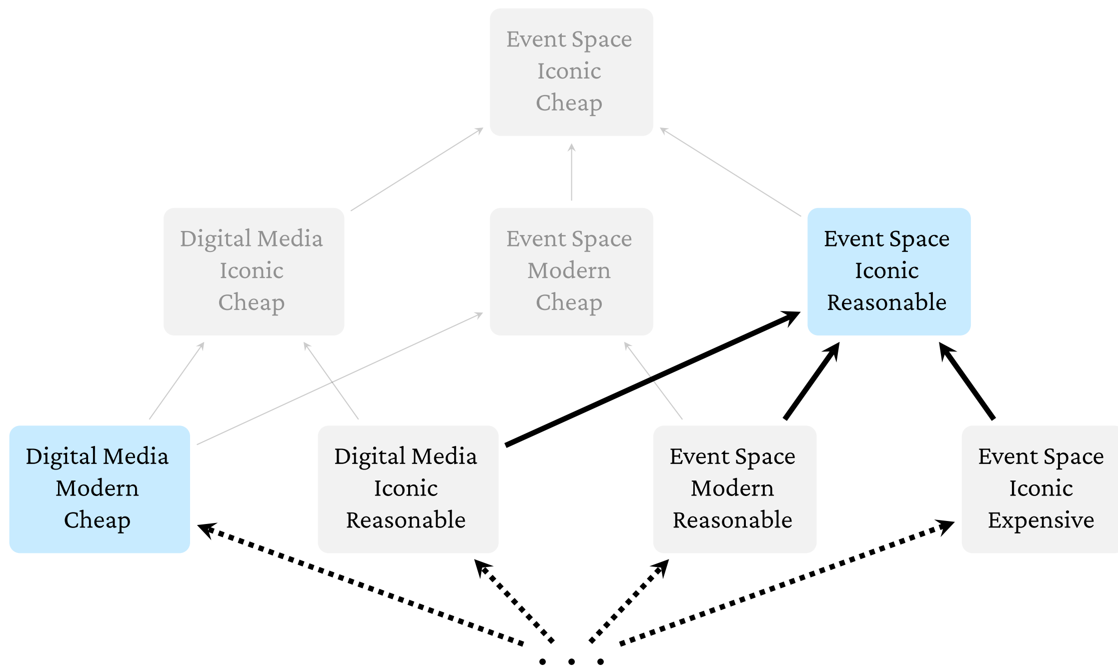


Figure 6.13: Library *careset* with feasibility and portion of best antichain

The Pareto frontier that the mcdp solver produces is a two-dimensional reduction of the antichain in the *careset* for the co-design problem. This projection visualizes two dimensions of the multi-dimensional trade-offs.

By using posets to describe our preferred worlds and by using implementations that restrict feasibility, monotone co-design allows us to use a computational solver that helps us move up these antichains within *caresets* and reveal the trade-offs between our best options.

Chapter 7

Compose

7.1 *Caresets*

In EPCODOT, users can create spaces to build out their urban systems. Within these EPCODOT spaces, users are able to create *caresets* freely. Every *careset* will be created with a set of elements (possible worlds) that are named by the user.

On creating the *careset*, there is an opportunity to provide a canonical ordering for the created *careset*. Most user-created *caresets* will likely be simple total orders to represent a specific quality, like *safety*, *sidewalk width*, *environmental impact* etc. The complexities, and corresponding antichains, come when these *caresets* are combined together through compositions.

An EPCODOT space can have as many *caresets* as desired, and those *caresets* can in turn be connected to co-design problems.

7.2 Co-Design Problems

Users will be able to create standard monotone co-design problems and an associated set of functionality *caresets* and resource *caresets*. We have seen that co-design problems live in the product *caresets* that contain the functionalities and resources, and users will be able to create these co-design problems and also view the corresponding product *caresets*.

In EPCODOT, co-design problems can be implemented in two ways: with standard implementations like is standard in monotone co-design and manually by users creating specific connections between resource and functionality elements. In manual implementations, each connection acts as an implementation for the co-design problem and represents a monotonic relationship. Like with standard implementations, connections determine feasible possible worlds when it comes time to compute best possible worlds in an EPCODOT space.

As more co-design problems are created they can be composed together and *caresets* can be used as wires between multiple co-design problems, or as intrinsic functionalities or resources within the overall EPCODOT space.

Because EPCODOT adopts the compositional capabilities of monotone co-design problems, it is possible to represent very complex systems at various levels of abstraction.

These co-design problems will be represented still by the standard co-design problem diagrams we have grown to know and love.

Figure 7.1: A co-design problem

7.3 Carespaces

In Chapter 2: Co-Design and Care, we made a connection between monotone co-design theory and the idea of *caresets* we've described in Part One: A Mathematical Language of Care. It turns out this connection is very natural, but so far it's been mostly a symbolic connection. How can we take advantage of this connection in EPCODOT to make it a better tool for participatory planning?

In order to take advantage of this generalization of co-design problems, we allow users to create *carespaces*. These are spaces that are composable just like co-design problems, and can be associated with *caresets* just like co-design problems.

There are two main reasons to use a carespace instead of a co-design problem: flexibility and accountability.

7.3.1 Flexibility

Co-design problems require defining every poset as either a functionality or resource. This works well for engineering contexts, or contexts where there are obvious, quantifiable relationships between different qualities. But sometimes the things we care about may not be easily mapped to functionalities and resources, or that mapping may not be immediately intuitive.

This is especially clear when moving up layers of abstraction. It is easy to understand what is a functionality and what is a resource for an engine that clearly "uses" fuel (a resource) to "generate" power (a functionality). But when we think of a higher level system, while we know there are trade-offs between the things we care about, we might not know how to structure them into a co-design problem with functionalities and resources.

A transit system always has a trade-off between frequency and coverage. Let's see how we might model that in co-design.

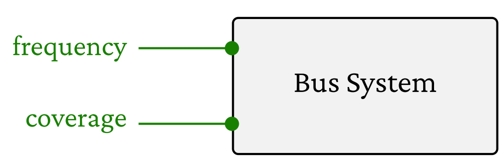


Figure 7.2: Bus system co-design problem

When a co-design problem has no explicit resource or functionality, monotone co-design instead uses a default, one-element poset. The single element is always feasible, and the co-design problem then represents feasibility relationships between the remaining posets.

Above, we've made a valid co-design problem that has no resource and only represents the feasibility trade-off between frequency and coverage. But there is no one canonical way to represent a real-world problem as a co-design problem, and we might not have captured all the relevant factors. Maybe cost, fleet size, or the number of available bus drivers are the resources limiting our bus system.

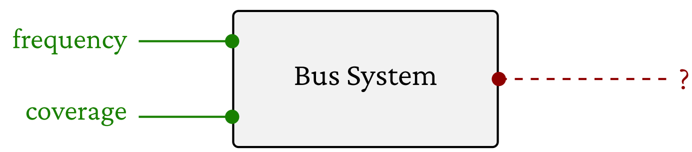


Figure 7.3: Bus system co-design problem. What's the resource?

Maybe we don't know which of these factors is actually limiting our system, so we aren't sure yet what to identify as a resource.

Functionalities and resources can be swapped by switching the order of the poset. This can be seen most easily with a resource like cost. We can invert any *cost* resource poset by multiplying all the values by -1, call it a *savings* poset, and use it as a functionality. Nothing would have to change about our implementations, and our co-design problem will have swapped a resource for a functionality.

Given this, we can try flipping our frequency from a functionality to a resource.

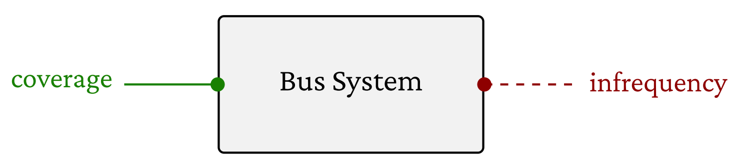


Figure 7.4: Bus system co-design problem with infrequency as a resource

We have another valid co-design problem that is a reasonable candidate for how to represent this real-world trade-off. In a sense, constrained bus systems do need to "use" more infrequency to "provide" more coverage.

But this may not align with how we naturally think about this trade-off. Functionalities can be inverted and called resources, but we care about the things we care about whether you show them right side up or upside down.

Carespaces allow us to represent our co-design problems without having to decide yet what is a resource and what is a functionality.

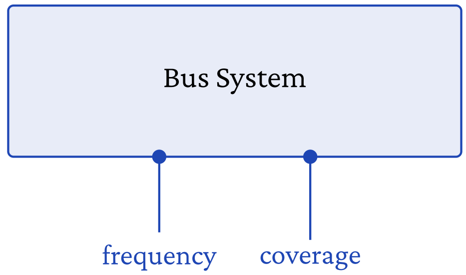


Figure 7.5: Bus System carespace

We can connect and decompose carespaces just like co-design problems. When we want to regain the computational capabilities of monotone co-design, we can assign the caresets of our carespaces to co-design problems. There are several more candidates for how to interpret our Bus System carespace as a specific co-design problem.

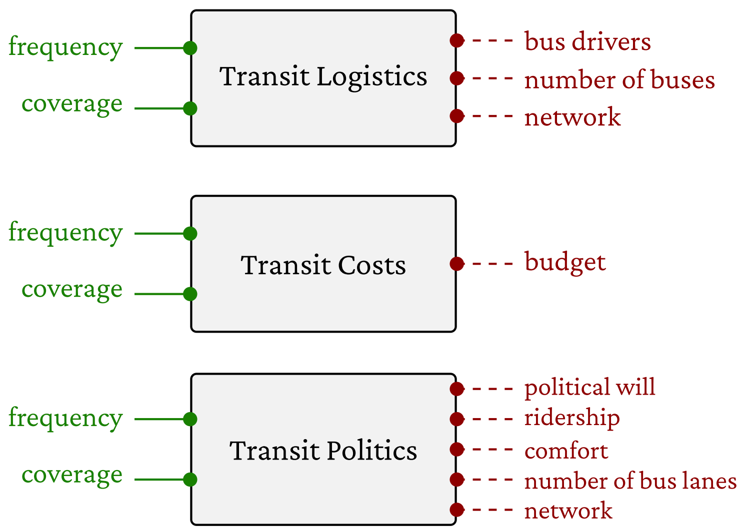


Figure 7.6: Candidate co-design problems in the Bus System carespace

Does higher ridership spur political will? Or will ridership only come once frequency and coverage are improved? These feedback loops are allowed in co-design problems, but by using carespaces we can defer the decision of whether a care is a functionality or a resource. We can use carespaces to put down the things we care about before deciding how to translate them into a workable co-design problem.

Carespaces will be represented by blue boxes, and *caresets* by blue wires. All co-design problems (gray boxes) are carespaces, and all functionalities (green wires) and resources (red wires) are *caresets*. But we will use the blue boxes and wires when dealing with carespaces outside of the context of a specific co-design problem.

7.3.2 Accountability

Carespaces also provide a tool for accountability outside of co-design problems.

We want to be able to model and design whole systems. To do that well it's useful to start from the goals of the system. The goals of the system should be the qualities that we care about intrinsically at the scope of the EPCODOT space.

In engineering contexts, it can be common to solve a problem because we *can* without understanding what greater improvement we are achieving through that solution. This accountability requires that we reckon with the goals of the system and justify our engineering solutions.

When users create carespaces, the terminology alone asks the question: what is it we really care about here? We can compose co-design problems together, but when we do so we risk just chaining together solutions without understanding what the problem we actually care about is. Carespaces allow us to design from the problem first before considering any solutions.

7.4 Composing Spaces and Problems

EPCODOT is as composable as monotone co-design. This means that carespaces can contain other carespaces and carespaces can be connected together, just like co-design problems. Carespaces and co-design problems can even be connected and nested interchangeably. *Caresets* are always the wires in an EPCODOT space, and the rule for these compositions along *caresets* is relatively simple.

Every child carespace must have every one of its *caresets* either

- connect to a *careset* in the parent space
- connect to a *careset* in a sibling space

This flexibility allows EPCODOT spaces to represent diverse types of composable and complex systems. We will see two examples of such systems in the upcoming case studies.

These composition rules ensure that all *caresets* are connected and accounted for and that there are no dangling wires. When an EPCODOT space is correctly connected, it becomes possible to use the mcdp solver to compute Pareto frontiers of possible worlds.

Chapter 8

Compute

8.1 Solver

In monotone co-design, we can use the mcdp solver to find what functionality we can achieve or what resources we need, and we can do so at any level. We can choose to use the mcdp solver on a small subsystem, or run the solver for the whole composed system. As long as all the wires are connected, the mcdp solver can work to find feasible solutions for any composition of co-design problems.

Recall the warehouse project composed co-design problem.

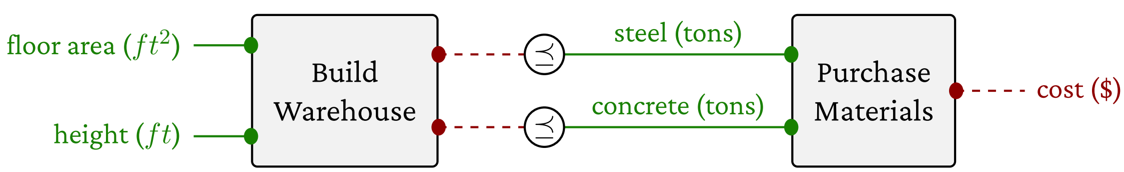


Figure 8.1: Inside the warehouse co-design problem

All the wires are connected, and if we have implementation options for every co-design problem, we can use the mcdp solver to see the tradespace for the composed warehouse project we described above.

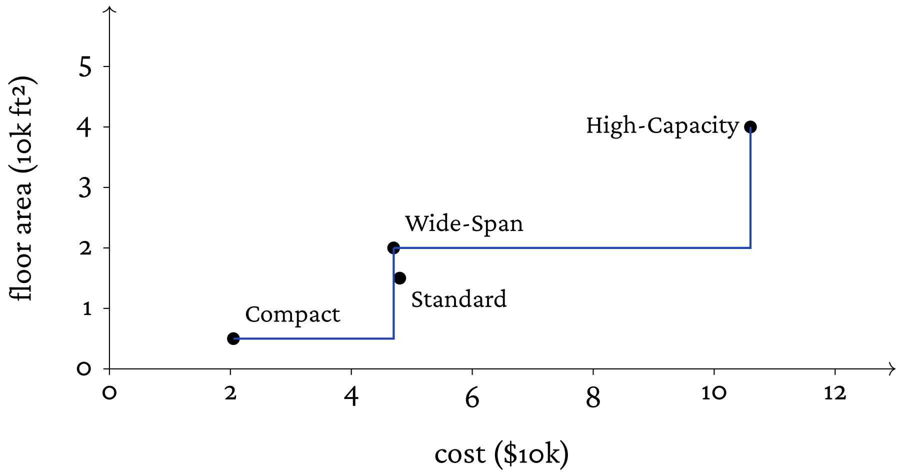


Figure 8.2: Warehouse project cost vs. floor area tradespace

When there are multiple combinations of implementations available for a composed co-design problem, the solver uses the monotonic connections between the co-design problems to filter feasible options and compute the non-dominated combinations for the selected functionalities and resources.

The Pareto frontier for this tradespace is given in blue. This Pareto frontier represents the functionality and resource values the solver returns for the warehouse project co-design problem if we want to maximize floor area and minimize cost. The upper left of this tradespace represents the best possible world we can achieve – maximum floor area and minimum cost. We can see that when we compare cost against floor area, the Standard Distribution Warehouse is dominated by other implementations.

This warehouse project is a simple example, but we can use it to see how monotone co-design uses posets and monotonic maps between those posets to represent increasingly complex systems and compute the trade-offs between different functionalities and resources.

8.2 Selecting Cares

EPCODOT takes full advantage of the mcdp solver. We know that the mcdp solver seeks the highest antichains in the product *caresets* for composed co-design problems. EPCODOT does not add any unique computational power to co-design, but it creates an interface to access these antichains and view them as tradespaces.

Because *caresets* are the primitives of EPCODOT, users are able to choose the *caresets* that they want to see projected onto a two-dimensional tradespace. If an EPCODOT project is fully composed, there is at least one co-design problem, and all the co-design problems have at least one implementation, then the user can select a care and run the solver. If the user selects two cares, they can see the tradespace of possible outcomes just like they can with monotone co-design.

Because the mcdp solver has certain computational limitations, workarounds are used to keep the solver interaction simple for users. It is important that EPCODOT be easily usable across all levels of expertise, so running the solver needs to be as simple as selecting which *caresets* you want to see charted, and running the solver.

8.3 Viewing *Caresets* and Antichains

A unique interface idiom that may be worth pursuing for EPCODOT is to allow users to view Pareto frontiers as antichains in *caresets*. The *caresets* that represent co-design problems explode in complexity as there are more *caresets* being combined together. But it would be novel to provide a user interface that allows users to browse and see inside the *careset* that represents all the possible worlds in the co-design problem or in the whole EPCODOT project.

Because the non-dominated Pareto elements that the solver discovers are necessarily part of an antichain, it would be possible to highlight the antichain within the entire *careset*. This unique presentation would reveal some topological qualities about the entire tradespace that are not possible to see when the query results are viewed as points on a two dimensional grid.

Chapter 9

Collaborate

9.1 Proposal Space

In standard co-design problems, implementations are provided as a way to describe feasible combinations of resources and functionalities. In order to make EPCODOT collaborative, the proposal space makes it possible to have the creation of those implementations be a participatory process.

A proposal space is very simple. It is a carespace that contains one discrete *careset* of proposals with no relations. The proposals are an open *careset* where users can submit their own proposals and add them to the *careset* directly.

The proposals *careset* can then be connected to a co-design problem to be used as the set of implementations.

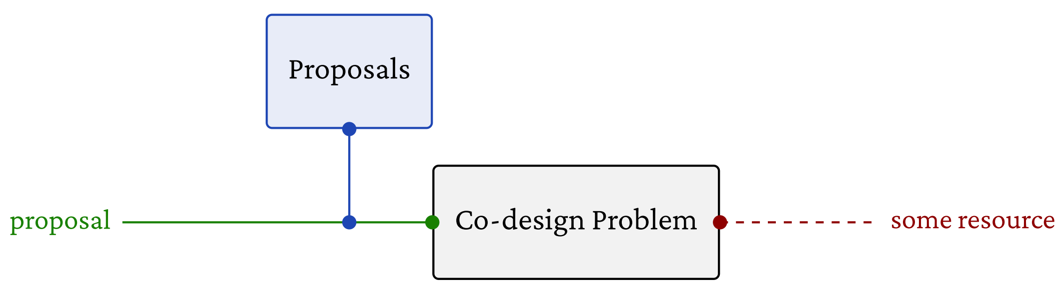


Figure 9.1: Proposal space connected to a co-design problem

The associated co-design problem can have other functionalities or resources as well, and of course the proposal *careset* can then be connected to other co-design problems or carespaces just like any other *careset*.

When we add a new proposal to the proposals *careset*, this triggers a recalculation of all connected co-design problems. Every proposal needs to be evaluated independently, and the associated functionalities and resources need to be calculated and trickled down to all the connected co-design problems.

In practice, professional help will likely be useful for vetting and aiding the submission of feasible proposals. But in order to be an honest participatory planning framework the proposals space needs to be fundamentally open to submissions from the public.

9.2 Consensus Space

Consensus spaces are a unique construction provided by EPCODOT to make the process of finding the best possible outcomes in an EPCODOT project collaborative. The consensus space is a carespace that lets users add new relations to reduce the size of the Pareto antichains for the representative *careset* of the carespace.

We draw a consensus space using dotted wires to represent the *caresets* that come to the top of the carespace.

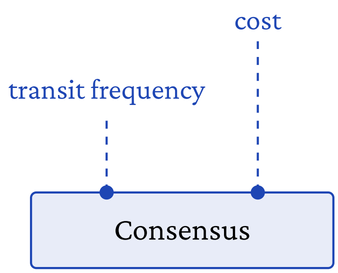


Figure 9.2: Consensus space

9.2.1 Casting Votes

The consensus space allows users to be shown pairs of possible worlds in the representative *careset* that are within antichains. When shown these possible worlds, the user is asked to decide which of the two elements they would prefer were realized.

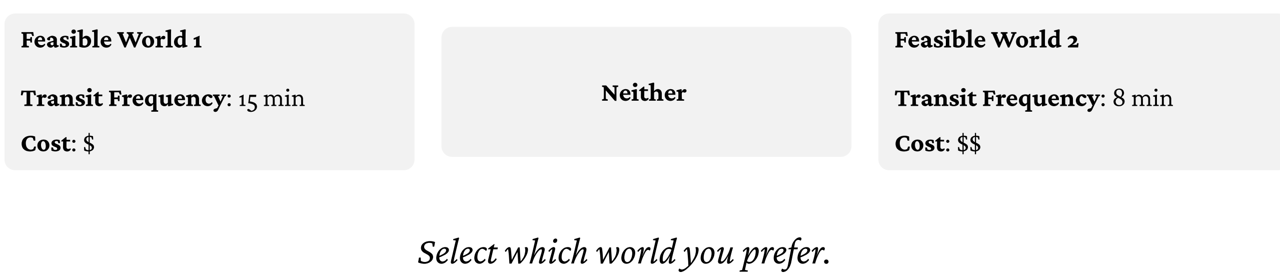


Figure 9.3: Possible comparison for Transit Frequency × Cost consensus

Every user will have their own copy of the original *careset* they are voting on, and every time a user makes a decision on a preferred world within an antichain, that preference relation will be added to their copy of the *careset*. The user may also choose to say that they have no preference between two worlds, and they will remain in an antichain together in the user's preference *careset*.

When asking the user to state their preference, EPCODOT will always show the user elements that are within an existing antichain. Relations that exist in the original *careset* are not presented to the user. When the user makes a preference, it's possible that there will be many other transitive relationships that are implied by that preference. This makes for a process that asks as few questions as possible from a user while getting as much resolution as possible regarding their preferences.

Let's look at the transit frequency × cost *careset* from Chapter 2: Preference, but this time focus on only the feasible possibilities.

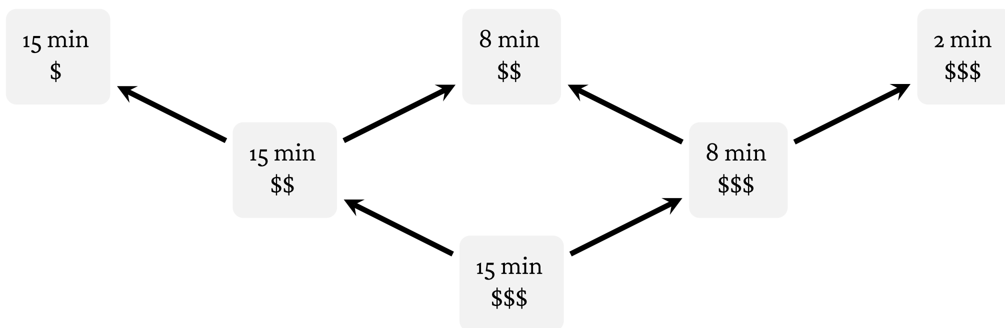


Figure 9.4: Transit Frequency × Cost design space feasible *careset*

It's important to remember that not all votable elements will be feasible. A consensus space filtered to only ask users to make preference decisions between feasible elements of the *careset* can significantly reduce the number of voting rounds required.

Still, a consensus space on this *careset* might have to show up to 5 comparisons to surface all the incomparable elements to each other for the user. But, the consensus space could show the following as its first comparison.

Feasible World 1		Feasible World 2
Transit Frequency: 15 min	**Neither**	**Transit Frequency**: 2 min
Cost: $$		**Cost**: $$$

Select which world you prefer.

Figure 9.5: Possible comparison for Transit Frequency × Cost consensus

If the user happens to choose World 1 over World 2, the width of the *careset* gets significantly reduced with this one new relation.

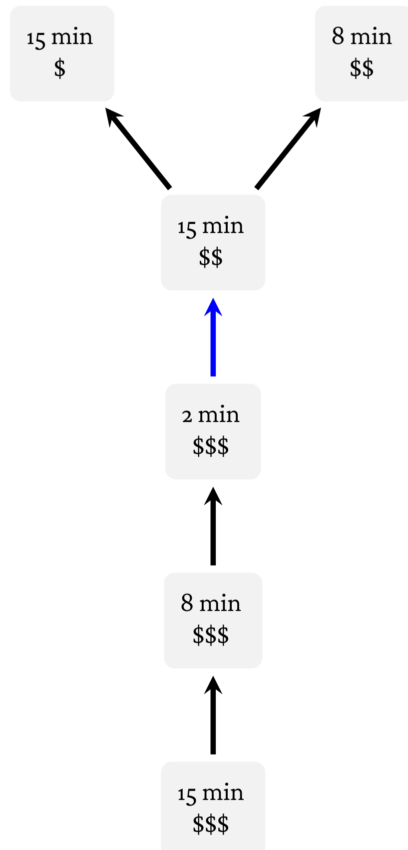


Figure 9.6: Transit Frequency × Cost *careset* after vote (new relation in blue)

And we're left with only one comparison left to show the user, removing the need to ask three questions because of the transitive repercussions of the user's first preference.

If the user chose World 2 or Neither, we would still have up to four remaining comparisons we would have to ask the user.

9.2.2 Voting with *Caresets*

EPCODOT allows users to be shown three (or more) possible worlds at a time from a given antichain, and asked to create a small *careset* from the presented elements. This leverages the familiarity of the system with *caresets* to reduce the number of actions required for a user to fully determine their personal preference structure in the *careset*. The cost to this more complex voting type is the added cognitive load required to complete a *careset* from the provided elements.

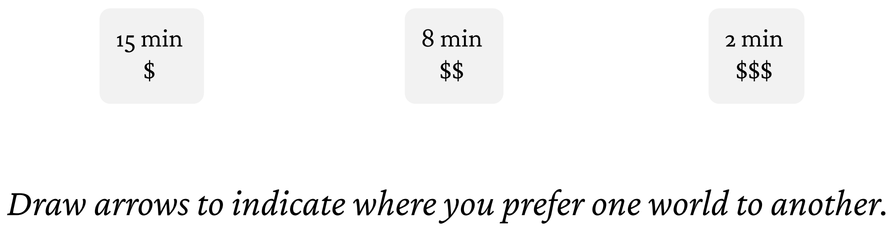


Figure 9.7: *Careset* antichain triplet before user input

The user can choose to draw up to two arrows, and the consensus space will consider any worlds left unrelated to be equally preferable according to the user.

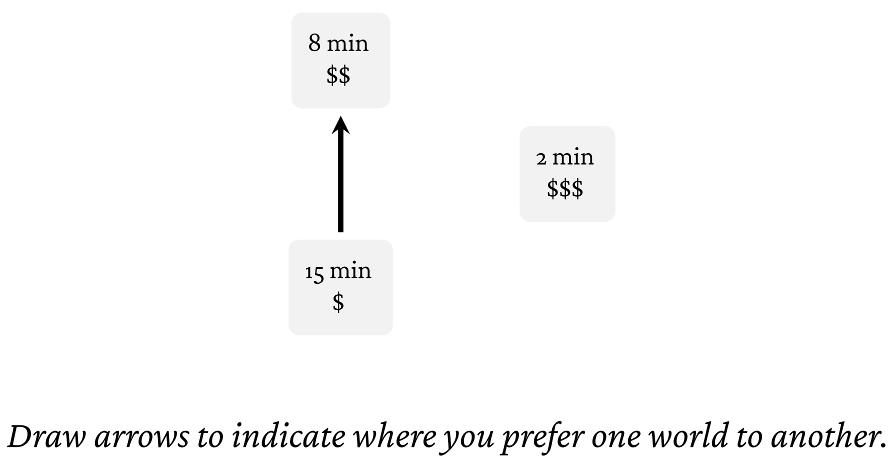


Figure 9.8: Possible voted *careset* triplet

9.2.3 Deriving Consensus

EPCODOT compiles all the *caresets* provided by users who have given their preferences, and uses a threshold to decide the amount of consensus required for a preference relation that does not exist in the original *careset* to be carried forward to the final consensus *careset*.

The threshold will be used to determine what margin is required for a preference relation to make it into the consensus *careset*.

A threshold of 50% will require that the only preference relations that exist in the consensus *careset* are the relations that were preferred by at least a 50% margin of the preference *caresets*. A threshold of 0% will make it so any preference relation can make it into the consensus *careset* even if its winning margin is by a single vote. A threshold of 100% will make it so only unanimously agreed upon preference relations are carried forward into the consensus *careset*.

If there are five participants, that means that a 50% threshold will take any relations where four or more voters agreed on the preference relation – four votes for vs one vote against is an acceptable margin for a 50% threshold ($\frac{4-1}{5} > 50\%$).

For a 0% threshold, a preference relation needs 3 votes. For a 100% threshold, it needs all 5 votes.

The complication in a poset is the requirement that there be no cycles. It is possible that different participants will think differently regarding the antichains of the original *careset*.

If, after the threshold filtering, the *careset* contains a cycle, then the lowest consensus preference relations will be dropped one at a time until the cycle is removed.

Each relation in the consensus *careset* is enriched with a fraction that represents the number of preference *caresets* that agreed with the existence of that relation. Naturally, the lowest fraction should be above 50%, as no relation that has more votes against it than for it will ever make it into a consensus *careset*.

The EPCODOT consensus space provides two interesting capabilities:

Votes can be provided as partial orders. This allows users to express their preference without needlessly forcing a total order of rankings.

Revealing consensus through vote-enriched partial orders. A threshold slider can visualize the distribution of consensus and help show participants where in the *careset* there is more agreement and where there is less.

Put together, these two capabilities provide a unique method and interface for surfacing shared values and accepting community input to transparently derive consensus.

9.2.4 A Note on Arrow

We are able to leverage the poset structure of our *caresets* to avoid Arrow's Impossibility Theorem. We know that by supporting poset structures of consensus we can escape Arrow's impossibility and actually produce structures that accurately reflect consensus (Arrow, 1963; Sen, 1970).

In exchange we live with antichains where disagreements are merged together as incomparable options. We can then use the consensus threshold to show how the consensus compares to individual preferences.

There are likely other interesting ways to use the poset structure to communicate and visualize consensus differently. There is also significant existing research on how to best use posets for voting and aggregation (Fishburn, 1985).

There is definitely room to leverage posets more when managing consensus in EPCODOT.

Chapter 10

Case 1: Trade-Offs in Urban Well-Being

10.1 Data Slots

In order to illustrate the basics of EPCODOT, we use the MIT Senseable City Lab's Data Slots research project. Data Slots was originally developed as a card-game method to assess trade-offs between data-driven innovation and privacy, and has been played both online and in-person. Participants trade data cards, propose a solution for one of three scenarios (home, workplace, or public space), players rate each proposal on perceived benefit and privacy invasiveness, and then "invest" in the ideas they favor (Mazzarello et al., 2025).

Data Slots has been played more than two thousand times across 79 countries, and many cities have used the in-person version of the game as a participatory design activity linking researchers, officials, and residents. For our purposes, we focus on the in-person workshop format with tables of four, mirroring this participatory design activity. We model the ideation, benefit/invasiveness scoring, and investment phases to show how EPCODOT composes these interactions and how the EPCODOT framework can be used for this fundamental controlled version of participatory design.

10.2 Table

In the context of Data Slots, one can think of the table of four people playing the game as a single EPCODOT space. The overall game space will be decomposed to subspaces that will cover different aspects of playing the game, but all the game actions will at the end result in a space like the following.

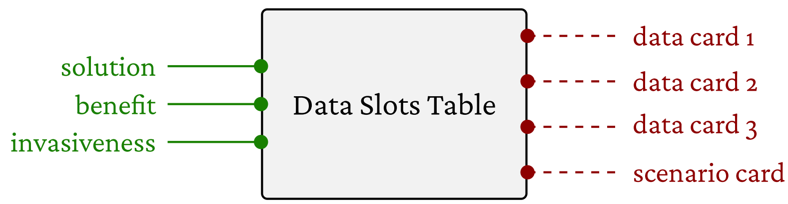


Figure 10.1: Data Slots table as a co-design problem

We have gone ahead and identified which *caresets* belong as a functionality and which as a resource. This will allow us to create a proper co-design problem that we will begin to decompose as we follow the structure of the game.

10.3 Proposals to Improve Well-Being

In the game of Data Slots, players are first given a scenario card to describe one of three scenarios that the rest of the game will be played in. We can consider this to be one of the first cares of our resource *carespace*. No preference relation exists here.

home	work	public space

Figure 10.2: Data Slots scenarios *careset*, no relations

Players are then given cards that represent different types of data with which to construct a product or solution that could be used to provide a benefit in the chosen scenario. These data types include examples like health data, mobility data, animal mobility data, and many others. The players play a few rounds of holding and discarding cards between each other to finalize the data cards they hold in their hand until they end up with three data cards. Given three types of data, the players are then asked to think of a product or solution that they could build using this data that would provide some benefit to urban well-being.

Greenery	Animal Mobility	Infrastructure	Environmental
Human Mobility	Utility	Vehicle Mobility	Health
Personal Profile	Dietary Habits	Social Networks	Electronic Transactions

Figure 10.3: Data Slots data cards *careset*, no relations

The actions players take to keep or discard data cards provides data on what types of data may be considered more valuable by players hoping to create a beneficial solution. This dynamic is not modeled in EPCODOT, but we can successfully model the simpler output: each player proposes a single implementation that uses three types of data.

Name	Solution	Data card 1	Data card 2	Data card 3	Scenario
Idea #4	Comprehensive workplace solution with personal and health data	Personal Profile	Health	Human Mobility	Work
Idea #8	Workplace wellness using health and dietary monitoring	Health	Dietary Habits	Human Mobility	Work
Idea #10	Workplace infrastructure and mobility optimization	Health	Animal Mobility	Vehicle Mobility	Work
Idea #11	Work-transport integration with social and transaction data	Personal Profile	Electronic Transactions	Social Networks	Work

Table 10.1: Data Slots proposals from Table B in Paris, France

We can draw the proposal stage of the Data Slots game using the proposal space and an associated co-design problem.

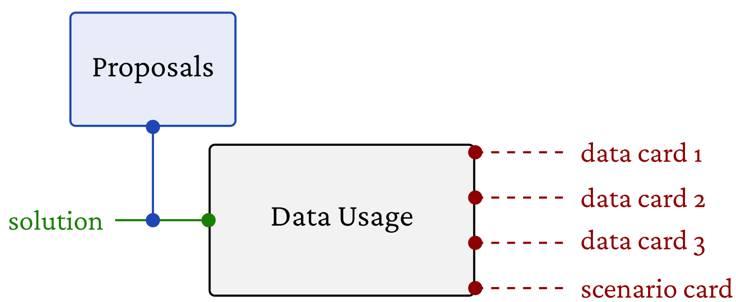


Figure 10.4: Data Slots solution proposal stage

We can see the *careset* of solutions proposed by the players at the table.

Comprehensive workplace solution with personal and health data	Workplace wellness using health and dietary monitoring	Workplace infrastructure and mobility optimization	Work-transport integration with social and transaction data

Figure 10.5: Data Slots solution proposals *careset* for Table B in Paris, France

10.4 Consensus Through Investment

In Data Slots, users are given a set number of investment tokens to distribute between the ideas they most prefer among those generated by the others at their table. Each player implicitly considers many factors when making preferences between the ideas at the table, including the invasiveness of the data being used, the benefit provided by the proposed solution, and also the feasibility of the idea itself.

We can consider this to be a consensus space that takes on all the cares that come out of the proposal space. Then users "vote" with their investment tokens, where the differences in the amount of investment tokens given to each idea indicates the user's preferences. If for any two ideas the user decides to invest the same amount, this will be considered no preference. This is an example of why it is useful to have *caresets* be posets.

For our purposes, we can model the Data Slots investment round as standard, non-enriched voting round. From this, each table refines the *careset* that came out of the proposal space.

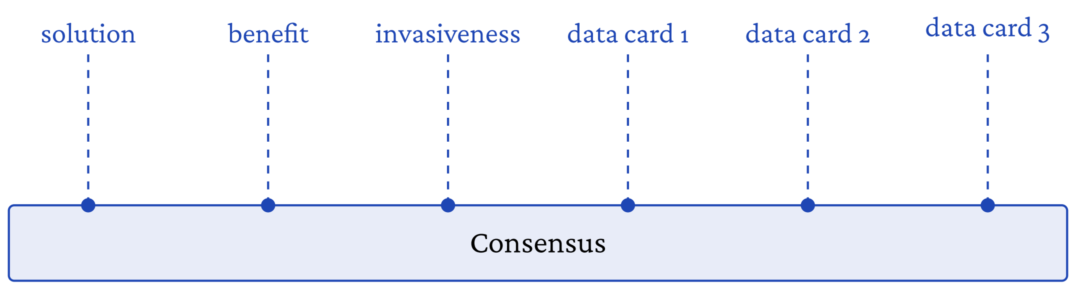


Figure 10.6: Data Slots table consensus space

The consensus space uses *careset* wires because it only manipulates the structure of the *caresets*. Because it does not modify any feasibility relationships, or create any new *caresets*, it does not need to connect any *caresets* specifically as functionalities or resources. We also show this difference by having the wires connect to the top of the consensus space.

And each vote in EPCODOT for a proposal look something like this.

Feasible World 1		Feasible World 2
Solution: Comprehensive workplace solution with personal and health data **Data Card 1**: Personal Profile **Data Card 2**: Health **Data Card 3**: Human Mobility **Scenario**: Work	**Neither**	**Solution**: Workplace wellness using health and dietary monitoring **Data Card 1**: Health **Data Card 2**: Dietary Habits **Data Card 3**: Human Mobility **Scenario**: Work

Select which world you prefer.

Figure 10.7: Example of a pairwise comparison in a consensus space

The consensus space adds order. Without a consensus space we might have a *careset* for the whole space that looks like this.

Solution: Comprehensive workplace solution with personal and health data **Benefit**: 4.0/5 **Invasiveness**: 3.25/5 **Data Card 1**: Personal Profile **Data Card 2**: Health **Data Card 3**: Human Mobility **Scenario**: Work	**Solution**: Workplace wellness using health and dietary monitoring **Benefit**: 4.25/5 **Invasiveness**: 4.0/5 **Data Card 1**: Health **Data Card 2**: Dietary Habits **Data Card 3**: Human Mobility **Scenario**: Work	**Solution**: Workplace infrastructure and mobility optimization **Benefit**: 3.5/5 **Invasiveness**: 2.75/5 **Data Card 1**: Health **Data Card 2**: Animal Mobility **Data Card 3**: Vehicle Mobility **Scenario**: Work	**Solution**: Work-transport integration with social and transaction data **Benefit**: 4.25/5 **Invasiveness**: 2.75/5 **Data Card 1**: Personal Profile **Data Card 2**: Electronic Transactions **Data Card 3**: Social Networks **Scenario**: Work

Figure 10.8: Discrete *careset* of 4 feasible possible worlds without consensus

When we let players state their preferences, we can add resolution.

Figure 10.9: Possible *careset* of feasible solutions with consensus

Of course, the consensus threshold used will impact the number of preference relations that can be added to the final *careset*. If all players believe their solution is best and all other solutions are equally bad, then the resulting consensus will be unchanged from the original *careset* – no preference relations will be added by the consensus process.

The difference between EPCODOT and Data Slots here is that votes are enriched with an investment amount. Players not only state that they prefer one idea over another, they can state how much more they prefer the idea. This enriched voting is also very much possible for EPCODOT to support, but would require a modified consensus algorithm. Likely the consensus threshold would become a tuple that contains both the percentage margin required for a preference relation to exist, and a minimum magnitude required for a preference relation to make it into the consensus threshold.

Until we add the enriched voting flow to this framework, we will live with more ties. Having the votes enriched with a value (like an investment amount) helps to provide more resolution and provide more opportunities to tie-break and reduce the size of the antichains. This has its benefits because it allows for more clear winners in a voting process. But it does risk that the added precision will not be as honestly representative of reality.

10.5 Scoring with Co-Design Problems

The Data Slots game asks its players to rate each idea at their table on two dimensions: benefit and invasiveness. We can model this using co-design problems in EPCODOT.

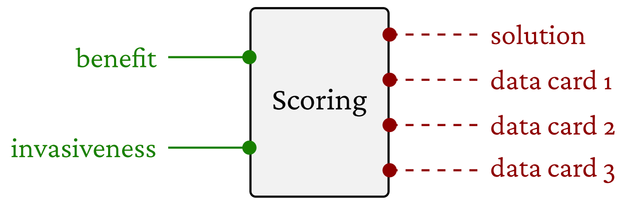


Figure 10.10: Data Slots table scoring co-design problem

Each player rates the benefit and invasiveness on a scale of 1–5, and the average is used to determine the score for the solution.

We can model this in EPCODOT using another co-design problem that takes all the individual scores and combines them.

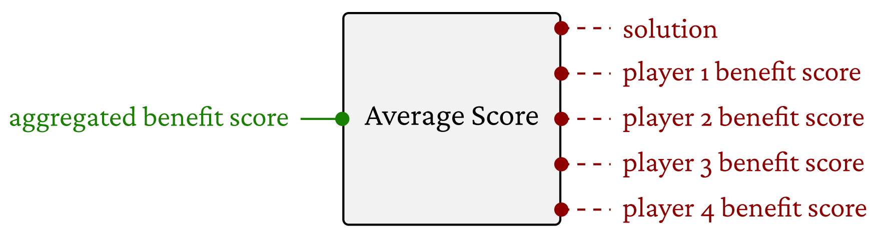


Figure 10.11: Data Slots table scoring co-design problem

For the sake of simplicity, we can assume that the *score* co-design problem contains within it one co-design problem for each player then uses a co-design problem like the one above to combine the average the individual scores score.

10.6 Composing the Pieces Together

In Data Slots, you can see the structure that forms when we put together all the spaces we used above.

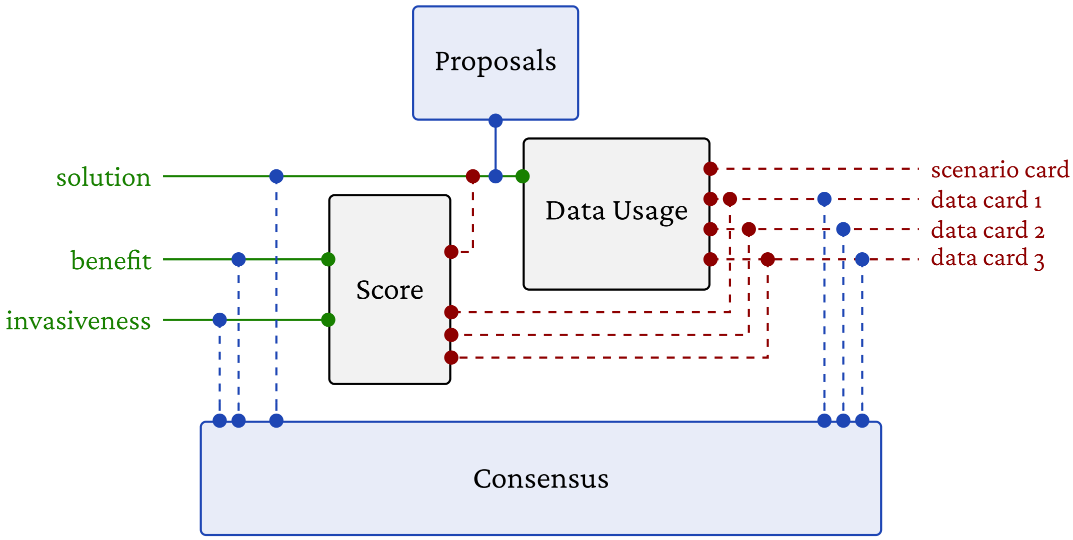


Figure 10.12: Data Slots table EPCODOT space

In co-design, when we split wires we have to do so carefully. It must be clear how the wire splitting works with regard to feasibility. We cannot use a resource or functionality twice in two different places. The wire splits here are allowed because we are not consuming the resources – scoring the invasiveness of a certain type of data doesn't disallow that data from being used by a solution. In general, functionality and resource wires have to be split carefully and often a co-design problem is needed to encode the splitting logic.

Blue wires represent when *caresets* are being used in ways that do not impact feasibility. Proposal and consensus spaces do not restrict feasibility, which is why they connect using blue wires.

We can then step up a layer of detail and summarize a Data Slots table like we did before.

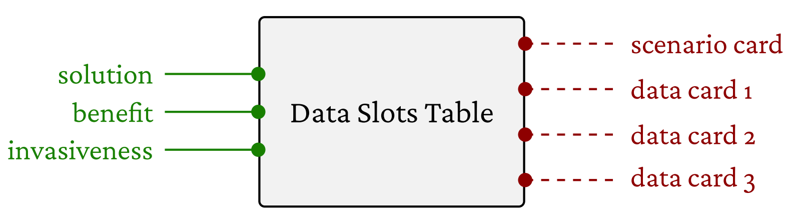


Figure 10.13: Data Slots table as a co-design problem

We can then see what it would be like to put all these Data Slots tables side by side.

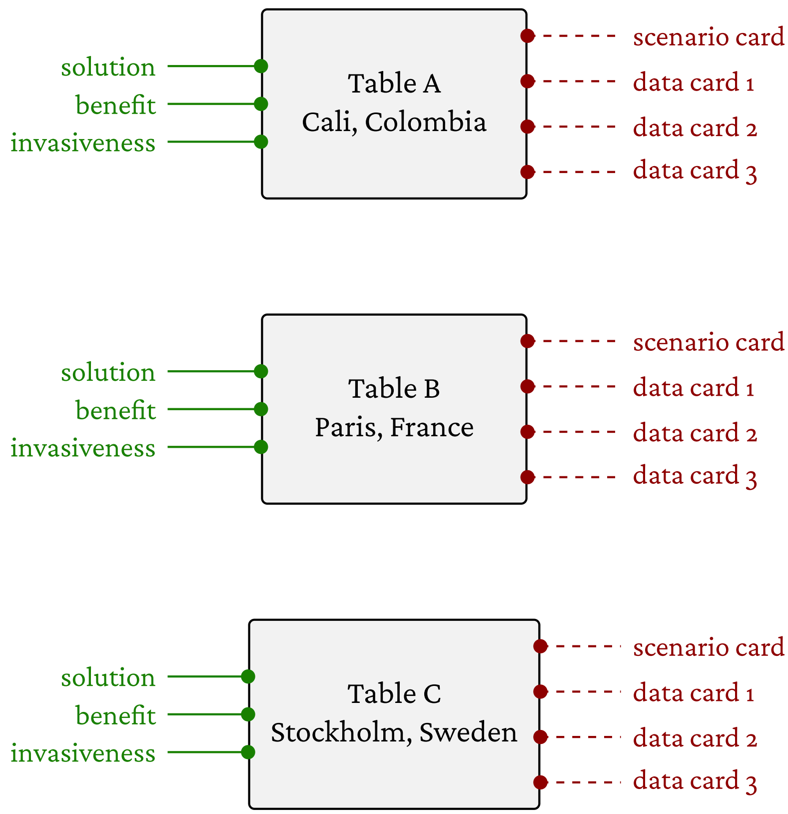


Figure 10.14: Data Slots tables as co-design problems

What if we combined all data from all the tables together?

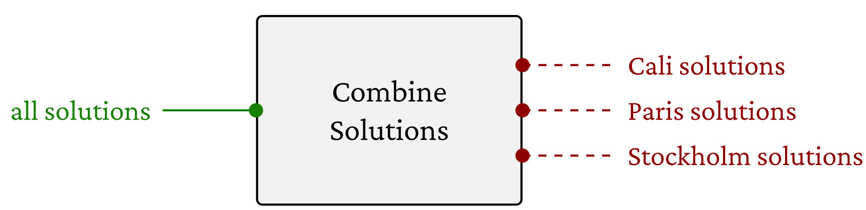


Figure 10.15: Combining solutions as a co-design problem

Then we can zoom out one more time to see all the co-design problem that represents all the in-person Data Slots games played around the world.

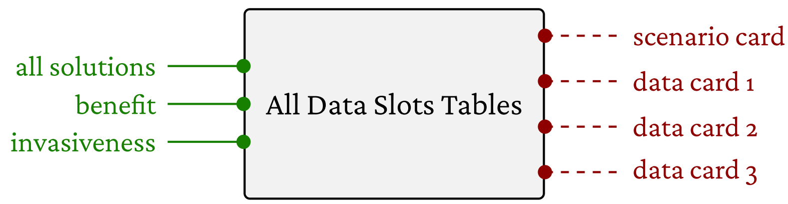


Figure 10.16: All Data Slots games as a co-design problem

We can combine the solutions produced by tables across the world, use the consensus space to have the public express their opinion on which trade-offs of data usage and solutions for urban well-being they prefer. We can then use the mcdp solver to determine to surface a Pareto frontier of best possible worlds in this EPCODOT space.

Chapter 11

Case 2: Participatory Planning in Durham

11.1 In the Public Sphere

In order to see if EPCODOT could be used as a tool for collaborative decision-making in the public sphere, let's see what it would look like to model a real urban project.

In late 2023, residents, advocates and officials in Durham, North Carolina, were in discussions for a possible improvement to the North Roxboro Street corridor (Roxboro). As I was on the Bicycle and Pedestrian Advisory Commission at the time, I participated in and observed these discussions. The case of North Roxboro Street contains all the complexities we have come to expect from issues that have to do with our local built environment: many stakeholders with different jurisdictions and incentives. This case will be a useful one to help see if EPCODOT has potential as a framework that can handle these real-world complexities.

Here we will walk through the steps for how the EPCODOT framework could be used to help realize a project on Roxboro, guided by the real interactions that were had with advocacy organizations, public institutions, and communities in 2023 (Durham Bicycle and Pedestrian Advisory Commission, 2024).

11.2 Initial Engagement

Whether spurred by an advocacy organization or by local residents, a sound first step for any public project is to try to learn more from those who would be impacted the most. What are top concerns local residents have? What are the problems they experience day to day? What do they really care about?

Community engagement processes vary greatly, but EPCODOT requires one specific output from any engagement process: a set of *caresets*. We use these *caresets* to ensure that as project design and planning continue, we measure the impacts on these *caresets*.

Reflecting the engagement process with the local Bragtown community, the following engagement carespace was created.

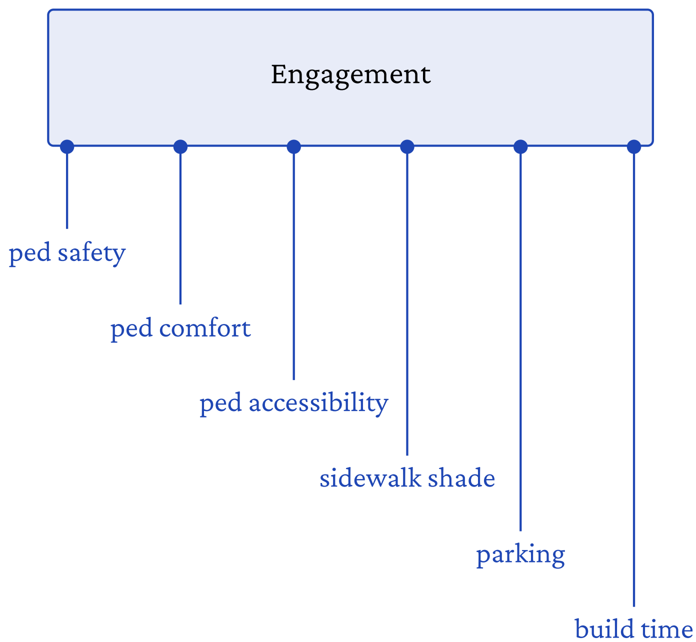


Figure 11.1: Engagement co-design problem with *caresets* for pedestrian safety, comfort, accessibility, shade, parking, and build time

We can see that many locals emphasized their concern for the pedestrian experience. Local businesses weighed in too – they care about the amount of parking available for their customers. And given concerns with how slowly public projects have moved in the past, there was also a question of what the build time would be for any improvements to Roxboro.

The engagement space does not require residents to rank which *caresets* they value most. It simply serves as an opportunity to document the things they want to see. When we create an engagement *carespace*, the dangling *caresets* are wires that must be connected at some point in the greater space.

The *caresets* described also do not have to have their elements specified from the outset. We can let the co-design process lead us to define these *caresets* more precisely, but for now we know we want *caresets* that match these descriptions surfaced at some point in our co-design process.

11.3 Initial Proposals

Given this initial engagement space, we can create a proposal space to begin taking in initial ideas.

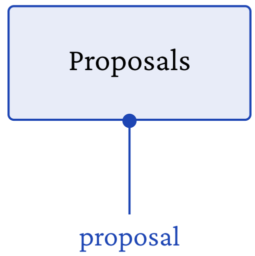


Figure 11.2: Proposals *careset* output

Likely a couple ideas came up from the engagement process or were suggested by an advocacy organization. These ideas might be ambitious, and they may not take into account all considerations. That's ok. The proposal space can be used as an entry point for any ideas that might realize a better possible world with the *caresets* in the engagement *carespace*.

Through the initial engagement two proposals were submitted: a proposal that converts Roxboro from 3 car lanes to 2 providing space for bike and increased sidewalk (a significant change), and a proposal that leaves the car lanes untouched but replaces on street parking with widened sidewalk and street trees (a smaller change).

11.4 Collaborative Co-design

11.4.1 Bike Walk Advocates

The local bicycle and pedestrian advocacy organization may be eager to show how changes to Roxboro could improve the experience for local residents. They take on a chance to build the following co-design problems to make their case.

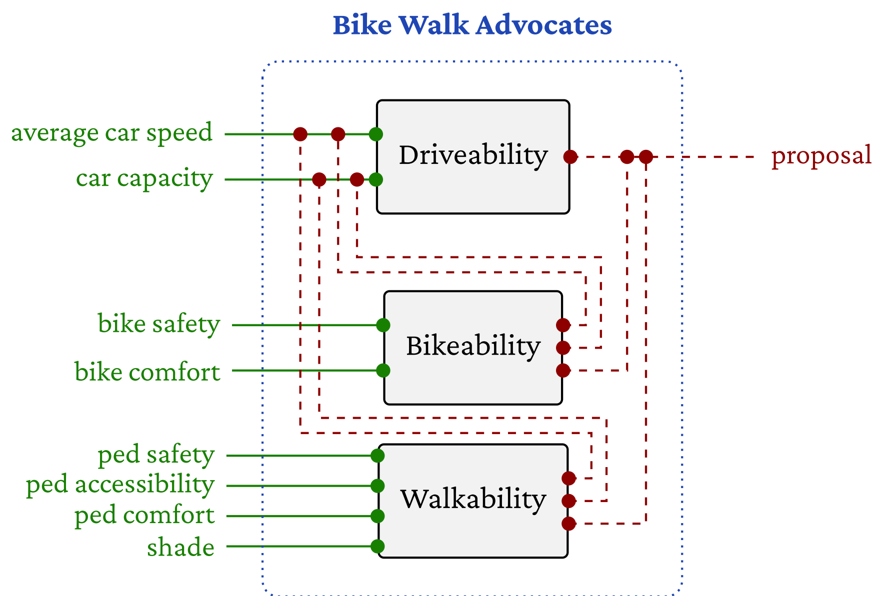


Figure 11.3: Bike Walk Advocates co-design problems

The advocates can break down the variables they think matter for making a better cyclist and pedestrian experience. They connect the car speed and car capacity to the bikeability and walkability co-design problems because they know that these three co-design problems are not independent, and through these connections they encode a trade-off. As we increase average car speed and car capacity we will likely reduce pedestrian and cyclist safety and comfort.

Inside the co-design problems there could be more detail than is shown here. The proposal could be fed into a nested co-design problem that produces a sidewalk width, number of bike lanes, bike lane protection, curb cuts, street crossing time, etc. Then these physical components could be used to evaluate the safety, comfort, and accessibility qualities desired at the parent co-design problem. This composition allows for as much detail as is needed, while also being able to show the work for how to get from a proposal to the *caresets* we are focused on.

You can see that the advocates created more functionalities from their co-design problems than are asked for by the engagement space. The engagement space provides the minimum *caresets* we need, but in a collaborative process other stakeholders can surface other *caresets* they believe are important.

For each of the two proposals, the bike walk advocacy organization works with city employees, in-house volunteers, and local residents to determine the functionalities provided for these co-design problems.

11.4.2 North Carolina Department of Transportation

The North Roxboro Street right-of-way is one of many streets in Durham maintained by the North Carolina Department of Transportation (NCDOT). This makes the stakeholder interactions more complicated than they would be otherwise, but we can always bring in more stakeholders with EPCODOT.

The City engages NCDOT to assess the existing proposals. To do this, NCDOT creates the following co-design problems.

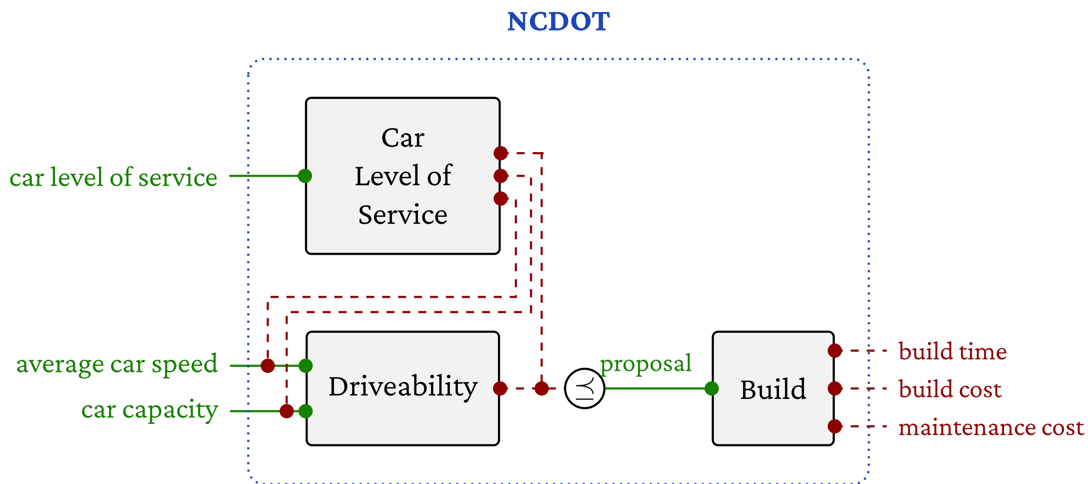


Figure 11.4: NCDOT co-design problems

Because NCDOT is the maintainer of the corridor, they provide their first estimates for what an implementation of these two proposals might look like. This co-design problem could have more detail within it. The proposal that only involves removing parking spaces and adding trees might be shared with the City. The details of what entity pays for what components can be broken down here. There could even be proposals with in this build co-design problem: what if the city purchased the right-of-way from NCDOT? These complexities can all be expressed through composition.

NCDOT also reused the Driveability co-design problem created earlier. This shows one of the values of EPCODOT and co-design in general. When reusing wires, it surfaces places for disagreement and requires that they be consolidated. Do NCDOT and the Bike Walk advocates agree on how the average car speeds and capacities would be impacted by the two proposals? If not, they will have to create two separate co-design problems that reflect their differing hypotheses. This still works fine in the EPCODOT structure.

NCDOT surfaces an important care that has not been brought up yet: level of service (LoS) for single-occupancy vehicles. The LoS is calculated using the existing average annual daily traffic numbers and projections for population growth. Depending on the capacity today, a right-of-way is graded on how well it can support future growth. This is one of the key performance indicators for NCDOT, and for all streets they require they have to ensure that the LoS meets required standards.

If the NCDOT requires a certain LoS to approve the project, they can reflect this by creating a feedback loop. They can connect the car LoS functionality as a resource required by the Build co-design problem. If a poor LoS grade can block a project, it's important to have co-design problems that communicate these dependencies so all stakeholders can see them, especially when they involve a less well-understood and often controversial metric like LoS.

11.4.3 Expanding Collaboration

We can keep integrating stakeholders. GoDurham is the organization that runs the bus system in Durham, and they might be brought in to assess each proposal for any potential impacts to the service quality of the bus route that runs on Roxboro. There might be spot co-design problems built to match specific concerns of certain stakeholders, like a co-design problem to surface the amount of customer parking that will be provided for businesses along Roxboro. One can imagine other stakeholders and even other engagements, checking with the local schools, and bringing in the fire department.

For now, we can include two co-design problems for the bus system and for parking.

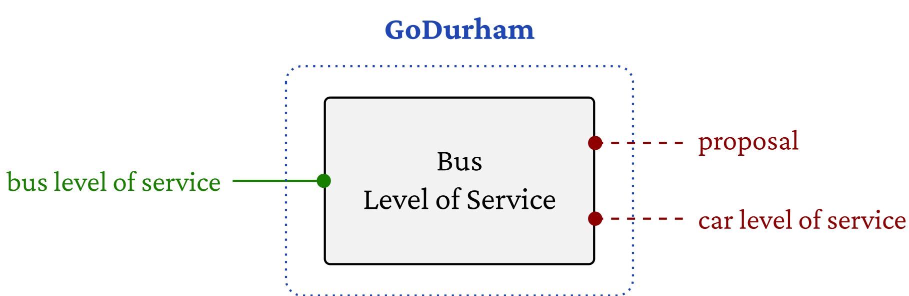


Figure 11.5: Bus Level of Service co-design problem

Figure 11.6: Parking co-design problem

11.5 Proposals Phase

After seeing details about how all the stakeholders view the different impacts of a given proposal and seeing what is required to get approval from the NCDOT, the project can be opened up to more proposals.

This provides an opportunity to leverage the community's own creativity and interest. We can imagine proposals coming in that range from large in ambition (make Roxboro car free) to those that are meant to provide a small but quick win (red light cameras).

As new proposals come in, any co-design problems that are downstream from the proposals will be triggered to be updated by the stakeholders that are responsible for the co-design problems. This continues until all the related conversion spaces are updated.

Some co-design problems will not need to be updated, because they are already implemented by formulas and not manual conversion spaces. The Car Level of Service co-design problem is an example of this, as it is implemented by a traffic engineering formula that depends on current average annual daily traffic, projections, and the car speed and capacity of the proposal determined by the Driving co-design problem.

By allowing all the new proposals to come in during a proposal submission phase, these updates can be deferred until after the new proposals are submitted and the submissions are closed. This allows all the manual co-design problems to be updated at one time.

During this phase or any phase, new co-design problems and carespaces can be added too. This process for any public project is dynamic, and EPCODOT supports this flexibility. For any new *carespace* or new co-design problem, all that is required is that the wires are connected in order for the space to be complete.

Once this period is over, we will have a whole new set of proposals to choose from that still fit the original *caresets* all our stakeholders had surfaced before.

11.6 Consensus and Governance

Once we have a full set of proposals, and all our co-design problems are updated for the new proposals, we can look at what options we have. Some options will be dominated by certain other options, but because we are not necessarily weighing any *careset* over another, many proposals will be on an antichain together. So we bring the space back to Bragtown residents to hear their thoughts about what possible worlds they prefer.

We create a consensus space that integrates all the *caresets* that have been provided by the engagement process, the co-design problems, and the proposals.

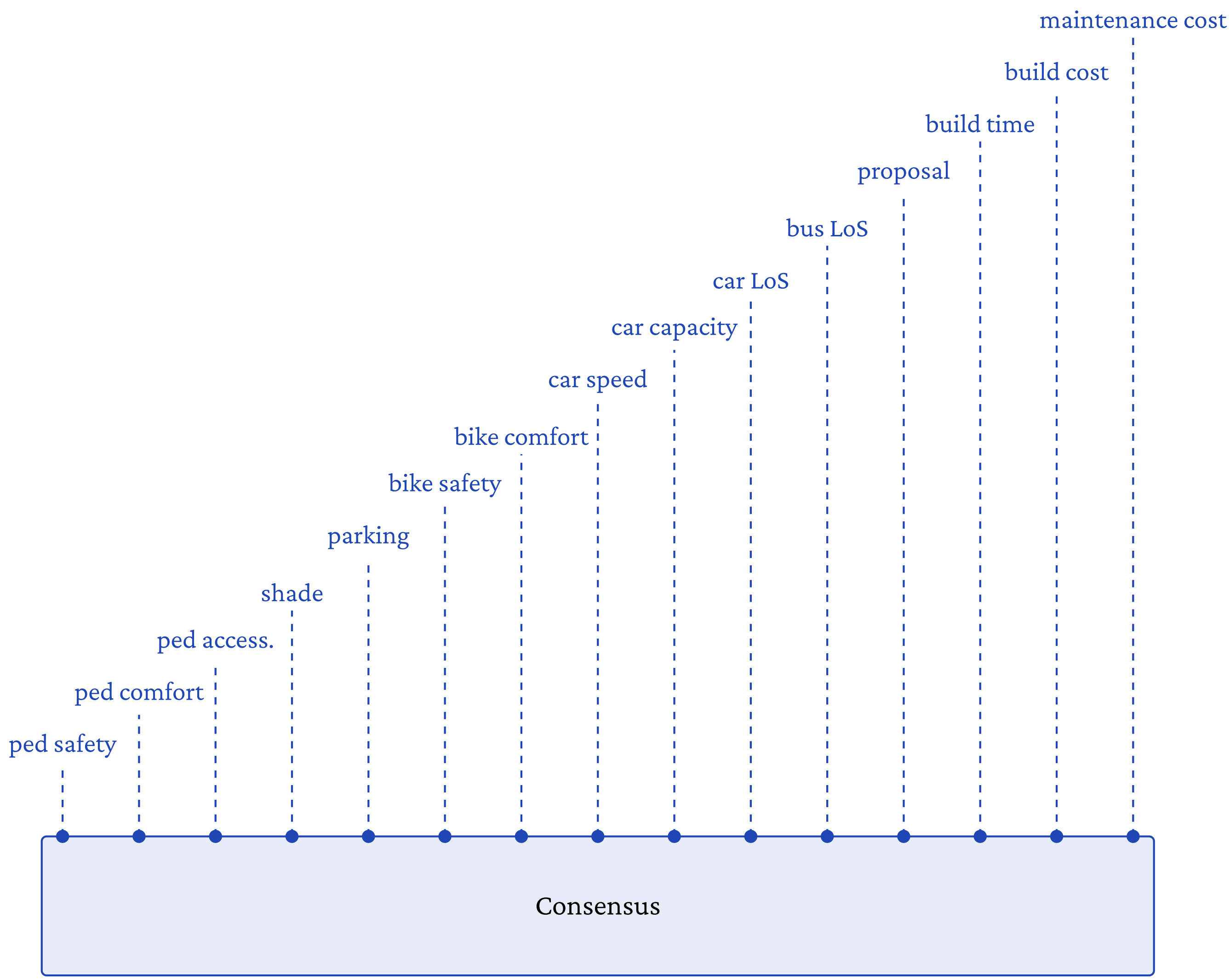


Figure 11.7: Consensus space

This consensus space will let Bragtown residents state their opinion on all the *caresets* that have been surfaced by comparing feasible proposals head to head.

As they provide a consensus, we will be able to see the *careset* of all options ordered according to their values. From there, we can surface the best possible antichain of options.

This consensus space creates a 15-dimensional tradespace of proposals, something that cannot be visualized easily, but with our *careset* structure we can browse the antichain of options. Then if users want to see a tradespace of the best options available to them, they can select two cares to use as a projection for a two-dimensional tradespace.

Residents, the City, and all other stakeholders can look at this Pareto Frontier of options that have been chosen and see the trace back to every *careset* and co-design problem. This traceability makes the participatory process trustworthy and transparent.

Likely, a few options will trickle to the top antichain, and the consensus threshold could be reduced to include more options in the top antichain. Elected officials will likely be the best fit to decide the threshold to produce a final Pareto Frontier of options.

It is from these options that elected officials can select a proposal to implement for the North Roxboro Street corridor.

11.7 Roxboro Project Carespace

We can put all the pieces together to see the full *carespace* we've created for the Roxboro project.

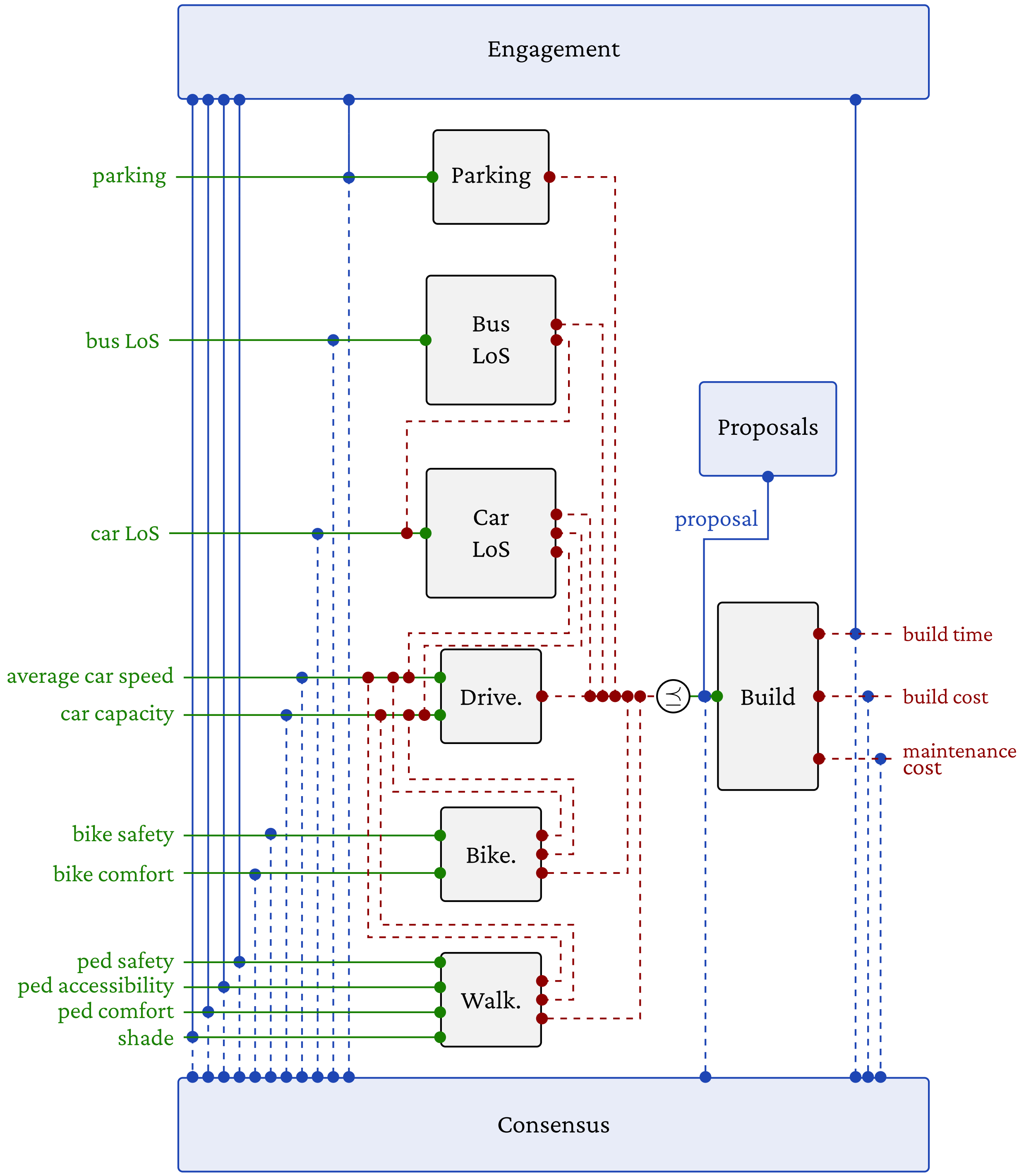


Figure 11.8: Roxboro EPCODOT space

11.8 A Few Things to Consider

11.8.1 More Composition

We could still add many more *caresets* and decompositions to provide more detail about the ramifications of each proposal and the potential benefits. Time to cross Roxboro depends both on crossing distance and on average distance between crosswalks. Those using wheelchairs and mobility devices may have particular priorities that they should be allowed to surface. Many more *caresets* could be added.

This Roxboro space could be expanded and connected to other spaces that stretch beyond the scope of this project.

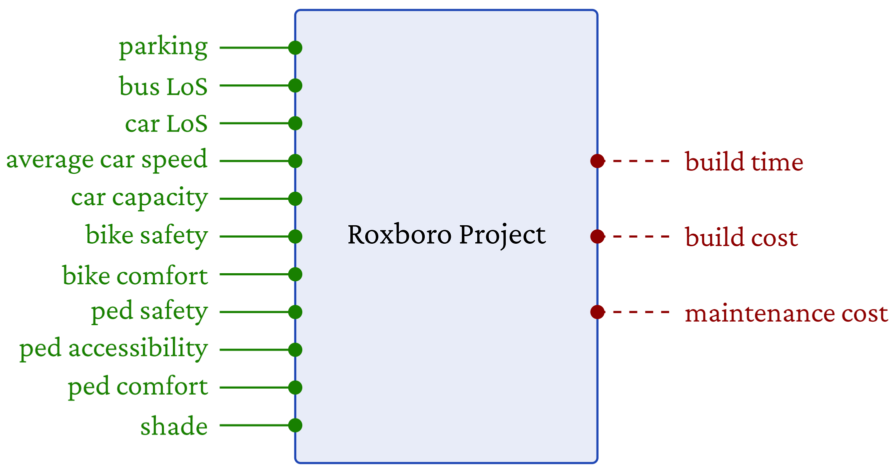


Figure 11.9: Roxboro Project *carespace*

We can imagine taking this project *carespace* and embedding it in a larger EPCODOT space of all the transportation related projects across Durham, composing the Roxboro space alongside other projects. The Roxboro project does not live in a vacuum.

11.8.2 Equity

How can we make EPCODOT equitable? This is where the use of consensus spaces and *caresets* must be managed well. By default, consensus spaces treat every individual with equal weight. Maybe local residents' opinions should be weighed more heavily than those of residents in nearby neighborhoods. Maybe residents who are disenfranchised or low-income should have their priorities emphasized.

It would be possible to create multiple consensus spaces for different populations and merge those consensus *caresets* with unequal weights that reflect equitable priorities. This would be one potential quantitative attempt to approach a complex problem, but there could be other ways as well.

The flexibility of EPCODOT gives it expressive power. But for projects that affect the public good, it is important that public institutions ensure that the distribution of power within EPCODOT reflects the right priorities.

11.8.3 Direct Democracy

It is fair to question how right it is to involve residents so deeply in a public project. Residents are not experts and may not have the time or expertise to understand the nuances of complex projects. This is a reasonable critique of direct democracy, and one of the reasons we elect representatives to make decisions on our behalf.

But the purpose of EPCODOT and monotone co-design is precisely to solve for this weakness of direct democracy. When different stakeholders have different levels of understanding and expertise, we should be able to compose and decompose spaces to provide each stakeholder exactly the interface that is most legible and understandable to them and speaks most directly to their cares.

Bragtown residents should not have to know what the Average Annual Daily Traffic is on Roxboro to be allowed to have an opinion on whether Roxboro should have a bike lane. Residents should be allowed to propose adding a bike lane, or widening a sidewalk, or adding a traffic light. And in turn, residents should see what the practical trade-offs are that relate to their proposals.

Domain experts should be able to translate the trade-offs into terms (i.e. cares) that residents understand and relate to, either through their existing models and systems or through their claims and research. Maybe a new bike lane will increase congestion, maybe there will be fewer parking spaces, or maybe research shows it might increase foot traffic for local businesses. It should be possible to express those hypotheses as co-design problems that residents understand.

Residents should also be able to dig deeper. If a proposal is not implemented, it should be clear where it failed to meet feasibility, or what the trade-offs were that put another proposal over it.

11.8.4 Design Better Together

From the Roxboro example, we can begin to imagine how community-led collaborative design might look in EPCODOT.

Tactical urbanism is a method used by communities to improve their neighborhoods and public spaces to be safer and more accessible, often with minimal and unsanctioned interventions (Lydon and Garcia, 2015). EPCODOT could formalize the tactical urbanism process for neighborhoods and provide a dynamic and multi-layered service for neighborhoods to experiment with low-budget, high-impact changes.

While we have not tested it here, EPCODOT can likely be used to model more general collaborative design projects. David Spivak's concept of Plausible Fictions describes a framework for collaboratively building futures that reflect our values (Spivak, 2024). EPCODOT may provide a path to implementing this framework.

The fundamental capabilities of EPCODOT (composition, computation, collaboration) together provide a unique take on how communities might be able to take on grassroots efforts to build better futures.

Conclusion

What happened?

We have seen that our mathematical language uses partially ordered sets to represent what it means to care. We have seen that our language offers a formal and visual expression of concepts like preference, ethics, and design.

We have seen that the Experimental Public Co-Design of Tomorrow can use our language of care to adapt monotone co-design to express not just what we *can* design, but what we *should* design. We have seen that EPCODOT can leverage the compositional and computational capabilities of monotone co-design, extending it with collaborative interfaces that accept proposals and integrate preferences.

Who cares?

Our mathematical language of care offers an a priori attempt to unify preference, ethics, and design under the umbrella of care. The language relates to existing theories in social choice theory, decision theory, and design theory but attempts to make explicit the connections between them while starting from a foundation of ordered sets. The language also attempts to provide novel unifying visual intuitions for concepts across the three forms of care discussed.

We hope that our language of care offers interesting ideas to the fields of applied category theory, decision theory, and ethical philosophy.

Through our cases, we have shown that EPCODOT has the fundamentals of a usable software for participatory planning. If put into practice, residents will be able to submit proposals, prioritize their cares, and directly impact project outcomes. EPCODOT takes a small step towards realizing the promise of participatory planning.

We hope that EPCODOT offers an interesting groundwork for urban planners, local officials, community organizations, advocacy organizations, researchers, and entrepreneurs looking for ways to improve the way local governments make decisions and engage communities.

For Tomorrow

Our work today has many limitations. Some discussed throughout, others not yet explored. These limitations also represent opportunities for future research.

Here are a few of the more important and interesting shortcomings of our work today, and what might be worth pursuing tomorrow.

A Mathematical Language of Care Tomorrow

Care*: The Category of *Caresets* and Actions** We defined this category in Chapter 3: Ethics. This category may be a version of *Pos*, the category of partially ordered sets and monotone maps, but there may also be differences. Exploring the ramifications of ***Care would be mathematically and philosophically interesting. Questions for future research include:

- What forms of preference are not represented in ***Care***?
- How do individual and communal preferences coexist in ***Care***?
- Can ethics be described as the 2-category form of ***Care***, where the 2-morphisms are evaluations of actions?
- What is the difference between ***Care*** and a category of *Possible Worlds* and *Actions*?
- Do these concepts really compose, or does category theory require too much restriction to comprehensively represent these forms of care?
- What category-theoretic structures and theorems can be translated to ***Care***, and what real-world phenomena do they describe?

Formalize connection to social choice and decision theory We nodded to social choice and decision theory, acknowledging that many concepts in *caresets* are equivalent to concepts in those theories. But, for the most part, we kept our own terms and built up our own ideas from scratch so we could avoid the work of having to do gymnastics to make existing theories fit the diverse fields we discussed (preference, ethics, design). But this leaves a number of questions that could be answered by future research:

- Are there actually important differences between the language of partially ordered sets we've built and the structures of preference relations over alternatives that are used in decision theory?
- Does the concept of a *Possible World* actually offer anything useful compared to standard descriptions of states and alternatives?
- Will Amartya Sen be upset with us?

The Experimental Public Co-Design of Tomorrow Tomorrow

Empirical validation with active projects and communities This work does not test EPCODOT's effectiveness for improving outcomes compared to traditional participatory planning methods. We have only shown what EPCODOT looks like in theoretical usage, albeit on a real problem. The natural next step is to take this software prototype and deploy it in a real test scenario, learning from the deployment challenges of other digital urban planning tools (Omrany et al., 2025). Questions for real-world testing include:

- Is the learning curve for composition of design problems and carespaces too high to be clear to residents?
- Can residents trace how their preferences impact final decision outcomes?
- Is the burden of building up a representative model of complex urban systems too high?
- Does partial order voting make it easier to vote for different types of projects or proposals?
- Does EPCODOT make it easier for residents to submit proposals that are taken seriously?
- Does using EPCODOT lead to better outcomes compared to traditional methods of community engagement and participatory planning?
- How should equity and inclusivity be managed in a practical deployment of EPCODOT?

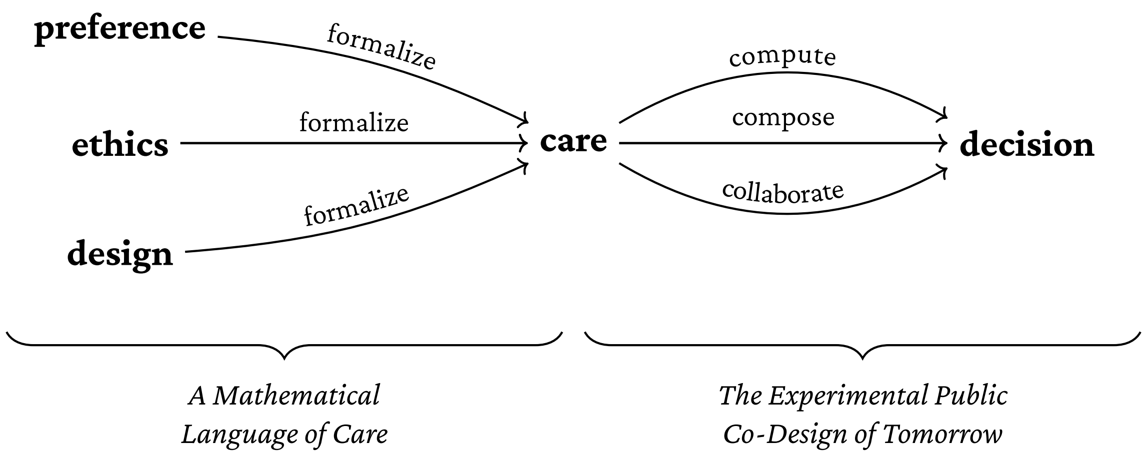
Improve computational capabilities The current EPCODOT prototype offers a layer above monotone co-design and the mcdp solver, but it does not make any improvements or optimizations for EPCODOT's unique use case. Questions that can be pursued include:

- Is there a way to adapt the mcdp solver to work more efficiently with consensus and proposal spaces?
- If complex, composed systems are represented in EPCODOT, can the solver be run at multiple levels of abstraction concurrently to save time on subsequent runs?
- Can the visual primitives of *caresets* and corresponding tradespaces be better integrated with the solver's output to give users a clearer understanding of what the Pareto-optimal solutions are?
- Are there better preference aggregation algorithms we should use for deriving consensus?

Coda

How can cities make better decisions?

Today, we have only taken our cities one small step towards better decisions. But next time we arrive where the sidewalk ends, we might give a thought to our *caresets* and the Experimental Public Co-Design of Tomorrow.


preference
ethics
design
formalize
formalize
formalize
care
compute
compose
collaborate
decision
A Mathematical
Language of Care
The Experimental Public
Co-Design of Tomorrow

The Plan

—

"Does anything matter?"

As the weather improved, our sunset walks got longer. She took that as an invitation to point at life's big questions. Yesterday, it was whether birds could laugh. But she couldn't prove it either way.

"Dad, does anything matter?"

"I guess it's very hard not to care."

We wondered together if that was true.

—

Acknowledgments

Thank you to everyone who cared, even a little. It meant a lot.

Gioele Zardini, Fábio Duarte, for letting me go for a walk.

Bailey Flanigan, Yujun Huang, Brian Hedden, David Spivak, for your attention and thoughts.

Brian Moser, Joan Rubin, for your support and encouragement.

Bill Foley, for 1,001 answers.

Bragtown, Durham, the Bicycle and Pedestrian Advisory Commission, for your efforts.

Célia Chaussabel, for your objectiles.

Areen Bahour, Sarah Baik, Carrie Deline, Andrew Grabowski, Eugenia Kim, Sophie Leamon, Rachel Lin, Steven Maa, Vivek Sahay, Arman Shantayev, for your camaraderie.

John Fate Faherty, for your tao.

Arman, Monique, for your love.

Buket, Metin, for everything.

Bibliography

Afzalan, N. and Muller, B. (2018). Online participatory technologies: Opportunities and challenges for enriching participatory planning. *Journal of the American Planning Association*, 84(2):162–177.

Alexander, C., Ishikawa, S., and Silverstein, M. (1978). *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York.

Alonso, L., Zhang, Y. R., Grignard, A., Noyman, A., Sakai, Y., ElKatsha, M., Doorley, R., and Larson, K. (2018). Cityscope: A data-driven interactive simulation tool for urban design. use case volpe. In Morales, A. J., Gershenson, C., Braha, D., Minai, A. A., and Bar-Yam, Y., editors, *Unifying Themes in Complex Systems IX*, pages 253–261, Cham. Springer International Publishing.

Arnstein, S. R. (1969). A Ladder Of Citizen Participation. *Journal of the American Institute of Planners*, 35(4):216–224.

Arrow, K. J. (1963). *Social Choice and Individual Values*. Number 12 in Cowles Foundation Monographs. Yale University Press, New Haven, 2nd edition.

Borge, R., Balcells, J., and Padró-Solanet, A. (2023). Democratic disruption or continuity? analysis of the decidim platform in catalan municipalities. *American Behavioral Scientist*, 67(7):926–939.

Broome, J. (1991). *Weighing Goods: Equality, Uncertainty, and Time*. Economics and Philosophy. Basil Blackwell, Cambridge, Mass.

Censi, A. (2015). A mathematical theory of co-design. *CoRR*, abs/1512.08055.

Chang, R. (2002). *Making Comparisons Count*. Routledge, New York.

Durham Bicycle and Pedestrian Advisory Commission (2024). 2023 walk audit of n. roxboro st. - report & recommendations. Durham, NC City Government. Report by Durham BPAC on pedestrian infrastructure assessment.

Fishburn, P. C. (1970). *Utility Theory for Decision Making*. John Wiley & Sons, New York.

Fishburn, P. C. (1985). *Interval Orders and Interval Graphs: A Study of Partially Ordered Sets*. John Wiley & Sons, New York.

Forester, J. (1999). *The Deliberative Practitioner: Encouraging Participatory Planning Processes*. MIT Press, Cambridge, MA.

Frankfurt, H. G. (1988). *The Importance of What We Care About: Philosophical Essays*. Cambridge University Press, Cambridge.

Gonçalves, S. (2014). The Effects of Participatory Budgeting on Municipal Expenditures and Infant Mortality in Brazil. *World Development*, 53:94–110.

Hare, R. M. (1963). *The Language of Morals*. Oxford University Press, 1st edition.

Harsanyi, J. C. (1955). Cardinal welfare, individualistic ethics, and interpersonal comparisons of utility. *Journal of Political Economy*, 63(4):309–321.

Held, V. (2006). *The Ethics of Care: Personal, Political, and Global*. Oxford University Press, New York.

Helm, B. W. (2001). *Emotional Reason: Deliberation, Motivation, and the Nature of Value*. Cambridge University Press, Cambridge.

Hovmand, P. S. (2014). *Community Based System Dynamics*. Springer, New York.

Hume, D. (1739). *A Treatise of Human Nature*. John Noon, London. Modern edition: Oxford University Press, 1978.

Kant, I. (1785). *Groundwork for the Metaphysics of Morals*. Cambridge University Press, Cambridge. Translated by Mary Gregor (1997).

Keltner, D. and Stamkou, E. (2025). Possible worlds theory: How the imagination transcends and recreates reality. *Annual Review of Psychology*, 76(1):329–358.

Korsgaard, C. M. (1996). *The Sources of Normativity*. Cambridge University Press, Cambridge.

Latour, B. (2008). A Cautious Prometheus? A few steps toward a philosophy of design. In *Networks of Design: Annual International Conference of the Design History Society*, pages 2–10, University College Falmouth, Cornwall, United Kingdom.

Lydon, M. and Garcia, A. (2015). *Tactical Urbanism: Short-term Action for Long-term Change*. Island Press, Washington, DC.

Mansuri, G. and Rao, V. (2013). *Localizing Development: Does Participation Work?* Policy Research Report. World Bank.

Mazzarello, M., Duarte, F., Mora, S., Heine, C., and Ratti, C. (2025). Data slots: trade-offs between privacy concerns and benefits of data-driven solutions. *Humanities and Social Sciences Communications*, 12(643).

Noddings, N. (1984). *Caring: A Feminine Approach to Ethics and Moral Education*. University of California Press, Berkeley.

OECD (2020). Innovative citizen participation and new democratic institutions.

Omrany, H., Mehdipour, A., Oteng, D., and Al-Obaidi, K. (2025). The uptake of urban digital twins in the built environment: A pathway to resilient and sustainable cities. *Computational Urban Science*, 5(20).

Rittel, H. W. J. and Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy Sciences*, 4(2):155–169.

Savage, L. J. (1954). *The Foundations of Statistics*. John Wiley & Sons, New York.

Sen, A. (1970). *Collective Choice and Social Welfare*. Holden–Day, San Francisco.

Simon, H. A. (1988). The science of design: Creating the artificial. *Design Issues*, 4(1/2):67–82.

Small, C., Bjorkegren, M., Erkkilä, T., Shaw, L., and Megill, C. (2021). Polis: Scaling deliberation by mapping high dimensional opinion spaces. *RECERCA. Revista de Pensament i Anàlisi*, 26(2):1–26.

Smart, J. J. C. and Williams, B. (1973). *Utilitarianism: For and Against*. Cambridge University Press, Cambridge.

Spivak, D. I. (2024). Plausible fiction. Topos Institute Blog.

Tronto, J. C. (2013). *Caring Democracy: Markets, Equality, and Justice*. New York University Press, New York.

U.S. EPA (2024). Public participation guide: Selecting the right level of public participation.

Veldhuis, G. A., Smits-Clijsen, E. M., and van Waas, R. P. M. (2024). Techniques to enhance the public policy impact of qualitative system dynamics models. *System Dynamics Review*, 40(4):e1758.

von Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1st edition.

Waddell, P. (2002). Urbansim: Modeling urban development for land use, transportation, and environmental planning. *Journal of the American Planning Association*, 68(3):297–314.

Weinstein, M. C. and Stason, W. B. (1977). Foundations of cost-effectiveness analysis for health and medical practices. *New England Journal of Medicine*, 296(13):716–721.

Wigner, E. P. (1960). The unreasonable effectiveness of mathematics in the natural sciences. *Communications on Pure and Applied Mathematics*, 13(1):1–14.

Zardini, G. (2023). *Co-Design of Complex Systems: From Autonomy to Future Mobility Systems*. PhD thesis, ETH Zurich.

Zardini, G., Lanzetti, N., Censi, A., Frazzoli, E., and Pavone, M. (2023). Co-Design to Enable User-Friendly Tools to Assess the Impact of Future Mobility Solutions. *IEEE Transactions on Network Science and Engineering*, 10(2):827–844.

Well, looks like I'm stuck here. Why don't you go on without me, and I'll be catching up with you somewhere along the line.