

Innovation Languages: Why and How

Author: Ivan Jureta

Affiliation: Fonds de la Recherche Scientifique - FNRS, Belgium, and University of Namur, Belgium

Contact: ivan@ivanjureta.com

Status: Unfinished draft, writing in progress. Feel free to share as-is, but contact me before making changes.

1 Technical Languages for Innovation

This book is an instructions manual, on how to make technical languages during, and to support innovation.

Why do that? Isn't it a fool's errand?

Technical languages represent stable knowledge, ideas that are recognized and agreed upon by many, in a community of experts. Such languages reflect the community's agreement on which terms to use, their definitions, and how to use them. Medicine, law, engineering, and their many specialties all have technical languages. We learn them through training and participation in the appropriate community.

Innovation, on the other hand, by definition involves uncertainty. We innovate when we create new ideas and things that turn out to be useful.

When we do this, we start from chaos in some sense, from disorder and confusion which do not fit our motives. We work to find and make patterns in chaos, to produce order, to change the environment we are in, so that it is less different than what we want it to be. Innovation also involves discovery. It starts with one set of motives, one rough understanding of how things are, and we learn along the way, new preferences, new facts, new misbeliefs, we change direction, often in unpredictable ways.

Does it make sense, then, to create technical languages when innovating? Shouldn't we wait until the outcomes of innovation are tried and tested, accepted, stable enough to warrant effort in producing rigorous representations of what we know about these innovations? If we can only make good technical languages when knowledge is stable, why invest that effort when we only have imperfect information, vague ideas, and repeatedly shed, through trial and error, our mistaken assumptions?

I argue in this book that this does make sense. There is a long story to tell, but the crux of the argument is that people who are doing innovation are creating a new language as they innovate, they do this anyway, and when we capture that language, when we represent it explicitly, when we write down the new words and definitions for new ideas and things, rather than leave them implicit, we are also capturing knowledge generated through innovation, however imperfect and unstable it may be. By doing this, I argue, we support the innovation process - we make communication and coordination easier in the team that does innovation, this helps go through trial and error faster, it makes it easier to grow that team, to convey outcomes of innovation to others, and to build a solid foundation for a future technical language that will grow around innovation outcomes and relate these outcomes to existing bodies of knowledge, communities of experts, and natural language of those who benefit and those who lose out from innovation, but are not interested in understanding its intricacies.

In short, I am arguing that it makes sense to be rigorous about language that takes shape informally and early on when generating new ideas, and to worry about this as soon as the innovation process kicks off. This can be done without impeding that innovation process, as long as expectations are realistic, the team that does innovation knows how to do this, and how much. The balance is not easy to strike. Being too rigorous can impede innovation, rather than support it.

I will use the term "innovation language" for the kind of technical languages that this book is about. They are very much alike technical languages in general, but depart in one key respect. An innovation language is a rough technical language, imperfect in many ways, capturing and conveying unstable knowledge. If the outcomes of innovation are useful, if they become relevant, then the innovation language eventually graduates to a technical one.

2 Outline

This book combines two narratives.

One focuses on the practicalities of how to make and maintain innovation languages. It is directed towards anyone who is involved in innovation, and is interested in capturing relevant information used and generated during innovation processes.

The other narrative takes an academic angle. It is about why we should want to make innovation languages, how much of it helps or hinders innovation, and how this relates to established work on the creation of informal and formal languages. I discuss the relationship of innovation languages to basic notions in epistemology (data, information, belief, knowledge, among others), knowledge representation, problem solving and decision making, and ontology engineering. The aim is not only to provide arguments to support the instructions I provide in the practical narrative, but also to give a broader picture in which innovation languages and their production fit. This also points you to literature and results in various research fields, to content that is good to be aware of, when making and maintaining innovation languages.

3 Innovating Out of Chaos

What could innovation languages possibly be and do? There are so many kinds already - natural languages, technical languages, planned languages, constructed languages, programming languages, domain-specific programming languages, formal and informal specification languages, and so on. With so many types of languages already there, how is this new, and why it matters?

Chaos, when used colloquially, denotes confusion and a complete absence of order. Innovation starts with people motivated to bring order to chaos.

As they do this, as they go on to find patterns, understand opportunities and problems, overcome confusion and see through disorder, produce new intangible ideas and new tangible things, they create new words, terms, new rules for communicating about these problems and solution, throughout the innovation process that produces order from chaos. They create new languages, even if these lean on natural languages they may be using, and technical languages they may mobilize as they draw on established, expert knowledge.

This language that they create, through which they make the process and outcomes of innovation their own, which they communicate with to coordinate in their collaboration, is what an innovation language is.

Innovation languages are created by teams making something new - new businesses, new products and services, academic research, R&D projects. Those are only some of many settings, environments, in which

innovation happens.

No one in the team did it before and no one has a ready-made answer. They see outlines of an opportunity or a problem. There is some partial, vague, imprecise, inaccurate information that there is a problem, and there is enough of a motive to solve it. But neither is the problem sharply defined, nor is there a ready made solution.

Within that chaos, the team sets out to understand what is there, what could be done, decide what to do, how to do it, towards a solution. I participated in a dozen such cases over the last decade, and these chaotic situations sometimes took months to be shaped into problems, then months and years of trial and error, of abandoned solutions, refinement, iterations, and overall, learning about what gives desired results. Sometimes this ended successfully, often not.

Turning chaos into some appearance of order is hard. It uses substantial intellectual effort, draws on various kinds of knowledge, requires careful observation, and much trial and error. It requires coordination, despite little being known. It involves many people. It takes a lot of time. Even when it looks quick from afar, many I have worked with say that one year doing this feels about the same as a few years doing known things, solving again, in the same way, problems that have already been solved before. Doing this means living with risks that are hard to evaluate and manage.

Thinking that there is a silver bullet, a method that works every time, is naive at best, and damaging otherwise. If there was such a thing, then there would be, for example, flawless entrepreneurs and venture capital businesses, there would be no need to worry about risk of failure when funding research and development, or even basic research. None of this is the case.

The point so far, is that we innovate our way out of chaos into order, and to do this, we have to communicate about the process and outcomes -new ideas and things - created through the innovation process. Since these are new things, we need, at the very least, new names for them, and these new names are part of the nascent innovation language, which is specific to the innovation we are involved in.

Why does this matter?

4 Learning from Chaos

If there is one aspect, a process woven into all those experiences, all instances of innovation processes that I have been involved in, it is learning. And I mean this in the usual, casual sense, "the acquisition of knowledge or skills through study, experience, or being taught" as it says in the online Oxford English Dictionary. [OED: Learning](#)

A problem with doing something new, is that you learn most from experience, via trial and error. Studying helps, as does being taught, or at least I am convinced of those from teaching students, but it only goes so far. It helps guide the trial and error, it helps select what to trial, and set up the test in the first place, but - again, because we are talking about new things, new ideas arising out of confusion and disorder - it does not give good predictions, and certainly cannot substitute to actually experiencing the errors and successes of the trials, then tirelessly predicting, generalizing, extrapolating from small samples of past outcomes, until results of some next trial bring down the last seemingly stable observation.

Much has been written on learning, of humans and machines, and this book does not add more there.

Instead, I take learning as a given phenomenon, a complex set of processes that happen, and it does not matter much if it is good or bad, if it can be better, or how. At least not this early in the book.

What I am interested in, is how to capture what was learned about the chaos as team progresses with innovation, so that it does not get forgotten, and that it serves as a foundation on which to continue learning and building value. And I see the need for capturing language, for documenting and structuring it, as central to this.

Specifically, I argue in this book that we should carefully observe and preserve the specific language that gets developed, as people on the team, who are doing innovation in their progress against chaos, describe, discuss, and in general communicate about what they are doing and creating, about their understanding of the problems to solve, of solutions, and of how to get to make and use these solutions.

You fight chaos with innovation. To quote OED again, innovation is synonymous apparently with all of the following:

"change, alteration, revolution, upheaval, transformation, metamorphosis, reorganization, restructuring, rearrangement, recasting, remodelling, renovation, restyling, variation, new measures, new methods, new devices, novelty, newness, unconventionality, modernization, modernism, a break with tradition, a shift of emphasis, a departure, a change of direction". [OED:Synonyms of innovation]

OED's imprecision is not due to, but certainly mirrors difficulties in research on innovation. We do not understand it well. In research on innovation in businesses, for example, it is that elusive process from ideas to new or improved products, services, organizations themselves even, in order to generate more economic value.

"Creativity is the production of novel and useful ideas by an individual or small group of individuals working together. [...] Organizational innovation is the successful implementation of creative ideas within an organization." [Amabile, 1988]

In a broader context, outputs and aims differ will, but the start is generally recognized to be ideas.

This lack of understanding is not strange, however, since if innovation does go against chaos, it goes against that which, by definition, eludes understanding.

For the rest of this book, there is no need to get into debates of what innovation exactly is, or is not. What matters, is that learning is the process to find and make some order in chaos, and as that happens, and especially if it gives good results, we call that process innovation. If innovation was successful, then it is simply that the learning that went on in that process, gave results that we recognized as useful.

Learning happens in various ways, different individuals will learn differently, and that variety only grows as we look at how teams innovation, or, as I argued above, collectively learn order in chaos.

My central assumption, grounded in anecdotal observation and participation in innovation processes over the last decade, is that however a team learns, whatever the motive that initiated their innovation process, whichever idea it started from, the team will create their own language to talk about the new, to describe - all in order to collaborate - on the structures they find and make in the chaos that they are working against.

They will speak, of course, English, French, Mandarin, Swahili, or any natural language they choose. But because they are making, having, talking about, working on, extending, and changing new intangible ideas and tangible things, they will create new words, new terms, new rules on how to communicate about these new ideas.

Telephone was a new word, even if long time ago, just as genetic mutation was more recently; neither were part of the hunter-gatherer vocabulary. The next time you sign up for insurance, for example, of any kind, look up a section typically called Glossary, Terminology, or Definitions, and you will see a language on its own, understandable to some extent through the natural language it uses in definitions of terms, but different from it.

Those outlines of a technical language itself grew out of innovation, that time in how to generate economic value while managing risks for others.

That is what innovation languages are. This book is about how to make them, perhaps more accurately, how to record and analyze them, since they are there anyway. Recording and analysis have a simple aim, which is to generate more value for the team, over the long run. How and why exactly will is a topic interwoven through the coming chapters.

5 What Language Has to Do with Innovation

When working on something new, you need a way to speak, write, and more generally communicate your ideas about it, so that you can coordinate your work with others, understand what they think and want to do about it, agree and disagree, and act, plan, and learn together.

New ideas need new names, to distinguish and not confuse them with what is already there. Those names, the words you choose to use, how you use them, and what you want and agree for them to mean, make up an innovation language.

As coordination is necessary, communication is too, and teams consequently invent their own terms, phrases, ultimately their own language to talk about, to explain, and define innovations that they create.

Even if teams will inevitable make a new language as they go along, such a language is rarely documented, its terms and phrases are seldom precisely defined, and much hinges on everyone's personal history with the team - you understand only if you were around when ideas are communicated, new terms introduced, and old ones discarded. Much of it is informal.

While informality may look faster and easier to everyone involved, it is also a waste. You can see that speed and ease, that you prefer early on, as an accumulation of debt to repay later, especially if the innovation is successful, that is, needs to spread outside the team.

The informal innovation language grows and becomes more complex as the underlying innovation and knowledge develops. At some point, innovation starts being relevant beyond the team itself, needs to be described, explained, justified, its mechanisms presented, so that it can be accessible to others. At that time, the team's innovation language needs to be learned by others, and its relationships with natural language clarified. One typical reality check for such teams, is when they interact with legal experts, for example, to check the regulatory compliance of their new business model, product, service, or other; at that time, they need to start repaying the debt I mentioned above, as they need to explain the innovation to experts who have their own languages, without an established translation to the team's.

The notion of innovation language comes from my experience and research on how to capture and maintain knowledge in teams that work on unstructured problems. Innovation often starts from unstructured problems and opportunities.

About a decade ago, when I started doing fieldwork in teams that worked on innovations, I did something unusual. As soon as we would start working, even when even brainstorming was going into many directions, I insisted on writing down keywords, key terms that popped up in those sessions, and more generally in all team communication, and sending my definitions of these terms to the team, expecting their feedback. Most were surprised, and many saw this as futile, given how fast ideas were changing. And it was true that many terms were thrown away, and that many definitions changed along the way. But I insisted on keeping even a simple lexicon, or terminology available to all, and asking for all to know it and voice any disagreements at any time.

Value grew over time, to the point that in many businesses where I did fieldwork, this became an accepted practice. I no longer had to promote it. Design, engineering, technical and other documentation started including pointers to a centralized terminology, which was actively maintained and commented. Most interestingly to me, this looked like a sticky tool, something that people accepted without formal requests to do so. It became the backbone for knowledge management. I certainly did not discover the value of glossaries, terminologies, and dictionaries of technical terms; their worth is clear in established disciplines, and many businesses, especially those with a long history and a complicated and large set of products and services, know how important it is to be careful about their internal technical language.

Where I was able to observe the use of innovation languages, it became simpler to talk about teams' innovations with interested parties, lawyers, future users, marketers, and any stakeholder group which started interacting with the team over the lifecycle of the innovation.

Over the last decade, I worked with dozens of teams which built up their innovation languages. That led to this book.

6 What an Innovation Language Is

You and I learned to use one or more natural languages while growing up, and then through basic education. English, French, Dutch, German, Swahili, Mandarin, Hindu, are only some among some 5-7,000 languages considered as ordinary live languages in use.

It is said that a natural language develops through use, without it being designed, or its use planned.

A technical language, in contrast, is designed to enable communication about a specific body of knowledge. Medicine has its own technical language; law too. There is a technical language in any field of inquiry which has been around for some time, and where a body of specialized knowledge is built up, maintained, and developed further.

An innovation language is a kind of technical language, in which the underlying body of knowledge is unstable, partial, limited, or otherwise deficient. Even if that knowledge is limited, new, and exploratory, an innovation language is intended to push those involved in growing and improving that knowledge, to be precise, accurate, and clear in communication, even when they are communicating about imprecise, inaccurate, and unclear knowledge.

With an innovation language, you try to impose precision, accuracy, and clarity in communication much earlier than what you know is anywhere near those benchmarks, while what you know is imprecise, inaccurate, and unclear.

This only sounds counterintuitive.

Suppose that you and I are tasked with creating a tool that predicts stock index values, e.g., the value of the S&P index (Standard & Poor's 500). The typical approach would be to fit a model to historical data, and so we need a source of that data. Our innovation language already needs to include at least three terms: Predicted Value, Historical Data, Prediction Model.

At this time, we do not know much about the Predicted Value, other than it will be the output of the Prediction Model; we do not know if that value is a single value, a range of values, if it is for a specific time unit, or for a time range, and so on. We have the same issues with Historical Data, which may be only data over a few years, or a few decades, if it will be better to have all decades or only the recent ones; and for the Model, we really have no

idea which type of statistical model this could or should be. In the end, most is unknown, but by agreeing on the definitions of the three terms, we would agree not to call the Prediction Model a "predictive model" in some communication, a "prediction function" in others, to use "historical data" in all communication, instead of only "data" in some, "statistical data" in others. We would agree not to casually use synonyms, at least.

At the same time, we would know what needs to be further defined and worked on - specifically, we need to work on identifying, finding, understanding the specifics of available Historical Data, and we could agree equally well on how to approach work on the Prediction Model, and understand what kinds of Predicted Values that model needs to produce. Inevitably, since so little is known that early on, the perspective on the problem and its solutions may change substantially, words thrown out, definitions revised, perhaps to the point of restarting a new innovation language. The potential for those is inevitable.

In more abstract terms, having a team make and maintain a innovation language forces its participants into creating explicit conventions on at least two fronts: one is how they cut up the problem and solution spaces in the work they are doing, because they need to decide on terms and therefore categories that matter to them (in academic terms, they are making decisions related to ontology), and the other is that they adhere to a common vocabulary, so as to remove some of the synonymy and ambiguity that reflects the lack of stable and structured knowledge of the problem and solution space.

7 What an Innovation Language Looks Like

The simplest innovation languages look like lists of terms, with a definition of each.

Let's look at an example.

In 2017, a team that combined business consultants, product designers, and software engineers, was asked to invent a process which speeds up the negotiation of simple business consultancy contracts. They focused on contracts involving two parties, a consultant and a client, who need to agree on a set of deliverables, dates for the delivery of each, and price. The blueprint of the process would then serve for the production of a software, which would support clients and consultants to negotiate such contracts.

The default way to negotiate such a contract, is for the consultant and client to hold meetings, and communicate in other ways, until the client was confident enough to want to invest, and the consultant was clear enough on what needs to be done, by when, and within which budget. It is a common, but unstructured process, thus hard to repeat, measure, and improve. It can take varying amounts of time, and end in no contract at all.

The team needed specific terms for the parties in this process, the roles they can have, and the object of their negotiations. This gave the following terms.

- Client: Party which is interested in having a Service completed by a Consultant.
- Consultant: Party which performs a Service for a Client.
- Service: The collaboration of one specific Consultant and one specific Client, and under the rules and guidelines set out by the Service Contract to which they have both agreed.

Definitions embed design decisions. A Client is not necessarily someone or something that has agreed to have a service delivered, but anyone who shows interest. If Client were defined, instead, as something that exists only if a Service exists, then one obvious implication is that we need one more term, for that party in negotiations which is interested in getting the Service.

Who can be a Client, and who a Consultant? Can a party which is in one Service a Client, be a Consultant in another Service? Answers to these questions influence the design of the process, that is, are part of decisions that shape the negotiation process.

Notice how a plain definition, one in a natural language dictionary, of client no longer works for this team. If a client is by default "a person or organization using the services of a lawyer or other professional person or company" [Google] then not every client is a Client, since the default definition seems to require a contract to be in place. How else would the person or organization be using services?

The point is not to debate if client or Client are better for this team, in relation to the problem they are solving. The point is, instead, that once the term Client is defined by that team, then their communication about Clients requires everyone to know that definition. If not, then there will be misunderstandings.

So far, the innovation language of this team has three terms. None of them shows how we decided to speed up that unstructured negotiation process. Our approach was to add constraints to it, to make it more specific by taking the following assumption seriously: negotiation can be sped up if it is clear what needs to be negotiated, i.e., which parameters of the Service, and if negotiations can only last a fixed amount of time. We introduced the notion of Negotiation Phase.

- Negotiation Phase: Part of the negotiation process which has a fixed duration and a structured set of Negotiation Phase Outputs.

Having decided that the negotiation process would have some unknown number of Negotiation Phases, we had to decide which phases it could have. The first that seemed appropriate, was a phase during which the Consultant and Client would set goals of their collaboration. This led us to introduce the Goal Setting Phase, as a type of Negotiation Phase, and to define Goal Setting Outputs.

- Goal Setting Phase: A Negotiation Phase, which lasts up to 14 calendar days, whose aim is for the Consultant and Client to approve a nonempty set of Service Goals.
- Goal Setting Outputs: A statement of goals of a Service.

This language quickly grew to a few dozen terms, and their definitions embedded our decisions in designing the negotiation process. As we progressed in the design, terms were revised, changed, some were removed.

One of the changes involved introducing the notion of Project, as a type of Service, defined as follows.

- Project: Service in which:
 - Client's Key Expectation is to have the Consultant achieve specific Goals on each of the agreed Targets;
 - Consultant's Key Expectations are:
 - To receive Consultant Project Fee, AND
 - To create Accounts, which, if the Consultant manages (during the Tail) to convert into Deals, will yield the Consultant the payment of the Deal Commission.

Note the additional terms this involves, namely Key Expectation, Target, Account, Tail, Deal, Consultant Project Fee, and Deal Commission. I leave them undefined here.

What I want to illustrate with the Project term, is how terms are used to write rules about the artifact being designed, and how this influences the definitions of these same terms.

As we were progressing in the design of the negotiation process, the scope of the problem widened. The team was tasked to propose a process which covered structuring and monitoring of a Consultant's execution of a Project for the Client. In other words, when contract is signed, execution starts, and we wanted to look at how to improve the monitoring of that execution. The motive was that we wanted to use that monitoring to evaluate the reputation of both the Client and Consultant, then use these for future recommendations of Consultants to Companies, and vice-versa.

At some point in the design, we converged to the following structure of Project Phases.

- Project Phases are:
 - Call For Help, in which a Client posts an Opportunity;
 - Proposal, in which a Consultant responds to the Opportunity;
 - Contracting, in which the Consultant and Client negotiate the Targets, Goals, Tasks, and Project Fee;
 - Delivery, in which the Consultant does Tasks to achieve Goals, and creates Accounts for the Client;
 - Approval, in which the Client confirms that Goals are met and Project is completed, and provides feedback on the Consultant;
 - Continuation, in which the Client decides if and how to continue collaboration with the Consultant.

We grew the set of terms, refined it and their definitions, while deciding how the processes should work. That "how" means setting rules the process should satisfy. We used the terms to define the rules. Here is a sample, on Project execution.

1. In Call For Help and Proposal Project Phases, Matching Consultants are shown to Client only in an anonymized format, so that their full name and contact information cannot be known to the Client.
2. Contracting Phase lasts one week, with the possibility on demand to extend this by one or more increments of two calendar days each.
3. Project duration is one calendar month.
4. When posting an Opportunity, Client must provide at least one Target, and one Goal for that Target.
5. Project Agreement includes one or more Approved Tasks.
6. In an Opportunity Response, Consultant provides:
 1. For each Goal in each Target, at least one of the following:
 1. Acceptance of Task as-is, that is, as set in the Opportunity;
 2. Rejection of Task as-is;
 3. At least one Sub-Task;
 2. Optionally, number of Man-Hours that the Consultant will dedicate to complete the Task or Sub-Task;
 3. List of zero or more Accounts that the Consultant;
 4. Project Fee.
7. In an Opportunity Response, Consultant can input relevant experience to the Targets, Goals, and Tasks in the Opportunity Response.
8. An Opportunity Response must include Effort Estimate and Hourly Price.
9. Client should be able to provide feedback and ratings on all Projects.
10. Retainer as a mechanism for extending Services should only be available to Client and Consultants who are involved already for several months in an ongoing Service.

These rules show how the terms become building blocks of the artifacts that the team is designing. In turn, rules placed further constraints on the meaning we wanted the terms to have. Rule 8, for example, tells us that an

Opportunity Response must specify Effort Estimate and Hourly Price, without these it is not an Opportunity Response.

This gets us to an interesting question, if rules are part of the language, or are something else, something built through language use. How, in other words, do we separate the innovation language from the ideas it is used to create, the artifacts it is used to make?

Rule 8 is not the only one leading to this question. Rule 10 says when a Retainer can be used. Should Rule 10 be part of a definition of Retainer? Rule 3 defines the allowed duration of a Project, isn't it, then part of the definition of Project?

To the extent that each rule restricts the interpretation of a term, it must be part of the definition of that term. If that is the case, then would everything written in the innovation language somehow constitute that language? I return to this later, in Chapter XX.

8 Four Core Ideas

The book is grounded in four simple ideas.

Language Reflects Knowledge

The first idea is that language reflects knowledge, which means this: languages you use say a lot about what you do and do not know.

Try explaining to someone else what cancer is; if you are an oncologist, your terms and phrases will reflect a more sophisticated, more elaborate, and hopefully more accurate, precise, and clear set of ideas about what cancer may be, than what I might say, not being an oncologist, or a medical professional at all. I will come back later to why this may be, but the point should be clear and is hardly controversial.

The important consequence of language being a reflection of knowledge, is that if you do want to convey what you know to others, and you want to do it in a precise, accurate, and clear way, then you need a technical language to do so, a language which can convey potentially complex relationships and dynamics between a myriad of ideas which are not in the world, so to speak, but in minds of people in the community which grows that knowledge.

That is, ideas are not there, separate from these people, for others to inspect, but exist to the extent that they put them there; and putting them there involves communication, with language being a critical tool. How else would that knowledge be exchanged?

Technical Languages for Specialist Knowledge

The second central idea is that general-purpose natural languages, such as English, French, Mandarin or Swahili, at least those parts which most of us learn through basic education, are not good enough to capture the nuances needed for being precise, accurate, and clear when communicating ideas that convey deep, specialist knowledge.

Law has its own specific language for speaking about legal entities for example, and while such terms as "limited company" may hardly be new, they have been introduced to language because it was necessary to designate specific abstractions, invented at some period in history, which did not exist before that.

There are two important implications of this second idea:

1. Technical languages are not accidental creations, but artifacts designed with specific purposes in mind.
Therefore, there must be good practices to follow when doing that design.
2. The number of technical languages out there does not depend on some intrinsic property of any natural language, such as English or German, but on the varieties of bodies of knowledge that expert communities develop over time.

Technical Languages Can Be Cheap to Grow

If there are thousands of natural languages, there are many more technical languages, any one of which borrows from live or dead natural languages, but adds its own flair.

Think of specialties in medicine, many of which use some of the dead Latin, and much from a variety of live natural languages.

The title of this book is a vague prediction, that soon there will be a billion new languages, not of any kind, but technical languages that are cheap to build and grow, and serve teams of specialists to convey, agree on, preserve, and collectively expand abstractions that make up deep, specialized knowledge accumulated and carefully grown over time.

While the design of specialized languages is common in some disciplines, such as law, medicine, and engineering, even there it looks like a painful and expensive experience. My aim is to show that it is not, provided that it starts early, and proceeds incrementally through dedicated routines, to mirror the growth of the knowledge that the relevant community is exchanging.

Innovation language Are Draft Technical Languages

The fourth important idea is that there are benefits to having draft, or rough technical languages, where terms and phrases convey new and immature ideas. This contrasts to technical languages, where the language is made to capture mature ideas.

This is really the only difference between a technical and a innovation language. Innovation languages include terms which are not necessarily a reflection of agreed or stable knowledge. A innovation language looks like a rough, draft technical language, something that can become, or has parts which can become a technical language.

9 Why Innovation Languages are Interesting Now

Technical languages have been around as long as there was a need to convey complex ideas in a precise, accurate and clear way. Medical knowledge in Ancient Greece developed its own language. It is hard to argue against the need to make and the practicality of using technical languages. Would they have been around for that long, if there were no strong merits?

But this book is not about any type of technical language. It is about immature technical languages, made along the way, as we solve unstructured and complicated problems. These are proto-technical languages. They may look as-if they are technical languages, but reflect and are used to communicate about much less developed, and less table knowledge than a typical contemporary technical language does.

Now, a technical language itself cannot come out of thin air, but has to develop over time and in response to the maturing of, and change in the underlying body of knowledge. All technical languages had to start somewhere,

somehow, and they were - then and there - also grounded in immature knowledge, looking perhaps much like innovation languages.

Why is now any different from any of the past times, for thinking more about how to make Innovation languages? Or, in other words, to think more about how to make proto-technical languages?

Three factors stand out.

Knowledge Assets

One factor that favors cheaper and faster production of Innovation languages, is the increasing emphasis on an old idea, that people in organizations acquire, hold, apply, and grow knowledge, and that they have incentives to do this in a systematic way.

A systematic approach is needed to manage risks, if only of that knowledge disappearing too easily.

Systematically maintaining a innovation language, as one of the records of the specific organizational knowledge, can generate value. It facilitates seeking protection under intellectual property laws, even if it does guarantee that this will be successful.

Having even the simplest innovation language for innovations, helps at the very least the communication with legal experts, and the identification of prior innovations, which may already benefit from legal protections.

How and Why Work Is Automated

The enduring interest to automate work, and how we do that, are the second and third factors which favor the creation of Innovation languages.

To automate work, you need to break it down into simpler tasks, and then automate these. To automate a task, you need a combination of data about past executions of that tasks and their outcomes, knowledge to distinguish good from bad executions, and knowledge about how to perform that task.

In some cases, such as the matching of people's names to their faces in photos, the task is easier to automate by having an algorithm learn from training data, where faces and names are already correctly matched. At least in part, this is due to our inability to make explicit, to document and explain how people recognize faces in images.

In the majority of cases still today, data about past executions is neither available, nor good enough for automated learning. We fall back to explicit rules, and to building task-specific software according to these rules.

To build the software, then, we need to describe the task and its execution. Just like we did for the consulting contract process in Chapter 2, which was subsequently automated.

Since we are automating by building software, we need to be precise about the operations that the software needs to perform, and the data it needs to perform them on.

If we need to be precise in software code, to make sure it does what we want it to do, then we also need to be precise in the specifications of that software, and finally, in the way that these specifications ensure that the software will automate the task. A innovation language moves across these artifacts, being present in the specifications and in code.

The language becomes the backbone for task descriptions, later in software specification documents. The software itself used these terms, not only in its user interface, but in code, as classes, relationships, methods

designed into the software followed the innovation language.

10 New Ideas Use both Old and New Words

Chapter summary: On the relationship between new words and old words in a innovation language, how old words build up definitions of new ones.

How to explain a satellite to a child? How about using one of these definitions:

Satellite: "An artificial body placed in orbit round the earth or moon or another planet in order to collect information or for communication." [\[OED: Satellite\]](#)

"In the context of spaceflight, a satellite is an artificial object which has been intentionally placed into orbit. Such objects are sometimes called artificial satellites to distinguish them from natural satellites such as Earth's Moon." [\[Wikipedia: Satellite\]](#)

If a child already has an idea of artificial bodies, orbit, earth, planet, of what collecting information and communication are about in relation to these notions, then the first definition might be a good start. It is not an explanation, but we will come back later to the differences between definitions and explanations.

How about this, which NASA says is for children ages 10+?

"A satellite is a moon, planet or machine that orbits a planet or star. For example, Earth is a satellite because it orbits the sun. Likewise, the moon is a satellite because it orbits Earth. Usually, the word "satellite" refers to a machine that is launched into space and moves around Earth or another body in space." [\[NASA Knows! \(Grades 5-8\)\]](#)

Or this version, from NASA too, for a slightly younger audience:

"A satellite is an object that moves around a larger object. Earth is a satellite because it moves around the sun. The moon is a satellite because it moves around Earth. Earth and the moon are called "natural" satellites. But usually when someone says "satellite," they are talking about a "man-made" satellite. Man-made satellites are machines made by people. These machines are launched into space and orbit Earth or another body in space." [\[NASA Knows! \(Grades K-4\)\]](#)

If you had to choose one, which one would it be? Are either of the first two good candidates?

You can think of these questions by looking at definitions backwards, from the definiens (which defines) to the definiendum (the term being defined).

Take the Wikipedia Satellite definition again. Notice it mentions the term context, a complicated term, an abstraction that bears multiple precise definitions, which are likely to vary across specific disciplines; i.e., context has a different definition in different technical languages. For example, Wikipedia gives this, for context in computer science:

"In computer science, a task context is the minimal set of data used by a task (which may be a process or thread) that must be saved to allow a task to be interrupted, and later continued from the same point." [\[Wikipedia: Context \(computing\)\]](#)

This is certainly not a unique or conventional understanding of the notion of context in computer science altogether. There are specific, and substantially different definitions of context in, e.g., knowledge representation,

requirements engineering, mobile computing, and so on. Here is a definition from mobile computing research:

"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." [Abowd, Gregory D., et al. "Towards a better understanding of context and context-awareness." International symposium on handheld and ubiquitous computing. Springer, Berlin, Heidelberg, 1999.]

In short, context has a definition that varies between technical languages. In philosophy, the relationship between context and knowledge is an ongoing and longstanding debate, see Epistemic Contextualism for one of lines of argument [SEP: Epistemic contextualism]. The Oxford English Dictionary gives this generalist definition:

"The circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood." [Oxford English Dictionary: Context]

Being twisted in this way, context is a not a term to use, when defining or explaining a satellite to children.

The important takeaway here, is not that it is hard to explain a satellite to a child, but that how you define a term depends on two things:

- what you assume the user of that definition already knows, and
- what they may want to do, once they learn the definition or explanation you give them.

You may, sometimes in error, assume that the child does not know enough for you to use the Oxford English Dictionary definition. If you also think the child needs it simply out of curiosity, then perhaps either of the two NASA definitions will do.

But what if an astrophysicist asks an engineer for a definition of "satellite", not out of curiosity, but because they are working together on the design of a new satellite, and they are trying to align their ideas. None of the definitions above might work; perhaps their definition would speak of gravitationally bound objects in orbits, or use other technical terms more appropriate to their background knowledge and the task at hand.

This gets us to the second and third takeaway:

- it makes little sense to think about one term and its definition in isolation from everything else, and
- a technical language leans on natural language to some extent, even if only by using terms from natural language in definitions of technical terms.

The following illustrates how the Satellite definition from the Oxford English Dictionary presumably leans on other natural language terms. Some are emphasized below.

Satellite: "An **artificial body** placed in **orbit** round the **earth** or **moon** or another **planet** in order to **collect information** or for **communication**." [Oxford English Dictionary: Satellite]

Each emphasized word (and others) is a plain English word, so to speak, that is, it is a part of English. If we take it that the Oxford English Dictionary provides good definitions for many words that make a large subset of English language, then this Satellite definition leans on the dictionary as the guideline for how to read each of the words in the definition.

But that definition of Satellite is taken from the Oxford English Dictionary, so by design, it leans on other words from there.

We go to a different example now, where some words in the definiens are not in the Oxford English Dictionary.

"Biosafety: The prevention of large-scale loss of biological integrity, focusing both on ecological and human health. Set of measures or actions addressing the safety aspects related to the application of biotechnologies and to the release into the environment of transgenic plants and organisms, particularly microorganisms, that could negatively affect plant genetic resources, plant, animal or human health, or the environment." [\[UNDP: Biosafety\]](#)

This definition of Biosafety comes from the United Nations Development Programme (UNDP), in a glossary for documentation on sustainable development. Below, terms set in bold are not defined in the Oxford English Dictionary, and so for all practical purposes here, not part of natural language. At the same time, these terms are not defined in that UNDP glossary; terms set in italics are not part of natural language, but are defined in the glossary. The rest of the terms are part of natural language.

"Biosafety: The prevention of **large-scale loss** of **biological integrity**, focusing both on ecological and human health. Set of measures or actions addressing the safety aspects related to the application of biotechnologies and to the release into the environment of transgenic plants and organisms, particularly microorganisms, that could negatively affect plant genetic resources, plant, animal or human health, or the environment."

To summarize, in the Biosafety definition above, there are two terms outside natural language and outside the glossary itself ("large-scale loss" and "biological integrity"), and only "Biosafety" is itself defined in the glossary, no other term from the definiens is.

Is that a good definition?

This is hard to say, as "good definition" itself is an open topic, to which I return in Chapter XX. It is clear, however, that there are several things to think about here:

- "Biosafety" is defined independently from the rest of the glossary,
- "Large-scale loss" is undefined within the topic of the glossary, i.e., the glossary does not help reduce the vagueness of that term in any way, and
- "Biological integrity" is undefined in the glossary, and since it is not part of natural language, the authors in the glossary must have leaned on a technical language, but did not point to it in the glossary.

If these look like quibbles, they are not. Here is a sample of the activity around getting to and agreeing on a technical definition of "biological integrity":

"[United States Environmental Protection Agency] convened a symposium [...] on the integrity of water soon after passage of PL 92-500 [Federal Water Pollution Control Act], but no clear definition of biotic integrity emerged. Many authors advocated the use of a holistic perspective. Karr and Dudley [...] argued that the "integrity" objective encompasses all factors affecting the ecosystem and developed a now widely quoted definition of biological integrity as the ability to support and maintain 'a balanced, integrated, adaptive community of organisms having a species composition, diversity, and functional organization comparable to that of natural habitat of the region.' A more recent paper defined ecological health (an umbrella goal, the maintenance of which motivates virtually all environmental legislation) as follows: 'a biological system...can be considered healthy when its inherent potential is realized, its condition is stable, its capacity for self-repair when perturbed is preserved, and minimal external support for management is needed' [...]." [\[Karr, James R. "Biological integrity: a long-neglected aspect of water resource management." Ecological applications 1.1 \(1991\): 66-84.\]](#)

Disagreement is common when specialists work on definitions. The problem is not disagreement, but disconnect, which happens if the discussion involves parallel monologues, each pushing its own definition, and no dialogue where there is convergence to an agreement on the definition of each debated term.

Going back to takeaways, these examples illustrated how a definition, even of a new technical term, depends on natural language and therefore of old terms and their readings accepted as part of natural language, however unclear they may be. As long as your technical language leans on natural language, which is not bad in itself, you are building something that is intended to be more precise, accurate, and clear from pieces which may not be that at all. In the worst case, you are simply changing where vagueness, ambiguity, and other deficiencies are coming from.

New ideas need new words. At the same time, new words are anchored in old ones.

11 Definitions Record Choices

Chapter summary: How the design of a innovation language reflects our problem-solving and decision-making.

A technical language is a record of past decisions. This is a simple idea with significant consequences.

Let's start with a simple example, from the consulting contract case introduced earlier. We had the following definition.

- Service: The collaboration of one specific Consultant and one specific Client, and under the rules and guidelines set out by the Service Contract to which they have both agreed.

Consider these alternative definitions. Differences from the original one above are emphasized.

- Service1: The collaboration of one **or more** specific Consultants and one specific Client, and under the rules and guidelines set out by the Service Contract to which they have both agreed.

Service1 allows any number of Consultants to work for a Client under the same Service Contract. This may be better or worse than the original, but in either case, it can only be so if we decided that it should be so. In other words, the original definition excludes this possibility, because we decided to exclude that possibility. The main reason was that it would require coordination between Consultants, and that could have an impact on the speed at which we get to an agreed Service Contract.

Service2 below, is another alternative.

- Service2: The collaboration of one specific Consultant and one specific Client.

Service2 is less specific than either Service and Service1. You can see it as a less mature variant of each of these. It fails to mention the contract as the framework for collaboration. By doing that, it fails to say how we go from negotiation to collaboration, the transition which happens because there is agreement on a contract.

The point here, is that definitions, especially about new things, and especially if these new things are abstractions, ideas, are not given somehow to us. We make them, and as a result, we define them.

To clarify that given/made distinction, here is the Oxford English Dictionary definition of giraffe.

Giraffe: "A large African mammal with a very long neck and forelegs, having a coat patterned with brown patches separated by lighter lines. It is the tallest living animal." [\[OED: Giraffe\]](#)

This definition conveys what one generally sees, when looking at the animal called giraffe. Giraffes, in other words, are given - it is not man-made, but a product of evolutionary processes. Alternatively, if we used a definition that lists a different set of properties, as below, we are still in the same case: a giraffe has these properties not because we made them so, i.e., the definition is not a reflection of our design of giraffes, but of our observation of giraffes.

Giraffe [[African Wildlife Foundation: Giraffe](#)]:

- "Scientific name: *Giraffa camelopardalis*
- Weight: Males: 1,930 kilograms (4,254 pounds) Females: 1,180 kilograms (2,601 pounds)
- Size: 5.7 meters tall from the ground to their horns (18.7 feet)
- Life span: Average 10 to 15 years in the wild; recorded a maximum of 30 years
- Habitat: Dense forest to open plains
- Diet: Herbivorous
- Gestation: Between 13 and 15 months
- Predators: Humans, lions, leopards, hyenas, crocodiles"

But things are very different for our Service term above, since it refers to an abstraction, one we fully designed - a team chose its properties and from there followed its definition.

There are also in-between cases, where the definition is of an observed complex process. "Globalization" is an interesting case, and below are four definitions that the United Nations Economic and Social Council cites in their [[Compendium of basic United Nations terminology in governance and public administration](#).]

"Globalization is increased global integration and interdependence. It has a multidimensional character: economic, political, social and cultural. It is characterized by unprecedentedly rapid flows of goods and services: private capital; circulation of ideas and tendencies; and the emergence of new social and political movements [...]

The process through which an increasingly free flow of ideas, people, goods, services and capital leads to the integration of economies and societies. Major factors in the spread of globalization have been increased trade liberalization and advances in communication technology [...]

Globalization describes the growth in international exchange and interdependence. With growing flows of trade and capital investment, there is the possibility of moving beyond an international economy (where 'the principle entities are national economies') to a 'stronger' version of the globalized economy in which 'distinct national economies are subsumed and rearticulated into the system by international processes and transactions' [...]

Globalization can be defined as universalization. In this use, 'global' is used in the sense of being 'worldwide' and 'globalization' is 'the process of spreading various objects and experiences to people at all corners of the earth'. A classic example of this would be the spread of computing, television etc."

The World Bank definition looks more abstract than the others, in that it defines globalization as a type of universalization (which I leave undefined here), while the other three definitions talk of integration and interdependence.

Below is another definition of "globalization", as it appears in the United Nations UNTERM, "a multilingual terminology database maintained jointly by the main duty stations and regional commissions of the United Nations system."

"Expansion of global linkages, organization of social life on global scale and growth of global consciousness, hence consolidation of world society. Most particularly (per Friedman), the term is used to refer to the seemingly "inexorable integration of markets, nation-states and technologies to a degree never witnessed before -- in a way that is enabling individuals, corporations and nation-states to reach around the world farther, faster, deeper and cheaper than ever before." This phenomenon also involves the spread of the gospel and forms of market capitalism to virtually every country in the world."

[UNTERM:Globalization]

Is any of these, or is there, at all, the definition of globalization?

It is a complex process, which we do not know how to measure. There is disagreement on when, where and how it started. We do not know what cues to look for, which data is representative of it. We do not know if it has reached a peak, what reaching a peak means anyway, and we have few ideas on how globalization might end.

"Today few doubt the reality of globalization, yet no one seems to know with any certainty what makes globalization real. So while there is no agreement about what globalization is, the entire discourse on globalization is founded on a quite solid agreement that globalization is. Behind the current and confusing debates about its ultimate causes and consequences, we find a wide yet largely tacit acceptance of the factuality of globalization as such, as a process of change taking place 'out there': even otherwise constructivistically minded scholars tend to regard globalization as an undeniable and inescapable part of contemporary experience" [Bartelson, Jens. "Three concepts of globalization." International Sociology 15.2 (2000): 180-196.]

Which of these would you choose?

This question depends on what, if anything you want to do about, with, or in relation to the phenomenon that is defined.

More specifically, a choice between definitions depends on the extent to which their differences make a difference in your decision-making and problem-solving.

Going back to our small-scale consulting contract case, the choice of one of those three definitions given earlier (Service, Service1, and Service2) does make a difference. "Service" requires a different negotiation process than "Service1", and both can involve different processes than "Service2".

Definitions indeed record choices. If they define what is given and easily observable, such as giraffes, then definitions are more a record of choices of properties to pay attention to, among all potential (and perhaps unknown) properties of giraffes. If, instead, they define abstractions which we fully imagined and designed, then the definition reflects our choices of properties to give to these abstractions. And in many cases, we will be creating abstractions to communicate about potentially complicated combinations of phenomena, as we saw for globalization, where properties reflect, again, what we see through observation, but perhaps too, some wishful thinking, what we want these phenomena to be.

A definition is the bridge of the new to the old, we piece together the new from old notions. The new, by being there, by being defined, becomes a tool or vehicle for change, for moving not only ideas from old to new, but also learning to make new choices and act differently than before. Conceptualizations of "globalization", its definitions, work the same way, embodying observations, choices, mashing them up with expectations, making it a vehicle, not a simple reflection of what seems to be a shared experience. Bertelson's summary, of his discussion of the concept of "globalization" nicely sums up the multiple roles a definition can have, especially for

complex phenomena, which lend themselves to analysis from different angles, and through tools from various bodies of knowledge.

As he puts it, the term "globalization" links the old and the new (label [A] in the quote below), denotes the interest in the phenomenon itself [B], tells us about what can be [C] and when it will be [E], and points to a vague reality [D], i.e., does not refer to something tangible or otherwise easily accessible, for which you and I could rapidly come to same or similar enough observations.

"Through its various connotations, the concept of globalization functions as [A: a mediating link between the modern world with its crusty social ontology and the brave new world that remains inaccessible and unintelligible not only to the subscribers to that ontology but also to the believers in global change as well]. Not only is globalization a moving target for social inquiry, but it also [B: signifies the movement of that inquiry itself]. The concept of globalization thus makes modern political experience meaningful while simultaneously releasing political expectation from the strictures that the very meaningfulness of this experience has imposed on political imagination. It does so through a double gesture: by [C: projecting expectations onto the global while simultaneously making those expectations constitutive of globality as a point of reference and convergence]. In this respect, the logic of the concept of globalization resembles that of the concepts of civilization and revolution as they were shaped before and during the French Revolution: [D: these concepts also lacked stable referents, but functioned as vehicles of social change by signifying change in its purest, most necessary and irreversible form: change as the condition of possible objects and possible identities in a possible future]. And like these concepts, globalization does not represent a mere prognosis for the future, but a prophecy in quest for self-fulfillment. So far from being here to stay, the metaphors of globalization will perhaps die when the concept has fulfilled its destabilizing task, that is, [E: when globalization has become something that goes without saying and therefore no longer stands in need of being spoken about]." [\[Bartelson, Jens. "Three concepts of globalization.\]](#)

[International Sociology 15.2 \(2000\): 180-196.\]](#)

12 Definitions Make Choices

Chapter summary: On how definitions make it easy do do some things, and hard to do others.

If you commit to a definition, meaning you choose one definition of a term over others, then you also take additional commitments. It is yet another case of buying one and getting more than you asked for. How much of this more do you know? Do you want this additional baggage? These are the two central questions.

Law is an interesting domain when thinking about definitions. This is for two reasons: one, it insists on using terms in a precise and consistent way, and two, how these terms are used has substantial effects on people's lives.

Consider the hard case, of "terrorism". How would you define it?

Here is how terrorism is defined in the United Kingdom Terrorism Act. The version below is from 2018. Read the colors as follows: blue text is the original definition, and it was included in law in 2000; text labeled [A] was added to the original definition in 2006, and the updated definition was published in the Terrorism Act 2006; text labeled [B] was added in 2008, and published in Counter-Terrorism Act 2008.

"Terrorism: interpretation.

- (1) In this Act 'terrorism' means the use or threat of action where
 - (a) the action falls within subsection (2),

- (b) the use or threat is designed to influence the government [A: or an international governmental organisation] or to intimidate the public or a section of the public, and
- (c) the use or threat is made for the purpose of advancing a political, religious [B:, racial] or ideological cause.
- (2) Action falls within this subsection if it
 - (a) involves serious violence against a person,
 - (b) involves serious damage to property,
 - (c) endangers a person's life, other than that of the person committing the action,
 - (d) creates a serious risk to the health or safety of the public or a section of the public, or
 - (e) is designed seriously to interfere with or seriously to disrupt an electronic system.
- (3) The use or threat of action falling within subsection (2) which involves the use of firearms or explosives is terrorism whether or not subsection (1)(b) is satisfied.
- (4) In this section
 - (a) 'action' includes action outside the United Kingdom,
 - (b) a reference to any person or to property is a reference to any person, or to property, wherever situated,
 - (c) a reference to the public includes a reference to the public of a country other than the United Kingdom, and
 - (d) 'the government' means the government of the United Kingdom, of a Part of the United Kingdom or of a country other than the United Kingdom.
- (5) In this Act a reference to action taken for the purposes of terrorism includes a reference to action taken for the benefit of a proscribed organisation." [\[UK Terrorism Act 2000\]](#)

This is a complicated, but carefully made definition of a complex and important phenomenon. It is a poorly understood phenomenon. Any definition, therefore, however attentively made, will have limitations.

One of the possible consequences of applying this definition is that some acts, which normally do not look like cases of terrorism, still could be categorized as terrorism.

"Civil disobedience, public protest and industrial action are among the activities that could fall within the definition. These types of activities should be excluded from any definition of terrorism. [...] Unlike the definition in the Australian Criminal Code (and its State equivalents), the New Zealand Terrorism Suppression Act, the Canadian Criminal Code and the 2003 South African Anti-Terrorism Bill, the United Kingdom definition does not contain an exception in favour of advocacy, protest or industrial action. The legislation simply requires the purported terrorist to have committed an act (such as endangering a person's life, or seriously damaging property), and to have committed that act in furtherance of a political, religious or ideological cause with the aim of influencing the government or intimidating the public (or a section thereof). This encompasses groups whose methods are generally non-violent and who do not aim to intimidate or to coerce the government or the public. For example, a long-running nurse's industrial dispute where staffing levels in public hospitals have been seriously reduced could 'create a serious risk to the health or safety of the public', within the meaning of s 1(2)(d) (as could the industrial actions of other essential services, such as fire officers, police, and so forth). If the strike were directed towards convincing the government to increase pay and conditions in public hospitals then this could also satisfy both the 'political cause' and the 'influencing government' requirements, in s1. Similarly, a mass student protest against the deregulation of university fees by the British Government could also fall within the definition of terrorism." [\[Golder, Ben, and Williams George. "What is Terrorism-Problems of Legal Definition." UNSWLJ 27 \(2004\): 270.\]](#)

Theses are interesting potential consequence, regardless of intentions that led to the definition.

Going back to less complicated topics, and more in the context of innovation languages, your choice of one definition over another has consequences on the structure of the problem space, of the solution space, and on the innovation process. This means that how you define something has an impact on what problem you are going to be solving - that is the problem space part - the potential ways you will be looking at to solve it - the solution space part - and on how you go through the formulation of the problem and its resolution - the innovation process part. In other words, it can affect really all that goes on during innovation.

In an earlier Chapter, I introduced the case where we were designing a process for contract negotiation. One of the issues that was highlighted early on, was that such contract negotiation tends to take an unpredictable amount of time. They can last long, and if they last too long, they may simply not lead to the contract at all. So one of the aims that was taken for granted, was that the process should shorten the time to contract, or more broadly, to the decision to sign or not.

How do these early assumptions shape the problem space? Consider two potential definitions for the contract negotiation process.

- Contract Negotiation 1: Process in which parties agree on terms of collaboration.
- Contract Negotiation 2: Process in which a party that needs to have assignments performed communicates with potential parties that can execute these assignments, until she has either agreed with one or terms for collaboration towards the execution of these assignments, or has agreed with none of the considered parties.
- Contract Negotiation 3: Process in which parties agree on terms of collaboration within two calendar weeks from the time the asking party requested a proposal from the supplying party.

The first definition is the least constraining. It leaves a lot open. Nothing is said about how many parties can be involved, how much time they can take, what they can collaborate on. The second definition is clear about the number of parties that can be involved, and how the process ends. Only the third definition mentions the duration of the process, and is precise about it, but is open regarding the number of participants.

If the innovation team adopts the third definition, they still need to decide on the number of participants, among others. But can they allow any number of participants? Does allowing fewer participants make two weeks a long time for negotiations? Is two weeks too short, when there are many participants? Does the number of participants influence duration? Is it the other way around, does the duration limit the number of participants? Should the process differ in maximal duration if it has many participants? Which numbers would require which durations?

If the team does indeed adopt the third definition, and makes it immutable, then this affects substantially the possible designs of the negotiation process. And it does so in ways which may not be easy to see as early on in the innovation process. We did, in fact adopt a definition which included a time limit. Because of this, we had to restrict communication between the parties. If we left it open, letting them freely communicate, we believed that they would not converge to a go/no go decision quickly enough. This, in turn, required that we provide template contracts, where it was clear which exact parameters could be negotiated. And that, then, required that we restrict communication between parties only to the negotiation of values of those parameters. Yet another unanticipated consequence of this, is that it can work only for highly structured contracts, where these negotiable parameters are few and well known. For example, a monthly retainer amount and a price per hour, together with the list of tasks. That is a far reaching consequence, because it affects the size of the potential market for the process, and the software that was to be made to support that process.

This does not mean that a more open, i.e., less constraining definition is better. If the team takes seriously the first definition instead, then it may ignore the relationship of duration to outcome, and more specifically that the duration of negotiations influences negotiation success.

All these issues seem obvious as you read this text, but they are hard to see and analyze when you are involved in the innovation process. It is often only in retrospect that their merits and drawbacks become clear.

In that respect, you can see a definition as a bundle of predictions. In the case I mentioned, these are that two weeks are a good duration for contract negotiation, that the number of participants needs to be two or more, and so on. These are predictions about what will make the outcomes of innovation useful.

An analysis to do, that helps understand the implications of a definition on the problem and solution spaces, is to identify choices in the definition, and identify at least one other option for each. I did it below on the three definitions.

- Contract Negotiation 1: Process in which (*one, two, more parties*) (*agree, disagree, cancel negotiations*) on terms of (*delegation, collaboration*).

I identified three choices in the first definition. One is the number of parties, with four options (either one of the three, and all three as the fourth). The second is the outcome of negotiations, with four options as well. The final is the object of negotiations, with two options. You could have gotten other choices and other options for each. What the analysis highlights, is that the first definition commits to one out of $4 * 4 * 3$ possible combinations; the problem space is restricted to the chosen options in each choice, and any solution will have to solve a problem in that part of the problem space.

Consider now the second definition.

- Contract Negotiation 2: Process in which a party that needs to have (*one, more assignments*) performed communicates with (*one, more than one potential parties*) that can execute (*all, subset of these assignments*), until she has either agreed with (*none, one, more than one party*) on terms for collaboration towards the execution of (*all, subset of these assignments*).

Above, I highlighted five choices, and a problem space of $3 * 3 * 3 * 3 * 3$ combinations. You could easily enlarge this problem space, by allowing participants to make their own decisions. That would look like this.

- Contract Negotiation 4: Process in which a party that needs to have (*one, more assignments, party's own choice of number of assignments*) performed communicates with (one, more than one potential parties, the number of parties decided for that contract) that can execute (*all, subset of these assignments, a minimum set by the asking party*), until she has either agreed with (*none, one, more than one party, number chosen by the asking party*) on terms for collaboration towards the execution of (*all, subset of these assignments, minimal set required for the asking party to accept the contract*).

And this is only for a rough definition of a single term. You can imagine how this becomes complicated to think about, if we start looking at choices in two, three, or more realistically, innovation languages with a few dozen, and sometimes hundreds of terms. Even if you dedicated effort to the analysis of this kind, to each definition alone, you would still have to do this for interactions of choices which are in definitions of different terms, but depend on each other.

Same analysis is now applied to the third definition.

- Contract Negotiation 3: Process in which (*two, more than two, number st by asking party*) parties agree on terms of collaboration within (*one, two, custom number chosen by asking party*) calendar weeks (*from the time the asking party requested a proposal from the supplying party, from the time the first supplying party responded to the request for proposal, either of these as st by asking party*).

Again, notice how a seemingly simple definition can cause headache, as the number of choices and options in each start to multiply.

Let's take a step back. What goes on in this analysis? I said I was identifying choices. In abstract terms, these are assignments of values to variables. I took easy examples in these definitions, because it is straightforward to think about, e.g., the number of participants as a variable that can take different numerical values, positive integers. But this is less apparent in the third definition, for the choice of trigger that starts the negotiation timer, where I had the choice "two calendar weeks from the time the asking party requested a proposal from the supplying party".

In simple terms, choices are all items in a definition that you could change. A choice in a definition is something that can or could be different.

A choice is not necessarily easy to identify in the definition. It could be straightforward, like in the cases I had above, or much harder, as in the following example.

- Shippable: A Shippable Unit (a.k.a. "Shippable" only) satisfies all the following conditions:
 - It involves one or more software functionality;
 - It is represented, in the release management process, by exactly one User Story;
 - It is released to UAT Environment;
 - Prior to its release to UAT Environment, it has been tested by Vibe to ensure that this software functionality operates in accordance to its Product Specification;
 - Its Product Specification has been approved, prior to the commitment of development resources to its implementation, i.e., prior to its User Story being introduced for the first time in a Sprint.

This comes from a terminology made by a software engineering team that was adopting for its own work style a general purpose development method. You might recognize that they were adopting and adapting for themselves, a variant of the Scrum method for software development. A first choice, which is hard to see in this definition, is that a shippable should involve pieces of functionality; it was only when the team started dedicating more resources to analysis tasks, which do not produce functionality, but do produce designs of future functionality, that they recognized that outcomes of these analyses cannot be presented as shippables, which skewed data on team's performance, and caused problems when performance assessments started being done by a third party. This was, in fact not a conscious choice, because the team was already - when the definition was made and approved - involved for months in developing functionality which was well specified up front, and there was no analysis effort to do. That option was not considered, and when no options are considered, this part of the Shippable definition seems, at that time, as immutable, it is not a choice at all, even if this turned out to be a mistake.

Another issue is the last item, where the Product Specification needs to be approved up front. This turned out to be a problem, because if we had to follow it rigorously, it would delay a number of decisions to change minor aspects of the product, decisions which we ended up taking anyway, without having the approved specification. However, when the definition was being made and approved, it seemed like the only option.

It can, in short, be hard to see that something in a definition is a choice in the first place. But even in the hard example above, choices were still items, parts of the definition.

It can be harder still, as when the definition itself is a choice, and taking another option would lead you to throw away that definition altogether.

In the Shippables example above, the notion of a shippable is associated, as are notions of User Stories, Sprints, and so on, with the Scrum method for software development. The choice of going along with that method, of aligning to it, is in fact a choice like any other. If we wanted a different method, there may not be a notion of Shippable there at all.

This highlights the fact that you need to worry about choices that led to the introduction of the term to the innovation language, then about choices captured in the definition itself, and the choices that are implicit, which you only learn through experience.

Troubles do not end there. I have a rather different understanding of options and choices when I am analyzing definitions that I create, as opposed to those that others made. Indeed, a term's definition is simply an outcome of the thinking and communication that led to it (including one's specific experience and expertise accumulated up to that point). If you did not participate in, or do that thinking, then your analysis of choices in a definition will be only on the definition itself, and the rest of the innovation language.

But even if I had to make a definition of Shippable, I can still know little about Scrum, or software development altogether. Your analysis, in other words, inevitably leans on the knowledge you have.

And even if I knew a lot about software engineering methods, it would be more appropriate to know specifics of that team, who will be using the definition, than if my knowledge was generic.

The bottom line is that the less you know and the more indirect that knowledge is, the more difficult it is to anticipate the effects of choices in the definitions you make, or are handed over.

I mentioned three parameters so far, which play into your ability to predict effects of choices in a definition. They are:

- If your knowledge is direct or indirect; it is direct if you have expertise and experience on phenomena at hand; it is not if they are new to you, and much of what you can think of these phenomena is, thus, speculation based on, for example, analogy;
- How extensive your knowledge on the phenomenon is. The less extensive it is, the more likely it is that many choices in the definition will remain implicit to you;
- If you are an author of the definition, or only its recipient, in which case you did not participate in whichever explicit choices were made when that definition was designed.

These three dimensions are orthogonal. Your level on one has no relationship to your level on another. With this in mind, your worst case is being a recipient of a definition, while having low indirect knowledge of the phenomenon that the definition is about.

But what if choices in one definition are linked to choices in others? That requires a different view of an innovation language, presented in the next Chapter.

13 Language as Network

Chapter summary: On relationships between definitions in an innovation language, how they lead to a network representation of the language, and why that is useful.

This Chapter elaborates a simple idea, of relating a term A to a term B if the definition of A mentions B, and looking at what to do with these relationships.

Why do it? Because any term in an innovation language is not isolated. It is one among others in a language, it co-exists with other terms there. Said differently, ideas in a language are intertwined, and so are the notions to convey with the innovation language.

Consider the following simple innovation language, made in a project to create software that should help collect data about how products are sold at points of sale. The data is subsequently used to help decision making in marketing and sales at a large electronics manufacturer.

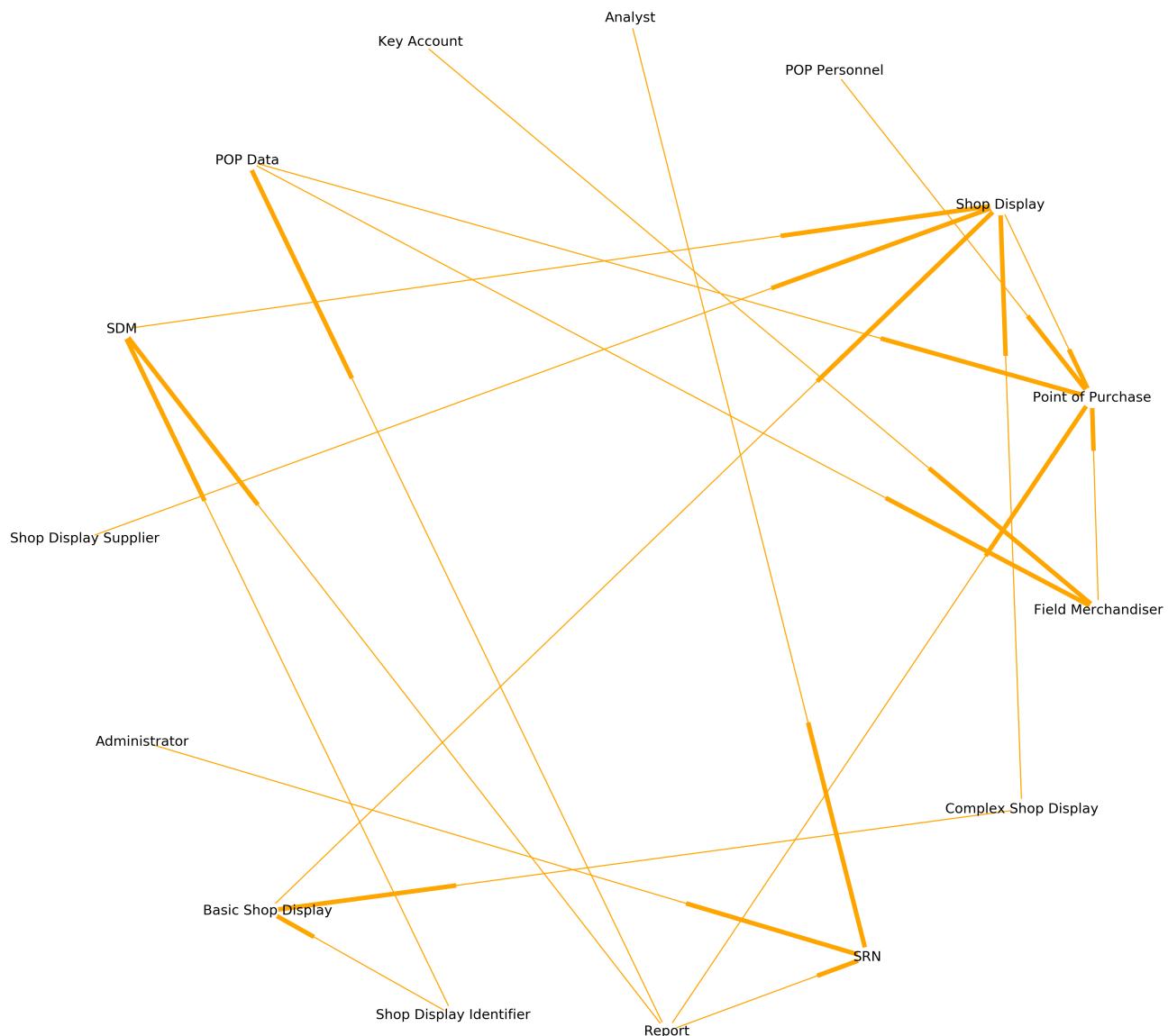
- SRN: Smart Retail Network software.
- SDM: Shop Display Management software.
- Shop Display: object installed at a POP, which is used to promote products; Shop Display is synonym with Fixture.
- Basic Shop Display (BSD): a Shop Display which cannot itself be subdivided.
- Complex Shop Display (CSD): a Shop Display made from two or more Basic Shop Displays.
- Shop Display Identifier (SDID): unique numeric identifier generated by SDM for each BSD, used to identify a BSD when a QR code scanner is not available.
- Shop Display Supplier: representative of the company supplying Shop Displays.
- Point of Purchase (POP): location (shop) where products are being sold.
- Field Merchandiser (FMR): an individual visiting a POP and collecting data about the POP; an FMR typically gets a list of POP to visit, and upon visiting collects data about products being sold at POP, the characteristics of a POP, and any other data of use.
- POP personnel (POPP): an individual employed at a POP and involved in the selling of, and/or presentation of Samsung products to customers.
- Key Account (KA): an individual supervising one or more FMR.
- POP data: data (text, measurements, images) collected at a POP by an FMR.
- Report: a table, chart, or Excel or PowerPoint presentation file generated by SRN or SDM to facilitate the understanding of POP data.
- Analyst: employee interested in analyzing the data stored in SRN.
- Administrator: employee interested in changing the settings and/or performing maintenance of SRN.

Here is a summary of how definitions depend on other terms in the innovation language. Writing A <- B says that the definition of A mentions B, and so we say that A depends on B.

- Analyst --depends on-> SRN
- Administrator --depends on-> SRN
- Basic Shop Display --depends on-> Shop Display
- Complex Shop Display --depends on-> Shop Display
- Complex Shop Display --depends on-> Basic Shop Display
- Field Merchandiser --depends on-> Point of Purchase
- Field Merchandiser --depends on-> Field Merchandiser
- Key Account --depends on-> Field Merchandiser
- POP Personnel --depends on-> Point of Purchase

- POP Data --depends on-> Point of Purchase
- POP Data --depends on-> Field Merchandiser
- Report --depends on-> SRN
- Report --depends on-> SDM
- Report --depends on-> Point of Purchase
- Report --depends on-> POP Data
- SDM --depends on-> Shop Display
- Shop Display --depends on-> Shop Display
- Shop Display --depends on-> Point of Purchase
- Shop Display Identifier --depends on-> SDM
- Shop Display Identifier --depends on-> Basic Shop Display
- Shop Display Supplier --depends on-> Shop Display

The Figure below shows the dependencies as a graph, called the term dependency network. It has 15 nodes and 21 edges.



How does this diagram help? One way to read it, is that it shows dependencies. A term's definition depends on another's. This dependency view of an innovation language gives three types of nodes: inner, stops, and outer.

Outer nodes depend on no term in the innovation language. The definition of an outer node depends entirely on terms defined in other languages, be they natural, technical, or other. That makes them outer, they are at the border with other languages, where the innovation language ends or begins. This is also a node which opens the innovation language most open to interpretation as it make it rely on definitions controlled outside the language, those definitions which are outside its scope.

Stop nodes are on paths between outer and inner nodes. They both depend on other nodes and are mentioned in definitions of other nodes.

Inner nodes depend on other nodes, but are not used to define terms.

The term dependency network can be large, and its visualization impractical. We can take an existing technical language for illustration. [\[UNESCO Glossary of World Heritage Terms\]](#) has 157 terms. Here is a preamble to that terminology:

"This Glossary of World Heritage Terms has been prepared in accordance with the request of the World Heritage Committee at its nineteenth session in December 1995. The Committee requested that the Glossary be a separate document, independent from, but referring to, the Operational Guidelines for the Implementation of the World Heritage Convention and to reports of relevant expert meetings. [...] The Glossary of World Heritage Terms has been prepared primarily by extracting terms and their definitions and interpretations from the World Heritage Convention, the Operational Guidelines and to reports of relevant expert meetings."

Here is a sample definition, for the term "Cultural heritage":

"Cultural heritage is defined in Article 1 of the Convention as shown below:

Article 1 For the purpose of this Convention, the following shall be considered as "cultural heritage":

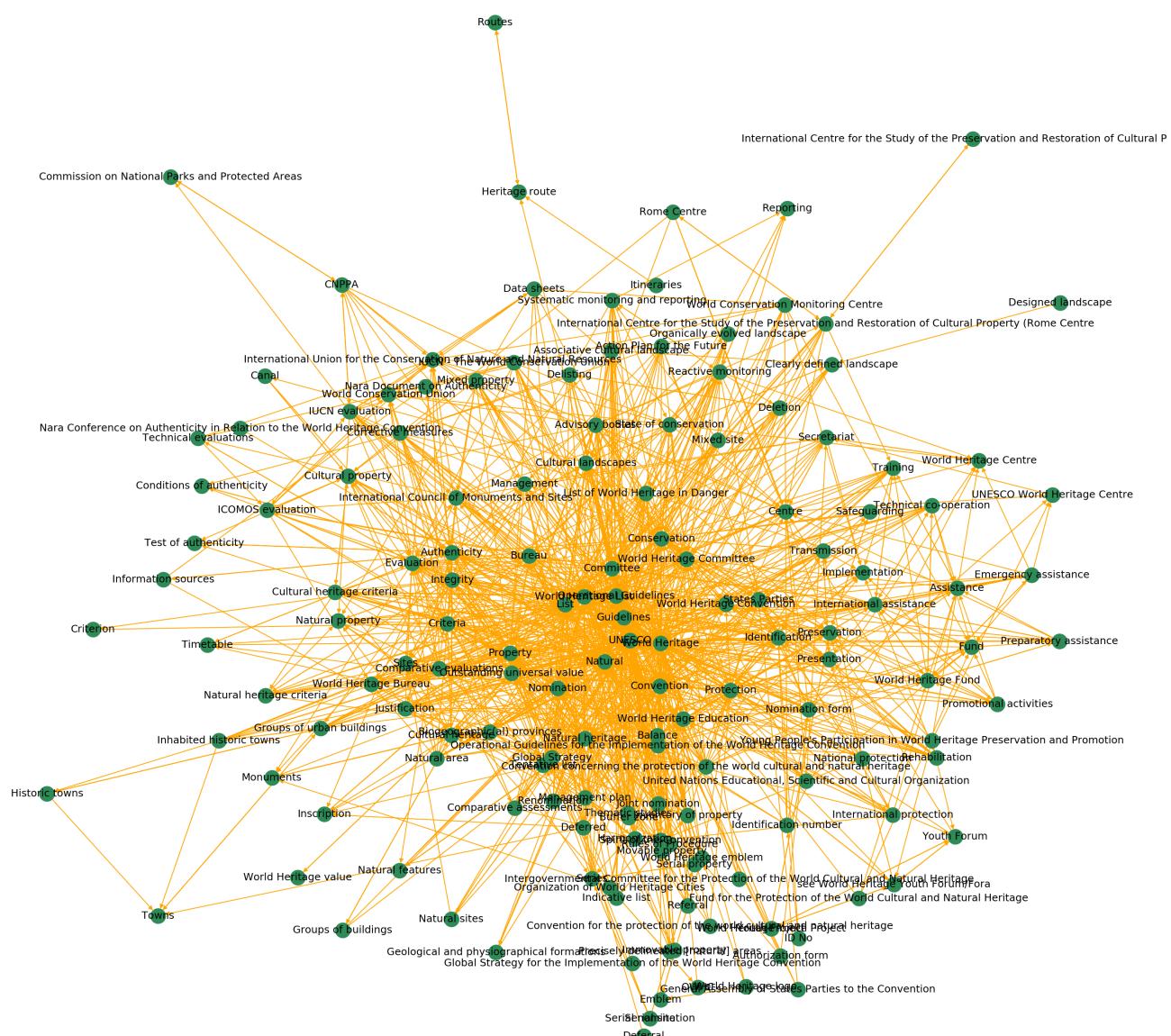
- monuments: architectural works, works of monumental sculpture and painting, elements or structures of an archaeological nature, inscriptions, cave dwellings and combinations of features, which are of outstanding universal value from the point of view of history, art or science;
- groups of buildings: groups of separate or connected buildings which, because of their architecture, their homogeneity or their place in the landscape, are of outstanding universal value from the point of view of history, art or science;
- sites: works of man or the combined works of nature and man, and areas including archaeological sites which are of outstanding universal value from the historical, aesthetic, ethnological or anthropological point of view (UNESCO 1972)."

Doing the same analysis as above, the term dependency network has 157 nodes, one for each term, and 1423 edges, for each dependency between terms. Here is a small sample of the network:

- Action Plan for the Future --depends on-> Action Plan for the Future
- Corrective measures --depends on-> Action Plan for the Future
- Advisory bodies --depends on-> Advisory bodies
- Reactive monitoring --depends on-> Advisory bodies
- State of conservation --depends on-> Advisory bodies

- Assistance --depends on-> Assistance
 - Balance --depends on-> Assistance
 - Convention concerning the protection of the world cultural and natural heritage --depends on-> Assistance
 - Emergency assistance --depends on-> Assistance
 - International assistance --depends on-> Assistance
 - International protection --depends on-> Assistance
 - List of World Heritage in Danger --depends on-> Assistance
 - Operational Guidelines for the Implementation of the World Heritage Convention --depends on-> Assistance
 - ...and many more edges.

The following Figure shows one possible visualization of the term dependency network, for the Glossary of World Heritage Terms.



With large dependency networks, only interactive visualization can be of any use, and tools such as Gephi and Cytoscape are relevant. See:

- [Bastian, Mathieu, Sébastien Heymann, and Mathieu Jacomy. "Gephi: an open source software for exploring and manipulating networks." *Icwsm 8.2009* (2009): 361-362.], and
- [Shannon, Paul, et al. "Cytoscape: a software environment for integrated models of biomolecular interaction networks." *Genome research* 13.11 (2003): 2498-2504.]

14 Evaluating a Term's Importance

Are all terms in a language equally important? Some are more frequently used than others. Use, here, can be measured by the number of times a term is mentioned in definitions of other terms in the language. An alternative is to look at the number of times a term gets mentioned in artifacts that use the innovation language, such as technical specifications, recorded communication between team members, and so on.

Here, we will assume that a term's importance is directly proportional to the number of times it is mentioned in definitions of other terms in the innovation language. The rationale is that, if a term is frequently mentioned in definitions, then your understanding of those definitions hinges on your understanding of that term. If you misunderstood it, that may transfer to your understanding of definitions which use it.

The following table shows the number of incoming edges to a node, in the term dependency network of the Glossary of World Heritage Terms. Given the number of terms (nodes), only those which are called in 30 or more other terms are shown. Each term in the table, is mentioned in the definitions of at least 30 other terms in the technical language.

Following our convention on what a term's importance means, this becomes a table of the most important terms, those whose understanding influences your understanding of many others in the same language. The cutoff at 30 is arbitrary - the count matters only to the extent that it shows relative importance of a term. The more often a term is mentioned in definitions of other terms, the more important is to be careful about its definition.

Term	Used in
Natural heritage	30
Nomination	32
Conservation	35
Property	40
Committee	45
World Heritage List	46
Operational Guidelines	49
List	60
Guidelines	61
Natural	61
Convention	64

Term	Used in
UNESCO	91
World Heritage	96

Even if all terms need to be carefully defined, the most frequently mentioned ones command special attention. Basic statistical analysis above suggests that the World Heritage term is what the language overall learns most on. This is its definition.

"World Heritage may best be defined with reference to the Preamble to the World Heritage Convention which states that:

... parts of the cultural or natural heritage are of outstanding interest and therefore need to be preserved as part of the world heritage of mankind as a whole (UNESCO 1972: Preamble).

Paragraph 1 of the Operational Guidelines refers to the World Heritage (cultural heritage and natural heritage) as being

... among the priceless and irreplaceable possessions, not only of each nation, but of mankind as a whole. The loss, through deterioration or disappearance, of any of these most prized possessions constitutes an impoverishment of the heritage of all the peoples in the world. Parts of that heritage, because of their exceptional qualities, can be considered to be of outstanding universal value and as such worthy of special protection against the dangers which increasingly threaten them (UNESCO February 1996: 1)."

For analyses of this definition's merits and limitations, see:

- [Francioni, Francesco, and Federico Lenzerini (eds.). *The 1972 World Heritage Convention. A commentary*. Oxford, 2008.],
- [Vecco, Marilena. "A definition of cultural heritage: From the tangible to the intangible." *Journal of Cultural Heritage* 11.3 (2010): 321-324.],
- [Pocock, Douglas. "Some reflections on world heritage." *Area* 29.3 (1997): 260-268.], and
- [Labadi, Sophia. "World Heritage, authenticity and post-authenticity: international and national perspectives." *Heritage and globalisation*. Routledge, 2010. 80-98.].

15 Evaluating a Term's Sensitivity to Change

If a term X depends on many others in an innovation language, their changes will affect how X is understood, even if the definition of X remains syntactically the same.

We can evaluate a term's sensitivity to change of other terms, by looking at the number of terms it depends on.

The following table shows, for a given term, the number of other terms its definition uses. All terms are in the same innovation language.

The fewer terms a term uses, the less sensitive it is to the change of these terms' definitions. The more terms it uses, the more sensitive it is. The table is cut, to show the least and most sensitive terms only. As earlier, cutoff values are arbitrary, the motive here being to save space.

Term	Uses
------	------

Term	Uses
UNESCO	0
Conditions of authenticity	1
Nara Conference on Authenticity in Relation to the World Heritage Convention	1
Nara Document on Authenticity	1
Test of authenticity	1
Designed landscape	1
Criterion	1
Deferral	1
Global Strategy for the Implementation of the World Heritage Convention	1
Itineraries	1
Routes	1
International Centre for the Study of the Preservation and Restoration of Cultural Property	1
World Heritage value	1
UNESCO World Heritage Centre	1
World Heritage Centre	1
...	...
Reactive monitoring	20
World Heritage Committee	20
Tentative list	20
Criteria	21
Advisory bodies	22
Balance	24
International assistance	26
Evaluation	26
Authenticity	27
Conservation	27
Integrity	31

Low or high sensitivity are not inherently good or bad. They are signals that trigger questions.

If a term has low sensitivity, it sits at the boundary of the innovation language. It uses little from that language. A boundary term begs questions of scope, of the innovation language, which we look at later. If a term has high

sensitivity, it leans substantially on the rest of the terms of that innovation language.

16 Importance and Change Sensitivity Together

We can focus on a specific term, and look at its importance and change sensitivity. In the Glossary of World Heritage Terms, "Cultural heritage" is used in 18 terms, listed in the table below. The table shows which terms need to be reviewed, when "Cultural heritage" definition changes. The review should ensure that changes to "Cultural heritage" do not produce inconsistency with the definitions of these notions.

Term	Uses
Balance	Cultural heritage
Cultural heritage	Cultural heritage
Convention concerning the protection of the world cultural and natural heritage	Cultural heritage
Authenticity	Cultural heritage
Conservation	Cultural heritage
Criteria	Cultural heritage
Cultural heritage criteria	Cultural heritage
Cultural property	Cultural heritage
Global Strategy	Cultural heritage
Information sources	Cultural heritage
Justification	Cultural heritage
Integrity	Cultural heritage
Spirit of the Convention	Cultural heritage
Groups of buildings	Cultural heritage
Monuments	Cultural heritage
Property	Cultural heritage
Sites	Cultural heritage
World Heritage	Cultural heritage

Sensitivity of "Cultural heritage" is shown with the following table. It lists all eight terms which are used in the definition of "Cultural heritage".

Term	Uses
Cultural heritage	Convention
Cultural heritage	Cultural heritage
Cultural heritage	Groups of buildings

Term	Uses
Cultural heritage	Inscription
Cultural heritage	Monuments
Cultural heritage	Outstanding universal value
Cultural heritage	Sites
Cultural heritage	UNESCO

Notice self-reference in the definition. Both the importance and sensitivity tables include a row where Cultural heritage uses itself, so to speak. This is due to the style of writing in the glossary, where the definition of each term includes, typically at the very start, the term itself.

When looking at importance and sensitivity, the question to ask is if the term can be made more independent from others. The definition of Cultural heritage references the Convention twice, and each time in a different way:

- "...in Article 1 of the Convention as...", by mentioning the term Convention from the same Glossary, and
- "...(UNESCO 1973)..." where it cites the World Heritage Convention publication.

This can be solved by removing one of the two references in the definition itself. To enforce consistent referencing to the Convention, and thus eliminate further superfluous interdependencies, it would be necessary to introduce a rule on how to reference the Convention in all terms.

There are two aims, when trying to make a term less dependent on the innovation language. One is cleanup, the elimination of superfluous mentions, as above.

The second is to check if the term can be rewritten to depend less on others, not because it does not depend on them upon careful reading, but because it depends on them for unclear reasons. For illustration, consider the term "Outstanding universal value" from the same Glossary.

"Outstanding universal value is referred to in several places in the Convention and the Operational Guidelines without specific definition. Parts of the cultural and natural heritage,

'because of their exceptional qualities, can be considered to be of outstanding universal value, and as such worthy of special protection against the dangers which increasingly threaten them'
(UNESCO February 1996: 1, Paragraph 1).

Outstanding universal value is sometimes simply referred to as World Heritage value.

At the 'Expert Meeting on Evaluation of general principles and criteria for nominations of natural World Heritage sites' held at the Parc national de la Vanoise, France on 22 to 24 March 1996,

The expert group recalled that different interpretations have been made of the term outstanding universal value which is a key to the establishment of a selective World Heritage List. In a number of cases the term has been also interpreted as implying best of its kind. The expert group stressed that the notion of outstanding universal value has been constructed over time and may be interpreted as a concept incorporating both uniqueness and representativeness (UNESCO 15 April 1996: 1).

The World Heritage Bureau and Committee will consider the substance of the report of the Expert Meeting at their twentieth sessions in 1996."

Here are its dependencies lists. It is used in 15 and uses 17 terms, if we remove self-reference.

Term	Uses
Action Plan for the Future	Outstanding universal value
Convention concerning the protection of the world cultural and natural heritage	Outstanding universal value
World Heritage Committee	Outstanding universal value
Cultural landscapes	Outstanding universal value
Global Strategy	Outstanding universal value
Tentative list	Outstanding universal value
Canal	Outstanding universal value
Mixed property	Outstanding universal value
World Heritage List	Outstanding universal value
Natural heritage	Outstanding universal value
Spirit of the Convention	Outstanding universal value
Cultural heritage	Outstanding universal value
World Heritage	Outstanding universal value
Thematic studies	Outstanding universal value
World Heritage value	Outstanding universal value

Term	Uses
Outstanding universal value	Bureau
Outstanding universal value	Committee
Outstanding universal value	Convention
Outstanding universal value	Criteria
Outstanding universal value	Evaluation
Outstanding universal value	Guidelines
Outstanding universal value	List
Outstanding universal value	Natural
Outstanding universal value	Natural heritage
Outstanding universal value	Nomination
Outstanding universal value	Protection

Term	Uses
Outstanding universal value	Sites
Outstanding universal value	UNESCO
Outstanding universal value	World Heritage
Outstanding universal value	World Heritage Bureau
Outstanding universal value	World Heritage List
Outstanding universal value	World Heritage value

If, instead, the following definition was used, dependencies would be substantially reduced.

"Outstanding universal value means cultural and/or natural significance which is so exceptional as to transcend national boundaries and to be of common importance for present and future generations of all humanity. As such, the permanent protection of this heritage is of the highest importance to the international community as a whole. The Committee defines the criteria for the inscription of properties on the World Heritage List. [...] To be deemed of outstanding universal value, a property must also meet the conditions of integrity and/or authenticity and must have an adequate protection and management system to ensure its safeguarding." [\[World Heritage Committee. Operational Guidelines for the Implementation of the World Heritage Convention. UNESCO, 2005.\]](#)

This variant was published in "Operational Guidelines for the Implementation of the World Heritage Convention", made after the glossary mentioned here. It has been argued that the newer definition above is also an improvement on former variants. For a discussion, see [\[Jokilehto, Jukka \(2008\) The World Heritage List. What is OUV? Defining the Outstanding Universal Value of Cultural World Heritage Properties. Technical Report. hendrik Bäßler verlag, Berlin, 111p. Monuments & Sites \(2001-\), Vol. XVI. ISBN 978-3-930388-51-6.\]](#)

17 Blowing Definitions Up

An interesting analysis to do with dependencies is to blow up the definition of each term.

For the sake of clarity, let's call "internal terms", all the terms in the innovation language, which have their own definitions. All others will be called "external terms". In the following definitions from the Smart Retail Network case, POP is an internal term, while 'product' and 'location' are external terms.

Point of Purchase (POP): location (shop) where products are being sold.

Term blow up works as follows. Count the number of times each external term appears in definitions of all internal terms. You then look at the most frequent external terms, and decide if they should be added to the language, that is, if they need to have a definition you want to control and agree on in the innovation team, or if there is no need for control and agreement.

We can do this for the Glossary of World Heritage Terms, and show only external terms which appear more than 40 times in all the definitions of all internal terms. We get the following table. The threshold is set at 40 for space reasons, there may be interesting terms that got cut off.

External Term	Count
---------------	-------

External Term	Count
heritage	348
world	278
unesco	193
cultural	144
1996	143
see	126
list	126
natural	122
guidelines	120
operational	114
convention	112
february	110
properties	96
conservation	87
property	85
paragraph	83
committee	73
states	58
assistance	54
protection	51
international	48
article	44
inclusion	43
parties	41

Notice that "heritage" does appear as sub-term in internal terms, World Heritage in particular. If we consider that any sub-term of an Internal Term is also an Internal Term (and this may not always be a good choice), then we can remove all such terms from External Terms, and get the following updated table.

External Term	Count
1996	143
february	110

External Term	Count
paragraph	83
article	44
inclusion	43

When you inspect the Glossary, "february" and "1996" appear frequently, in references to the following citation:

UNESCO February 1996 Operational Guidelines for the Implementation of the World Heritage Convention.
Intergovernmental Committee for the Protection of the World Cultural and Natural Heritage.
WHC/2/Revised. UNESCO, Paris, France.

It is clear what they are used for. "Paragraph" and "article" are also not questionable. They are frequent because of references to parts of documents cited in the Glossary. They are also unambiguous. This is a sample use of "article":

"...**Article** 8(3) of the World Heritage Convention establishes the International Centre for the Study of the Preservation and Restoration of Cultural Property".

Consider "inclusion" now. Here are a few cases where this word appears in definiens:

- Criteria: "...In accordance with Article 11 of the Convention, the World Heritage Committee have established criteria for the **inclusion** of cultural properties and natural properties in the World Heritage List."
- Cultural landscapes: "...included guidelines concerning their **inclusion** in the World Heritage List".
- Deferred: "When the Bureau and Committee examine nominations of properties for **inclusion** in the World Heritage List they classify their decisions into those they choose to inscribe, those they decide not to inscribe, those referred back to States Parties for further information, and those whose consideration is deferred (UNESCO February 1996: 23-25, Paragraph 65)."

And the following table lists all internal terms which depend on "inclusion".

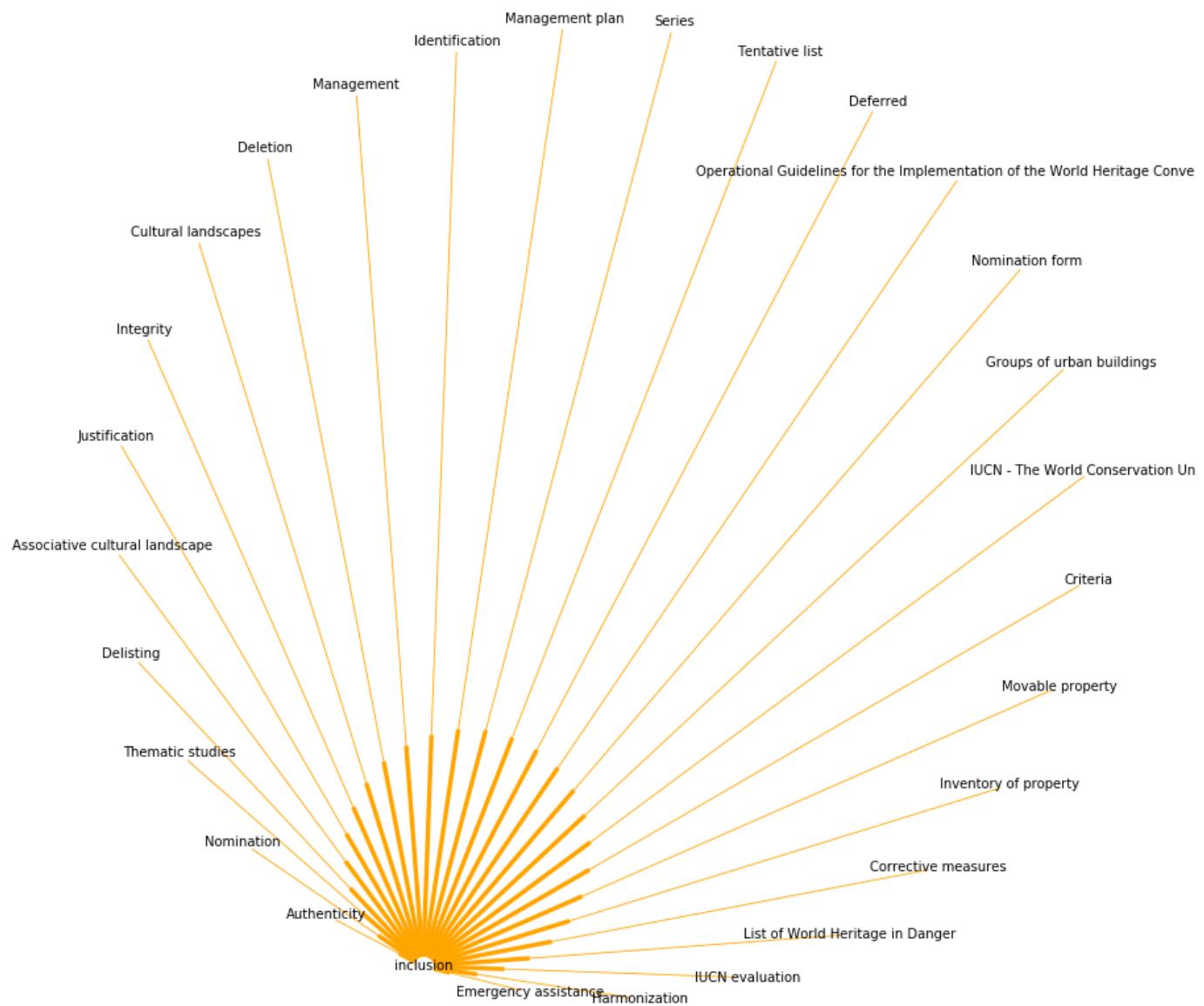
Internal Term	External Term
Associative cultural landscape	inclusion
Authenticity	inclusion
Corrective measures	inclusion
Criteria	inclusion
Cultural landscapes	inclusion
Deferred	inclusion
Deletion	inclusion
Delisting	inclusion
Emergency assistance	inclusion
Groups of urban buildings	inclusion

Internal Term	External Term
Harmonization	inclusion
Identification	inclusion
Integrity	inclusion
Inventory of property	inclusion
IUCN - The World Conservation Union	inclusion
IUCN evaluation	inclusion
Justification	inclusion
List of World Heritage in Danger	inclusion
Management	inclusion
Management plan	inclusion
Movable property	inclusion
Nomination	inclusion
Nomination form	inclusion
Operational Guidelines for the Implementation of the World Heritage Convention	inclusion
Series	inclusion
Tentative list	inclusion
Thematic studies	inclusion

"Inclusion" seems to refer to the adding of a property into the World Heritage List. But, as the excerpt from Deferred shows, inclusion is not a simple act of adding an item to a list. Rather, it is a process, where a property can go through different stages, with different outcomes; it may be nominated for inclusion, under evaluation for inclusion, and if it is so decided, be included in the list.

By doing the above, you identify terms which are often depended on, but not part of the innovation language, and decide if they are candidates to become included as new outer terms.

Visually, this means adding most frequent external terms as new nodes in the graph, then deciding if they should stay as new nodes or not. The following subgraph plots all internal terms which depend on "inclusion".



We can go back to the Smart Retail Network case, for a smaller sized example. The following table shows the frequency of external terms.

External Term	Count
pop	8
shop	7
data	6
display	4
products	3
individual	3
fmr	3

External Term	Count
srn	3
software	2
used	2
displays	2
generated	2
sdm	2
bsd	2
sold	2
visiting	2
and/or	2
presentation	2
employee	2
interested	2

Removing Internal Terms and their sub-terms, we get the following.

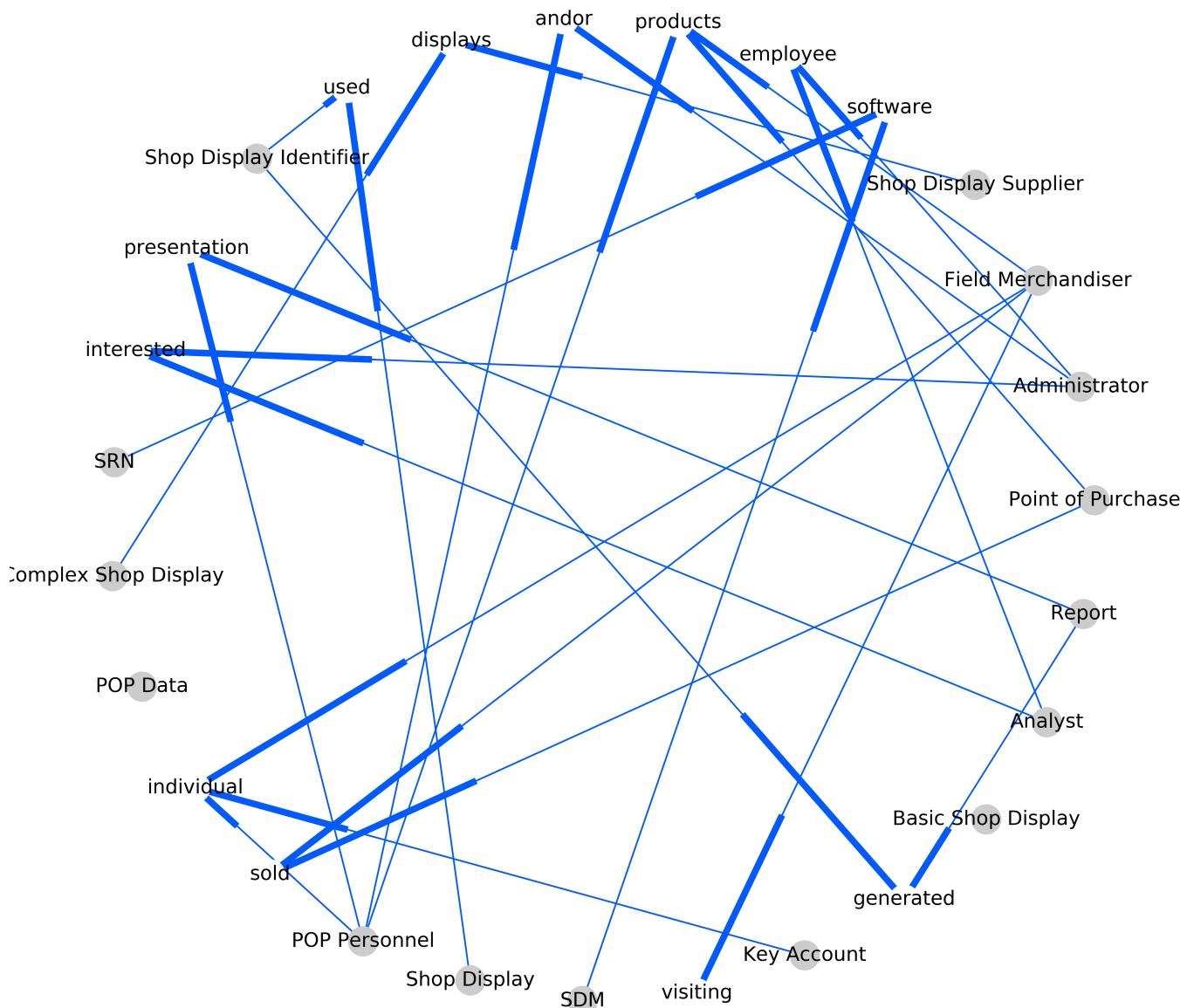
External Term	Count
products	3
individual	3
software	2
used	2
displays	2
generated	2
sold	2
visiting	2
and/or	2
presentation	2
employee	2
interested	2

18 Managing Language Scope

This Chapter is about deciding which external terms to promote into the innovation language.

Term blowup is useful when deciding the scope of the innovation language. The aim is to include important terms in the language, rather than leave them defined outside. Importance depends on two dimensions at least, frequency and potential for misunderstanding. Frequency is the count of times the external term appears in definitions of internal ones, a measure we reviewed in the previous Chapter. Potential for misunderstanding means what it says, that the term has no shared and accepted definition, or if it does, that it is not clear, precise, or accurate enough for your purposes.

With blow up, you widen the scope of the language by including external terms which are frequent and prone to misinterpretation. The following Figure shows two sets of nodes - those in light grey are internal terms of the Smart Retail Network innovation language (Shop Display Identifier, SRN, POP Data, and so on), while those in white are external terms (displays, products, used, individual, among others). Arrows point from internal terms to external terms, so that if an arrow goes from "software" to "SRN", this means that the definiens of SRN includes the term "software".



Should "products" be included inside the innovation language? Should "individual"? How about "visiting"?

There is no general rule for answering these questions. It is not something to decide only by counting occurrences of external terms in definiens of each internal term. Rather, the crux of these questions is risk. If you let every user of the language decide themselves what "products", "individual" and "visiting" each mean, then you run the risk that they disagree. That disagreement is not an issue, if they have no need to decide and act together in ways which, to be successful, require agreement. In such a case, you can accept the risk that this latent disagreement is there, but that its effects will not matter - either will never manifest themselves, or if they do, it makes no difference. But if the manifestations of disagreement can be disruptive, if the effects of disagreement becoming apparent will matter, then this risk needs to be managed. One approach to doing so is to seek explicit agreement on the reading of any one, or all of the said terms. And that means taking an external term, giving it a definiens, and thereby including it in the innovation language.

During the innovation process, in the Smart Retail Network case, we did decide to define "product" and "visiting", and thereby promote them into the innovation language. While their generic, that is dictionary definitions worked early on, this was only because we did not look into the potential consequences of remaining with such definitions; once we did, it became clear these terms had to be promoted into the innovation language.

"Product" is important, because when a SRN user visits a point of purchase, she needs to collect data for a set of products that are assigned to her. Now, that point of purchase does not necessarily carry only those products, that are made by the business which owns SRN. It follows that our definition of "product" is closely related to the scope of the work that a SRN user will be doing. SRN user will not be asked to collect data on all products in a POP, or for products outside a POP. She will only be asked to collect data on some of the products in the POP, and all these products will always be made by the SRN owner. Products, then, were defined as only that which is made and commercialized in any POP by the business which owns SRN.

"Visiting" depends on one's understanding of a visit, and a visit is critical for the design of the SRN, since all data which goes into the SRN, is collected during visits. OED's generic definition of visit, as going "to see and spend time with (someone) socially" cannot work. In the context of SRN, a visit's purpose is data collection, about specific products. Moreover, visits can be of different quality, something that we looked into much later during innovation, when we started looking into average visit durations, depending on POP size, number of products in the POP, and so on. In short, an SRN visit can be, needs to be read in a much more constrained way than a generic definition. Notice how this influences other innovation, and subsequent engineering activities. SRN needs to hold data about visits, so keeping a generic visit definition would lead us to design an SRN which does not hold (or only accidentally does) the data we need to compare, and improve visits over time.

Managing scope also means taking terms out of the language. Here is a simple approach which works in that opposite way.

If you look at the number of internal dependency edges going into an internal term, you can see terms which are weakly connected or disconnected from others. Nodes of most interest now, as candidates for demotion from the innovation language, are those with no incoming links at all. Check why they are internal. Why are they not connected to others? Disconnect suggests that they are conveying ideas which are unrelated to the rest that the innovation language is meant to convey. Is their isolation due to isolation of ideas? If not, are they isolated simply because we have not yet added terms that connect them to the rest of the language?

The following table shows the importance of the internal terms in a version of the SRN innovation language. Shop display is mentioned in definitions of five other terms in the innovation language, five too for Point of Purchase, and so on. Report, Analyst, Administrator, and others, are not used in other definitions in that innovation language.

Internal Term	Used in	Uses
Shop Display	5	2
Point of Purchase	5	0
SRN	3	0
Field Merchandiser	3	2
SDM	2	1
Basic Shop Display	2	1
POP Data	1	2
Report	0	4
Analyst	0	1
Administrator	0	1
Shop Display Identifier	0	2
Complex Shop Display	0	2
Shop Display Supplier	0	1
POP Personnel	0	1
Key Account	0	1

But we do need Report, for example, in the language because demoting it would mean reverting to a generic definition, or more loosely, to everyone's own definition of a report. Another count to look at, is change sensitivity discussed earlier, or the number of internal terms that a given term uses. This is shown in the third column, in the table above.

We see that, while Report is not used in other definitions, it uses several other internal terms, and it follows that Report is a term that is already specific to the innovation language.

Administrator, Analyst, and others who have low counts in the table on both Used in and Uses, need to be looked into more closely. These, at least in the SRN case, are underdeveloped terms. They require more work, and so will become more connected as innovation progresses. This is because it was decided that SRN should offer different functionality to Administrators, Analysts, Key Accounts, and Field Merchandisers. As that functionality gets designed, these roles get assigned rights to access it, and thus details get added to their definitions.

19 Distributed Definitions

If a definition is a pair made of a definiendum and definiens, then consider this simple question: Does the definiens always amount to a few sentences, perhaps paragraphs? Can we always say enough about the definiendum, in a self-contained, neatly packaged definiens, which looks like an encyclopedia entry? The answer is indeed the negative, but what are the implications of that negative? How do you decide where the definiens starts and stops, and what is in, and what lays outside of it?

What I call a *distributed definition* in this chapter, is a definition where the definiendum is in pieces, and these are across varius artefacts. Parts of the definiendum are in one place, parts are elsewhere. The definition is not a tidy whole. There is no way, in a distributed definition, to synthesize the definiens into a few lines, as we want to do in glossaries for example. Instead, if you want the definiens, you need to piece it together across various locations, so to speak.

So far, a term dependency network was made from term and definition pairs. This corresponds to how definitions usually look like, in glossaries, encyclopedia, technical literature, among others.

Can a definition hold all there is to say about the definiendum, especially if that adds to the definiens? There are two cases to think about. Either the definition is all there is to say about the definiendum, or there is a definition and there is more information, which is outside the definition, but which tells us more about the definiendum.

Definitions of abbreviations are an example from the first case. UN stands for United Nations, USA for United States of America, and so on. It is not that the abbreviation describes or tells you much more about what UN or USA are. This applies, more generally, when the definiendum can be replaced, without any loss, by the definiens. But it is likely to work well only when the definition establishes equivalence of two strings of symbols, that is, when the ambition of the definition is to equate at the level of syntax. We could even argue that those equivalences are not definitions, at least not those we talked about so far.

The second case is one where the definiendum is described by the definiens.

A digression is needed. There are, generally speaking, two ways to define, and they can be combined. One is called *extensional definition*, and you use it when you define a term by giving its examples, any number of them. To be precise, you are giving the instances of the class or category that this term designates. This means that if I want to define a tree for you, I should show you trees, give you examples of kinds of trees, and so on. The other kind is the *intensional definition*, where I am defining a term by listing the necessary and sufficient properties or attributes each of its instances should have. Typically, we would say an instance of class, not of a term, but this is a detail that makes no difference here. This is what the Oxford English Dictionary does, when it defines a tree as follows.

Tree: A woody perennial plant, typically having a single stem or trunk growing to a considerable height and bearing lateral branches at some distance from the ground. [OED: Tree]

What attributes or properties does this mention? That a tree is a woody plant, that it has a single stem or trunk, that it has lateral branches, and so on.

Back to the discussion. In an extensional definition, unless the definition enumerates all instances there are, there are going to be instances which are left out. The more instances there can be, the less likely for all of them to be enumerated. As we said before, the extensional definition is going to give a sample of instances, not list them all.

If the intensional approach fails to identify all properties, if some properties are attributed to the definiendum outside the definition, then we are in that second case again.

But this problem of distributed definitions is not only about practicality - that we do not want to list all instances, or all properties. There is more to it. A term's use defines it, and that thought has quite some ramifications.

What if we have a glossary that has a short definition of "internal combustion engine", and then we have a detailed description of how internal combustion engines work in general? Is that description of how they work part of the definition? Where do we fit the information from these descriptions, if information adds to or changes

the definition? Are they part of the innovation language? Are they something to put in a term dependency network?

These are questions about how an innovation language is related to descriptions, explanations, predictions, and prescriptions which use its terms. Is a description of some aspect of the innovation, a definition of that innovation, a part of that definition, or something else altogether, detached from the language?

Let's rephrase these questions. Is the specification of an algorithm for search, part of the definition of the term "search result"? Is the law that uses the term "terrorism" and its definition part of the definition of the term "terrorism"? Can we, in other words, ignore how the term "terrorism" is used in law on terrorism, when we try to define the term "terrorism" in that law? Does it matter to know the details of a search algorithm, to know what a "search result" is?

There is no simple answer.

Take giraffes for illustration. The word "giraffe" is an English word; its plural is too. But is the definition of "giraffe" also part of English, the natural language?

One way to look at this, is to ask, first, if the definiens of giraffe only includes English words, and can be written using English grammar. If not, then clearly the definiens of giraffe cannot be part of English.

But say that you have a definition of giraffe, which does follow English grammar, and uses only English words (it does not matter here which English you are thinking about). Is that definition part of English language, or is English used to write the definition, but the definition is not part of the language?

This seems an easy answer - we certainly use language to make the definition, or better, we communicate the definition using a language. That perspective works, if we reduce language to syntax only.

The problem is, of course, if there you have a definition of giraffe, and then someone else says, still in English, something about giraffes that your definition ignores, yet, seems, or is well justified. For example, it says something about there being a giraffe called April in New York City. Regardless of your original definition being intensional or extensional, this information, even if it is only new to you, gives an additional instance of giraffe. Someone else might tell you that, reticulated giraffes are the most frequently encountered giraffe in zoos worldwide, something you were not aware of. You might learn another fun fact, that female giraffes give birth standing up.

The point is not that you should have known better, before giving an incomplete definition. You could, of course, add the new information to the definition. But that looks like a question of appearance, not substance - if you prefer to accumulate and arrange information in the same place, so to speak, akin to a Wikipedia page, or a manuscript on giraffes. The important point is, rather, that statements about the definiendum, which describe it, can be all over the place, and those making the definition may only be aware of some, and ignore, either willingly or accidentally all others. Needless to say that learning means new information will keep coming in, adding to, or changing what was thought to be appropriate descriptions. So statements about a definiendum are all over the place *and* across time.

Two problems are intertwined so far in this chapter. The last one is about the place and time of pieces of definiens - when making a definition, some of these pieces we have, some we don't, and this could either be because they have not been discovered, or made yet, or because it is us who are not aware of them, while someone else is. A definiens is thus a bunch of pieces, some in, some elsewhere. Definiens, thus, is not only

distributed across space, but also time, provided that knowledge of the definiendum changes over time, that learning happens.

The earlier problem was whether a definiens is part of language. And the temporary solution, while I went on about the second problem, was that definiens is not, but that we use language to make it.

Let's stick with that for the moment, and consider two cases. In one, definiens includes only words which already are part of the language. For instance, this is the OED definiens for giraffe:

"A large African mammal with a very long neck and forelegs, having a coat patterned with brown patches separated by lighter lines. It is the tallest living animal." [\[OED: giraffe\]](#)

To simplify, if OED was the universal reference list of all English words, then this definiens is made of English words alone. OED adds the following note, on the origin of the word:

"Late 16th century: from French girafe, Italian giraffa, or Spanish and Portuguese girafa, based on Arabic zarāfa. The animal was known in Europe in the medieval period, and isolated instances of names for it based on the Arabic are recorded in Middle English, when it was commonly called the camelopard."

[\[OED: giraffe\]](#)

Is this now part of the definition of giraffe? That depends on whether you need it to be - do people who should use the definition know anything about, and agree on the origin of the word, for the decisions and actions they need to make? Let's say that they do, and so make word origin a part of the definiens above it.

The definition of giraffe is no longer in English only. This translates into innovation languages in a simple way, that if we need additional words, we add them - be they from another used or unused language (in which case we either use or promote them into the innovation language), or we invent new words. Instead of using one language, we use more, or we make them up. Needs for agreement, clarity, precision, and accuracy drive the process, rather than care for conventional boundaries between natural, technical, and other languages.

But as we do that, as we add new terms to the innovation language, what if the meaning we want for these terms, that which the definiens should convey, is new too?

Suppose, to continue with giraffes, that I give you the following definition.

A giraffe is a camelopard.

Today, we use none of the languages which fall under the name Middle English. Based on that definition alone, could you recognize a giraffe when you saw one? Yes, because regardless of the present discussion, you were taught better definitions, and had better descriptions of, and perhaps saw live giraffes.

But let's say that we need to agree on the use of the term Alskdjfhg, and that I give you the following definition.

Alskdjfhg is a Qpwoeiruty.

Since these two, as far as I know, do not exist, you have nothing to anchor them to, and so meaning is absent. That absence, importantly, means that if I ask you to go and get a Alskdjfhg, all you know is that you need to get a Qpwoeiruty, and that is equally useless.

The takeaway is that meaning, whatever exactly it may be, cannot be disassociated from the innovation language - it is not only terms and words used in definiens that make it up, it is the whole package, each definiendum and each definiens, at the very least. That was obvious before this chapter; what this chapter does, is that it extends

an innovation language to cover not only the list of definitions, or the glossary, but also artifacts in which the terms in the language are used.

That looks like a slippery slope, going from a list of definitions, to cover much more, and in fact, all uses of the terms defined in the list. But it is not clear where to stop on that slippery slope, or why. If meaning is made by use, as Wittgenstein seems to have argued, then this is what that looks like in the context of innovation languages. And this seems less odd, if we recall that the purpose of an innovation language is clear, precise, and accurate communication about new ideas and things - to accomplish it, we cannot only want to influence, or control the glossary. We have to worry about use as well, whenever and wherever use happens. Either because use will revise previous definitions, extend them, add to them, or because it will go against them in ways which are not desirable.

The fact that you can use terms of the innovation language to define other terms of that innovation language, and the fact that you can use these terms in explanations, descriptions, predictions, and so on, also means that the term's definition is no longer simply that one or few sentences you gave, but also all that you wrote as well, which involves that term. The understanding of that term hinges on understanding all its uses, and while this is elusive for natural language terms, it is not for innovation language terms - the whole point is, with innovation languages, to add new terms, new names for new ideas and things, and to define them. It is just that the definition is no longer simply one or few sentences given for that term, but the whole set of statements that uses this term.

Here is a first example. Recall that a Basic Shop Display was defined as follows.

- Basic Shop Display (BSD): a Shop Display which cannot itself be subdivided.

Shop Display had the following definition.

- Shop Display: object installed at a POP, which is used to promote products; Shop Display is synonym with Fixture.

As the team progressed in the innovation process, they identified the requirement that the software should hold up to date data about BSDs. This made it necessary to decide which data would be held about a BSD. They had the following.

A BSD is defined with the following attributes:

- SDID;
- Unique name;
- Description;
- Type: e.g., Experience zone, Wall display, End cap, etc.; SDM should allow the Analyst users to add/change/remove type of BSD;
- Product category, from a list of Product categories defined by the Administrator;
- Dimensions (Width, Height, Depth);
- If this is a BSD, and if it is part of CSD, then the Unique name of one or more CSDs which this BSD is part of;
- Photo;
- Documentation, which is a set of files (examples include PDF, Word document, but should not be limited to these); it should be possible to upload one or more files, and for each file to input a file description (non-rich text);
- Production date;

- Shop Display Supplier unique identifier; Shop Display Supplier name, address, city, contact email and phone should be shown to Analysts when viewing the data about this BSD;
- Status, being one of the following:
 - Active: currently used at a POP;
 - Out of order: currently at a POP, but not in use (i.e., not visible to customers);
 - Damaged: currently at a POP, but not in use because it has been marked as damaged;
 - Out for delivery: dispatched by Shop Display Supplier for delivery to POP;
 - Ready for delivery: at Shop Display Supplier, but not yet dispatched to a POP;
- POP: unique identifier of the POP at which the BSD is used or should be used, if it is not yet delivered; POP name, address, city, contact email, phone and link to displaying the POP on a Google Map should be shown to Analysts when viewing the data about this BSD;
- Position on POP floor plan; this amounts to an attached file which should show a drawing of the floor plan and the position of the BSD marked on the drawing; SDM should allow only the upload of the file, without visualizing the content of the uploaded file;
- Location, such as Warehouse, POP, etc.; SDM should allow the Analyst users to add/change/remove types of BSD Location;
- Capacity, being the number of products that the BSD can hold.

The question for you is this: how do these attributes of a BSD relate to the earlier definition of the BSD? What should we make of the relationship between its earlier, one-sentence definition, and the list of its attributes?

Consider what these two accounts of BSD are for. The earlier one, the one-sentence definition, is likely to be good enough for any lightweight communication about BSDs, during innovation. If you encountered the term Basic Shop Display, and did not need to know a lot about them, that initial definition would probably be useful.

But if you had to design a database, or part of a broader database, which should hold data about BSDs, then the list of attributes is critical; you would fail if you freely chose attributes, which is what you could in fact do, if you thought that the initial definition was the only definition of BSD. But it isn't, and there is that list of attributes.

Let's take a different perspective. Suppose that you did not have that one-sentence definition of BSD. You only had the list of attributes. Would that count as a definition of BSD? Clearly the list of attributes wold be good enough as an intensional definition of BSD.

In conclusion, we have two definitions of BSD. Both are intensional, and they only overlap on one property, that BSD is set up at a specific POP.

This, then, seems to be an easy case. The second definition adds properties to, and is not in conflict with the first. One is short, the other longer, and both can be kept. You use the short one, until or unless more details are needed.

One thing to be careful about, is to be sure both definitions are about the same extension, about the same set of instances. It is easy to mix the real and the virtual. The first definition could be read both as being about actual BSDs, the shelves or such, which are tangible things, to be found at points of sale. The second definition could also be read as an intensional definition of tangible BSDs. Both definitions could also be read as being about representations of BSDs, that will exist not as tangible objects, but as data recorded in a database.

Is the second definition still an extension of the first? If you carefully read the definition of Shop Display, it says that a Shop Display is an object at a POP, not a representation of that object. Could all properties in the second definition be properties of Shop Displays as objects, not representations? It seems so; even the status can be

written down on the actual shop display, on a sticker placed on that display. But notice that the second definition is written as a requirement for a system that manages data about actual shop displays; it follows that the second definition is not one of shop displays as objects, but of their representations in or by software.

This distinction, of one definition being about objects and the other about their representations, needs to be clear. It follows that terms should not be identical. There is a Basic Shop Display, and its definition is the short one, and there is a Basic Shop Display Representation, or Basic Shop Display Record, which is what that list of attributes defines.

Clarifying this is important. There is no need to confuse actual shop displays with the data which represents them. The data will usually be only about some physical properties of the objects, such as dimensions, but not all physical properties, and there will be properties which are meaningful only through object use and purpose, and are not observable on any particular instance. Examples of the latter, for a Basic Shop Display, are the photo, and the position on the floor of a point of purchase, if that position is the desired one (the actual is observable, and it could differ from the intended one).

Sometime after the initial attributes of BSDs were identified, it seemed useful to have, on each BSD, a machine readable marker - a barcode or QR code. Anyone with the appropriate scanning device, and secure connection to the database of BSD Representations, would be able to scan the marker and retrieve the attributes of the BSD Representation associated with that BSD. Specifically, the following requirement was added.

"When adding (and, as relevant, changing) the BSD data, a QR code should be automatically generated and be downloadable by the relevant Shop Display Supplier as an image, so that it can be printed. The image containing the generated QR code should also - when created - be automatically emailed to the Shop Display Supplier."

How does this change the earlier definition of a BSD Representation? What is clear, is that it mentions a new attribute, the QR code for the instance. Further, it introduces images showing QR codes, says that they need to be automatically made, downloadable, and emailed to specific roles. While the attribute, identifying the QR code, is one more of BSD Representation's attributes, the rest are not attributes. They are properties of another representation, for QR codes associated to BSD representations. But they do provide more details on BSD Representations → it is not the same to know only that a BSD Representation has a QR code, without knowing how a QR code is used, and to know how that QR code is used.

This is what I mean by a *distributed definition*. We have one definition of BSD Representation, the earlier list of attributes. We then added one more attribute via a requirement for the software that manages BSD Representations. Furthermore, that requirement described what the QR code is to be made, and what it should be used for. Overall, we have more about BSD Representations, and the practical question is how to update the original BSD Representation's definition to accommodate the new information.

One approach is to try to rewrite the original definition, and include all new information in it. This is clearly impractical, because there is no guarantee that we know, by this point, all there is to know and that is relevant for BSD Representations. Instead, we can have more changes, attributes removed, new ones added, and more information about what is to be done with BSD Representations. Consider the following description of how a BSD Representation should be added.

Steps:

- The Shop Display Supplier can access SDM only through the Web interface of SRN; the Shop Display Supplier logs into the SRN;

- The Shop Display Supplier clicks on the link Add new BSD;
- SDM shows the list of BSD templates; Shop Display Supplier can choose one or none of the BSD templates;
- SDM shows the input fields (filled out and editable in case the Shop Display Supplier chose a BSD template) needed to record the following attributes (unless specified otherwise, the fields are mandatory):
 - Unique name;
 - Description (free non-rich text);
 - Dimensions (Width, Height, Depth);
 - Documentation [optional field];
 - Status;
 - Location;
 - Product category;
- SDM generates and displays:
 - The QR code for the BSD; the Shop Display Supplier can download the QR code in the JPG format;
 - The SDID for the BSD.

With this, we discover that there are templates for BSD Representations, that some data about a BSD Representation must be provided when it is created (that data is mandatory), that BSD Representations can be added by Shop Display Suppliers. None of this is in the original definition, yet tells us more about BSD Representations.

The other approach, is to let a definition be distributed. A term is defined not only with a sentence or a few, but by the artifacts it is mentioned in. Above, these artifacts were specifications of requirements for a system that needed to be built. These artifacts, together with the original definition, act as a single definition of the BSD Representation, a distributed one. It is distributed, in the sense that there is no single sentence or paragraph, that acts as the sole definiendum for the term; it is now a potentially complex network of rules, instructions, descriptions, and so on, which mention that term, and tell us more about it.

If innovation languages were formal languages (which they are not), I would have said that a term is a predicate, and the definition of the predicate is the theory in which it is used.

The practical implication of this, is that you will have a rather poor innovation language if it looks simply like a terminology. It is a richer structure, which is the topic for the next Chapter.

20 Structuring Innovation Languages

This chapter comes out of two ideas discussed so far. One, we saw that it is possible to generate precisely defined structures, such as graphs, in order to highlight specific properties of an innovation language, then saying something about these properties by analyzing the graphs. Two, we saw that an innovation language does not amount only to a set of definitions - if we see it only as that, then we ignore that definitions, or more specifically, definiens of terms in the language, can be distributed.

The aim in this chapter, is to map an innovation language to a richer structure, which lets us analyze more properties of the language than what we did in prior chapters. This is interesting not only because we want to do additional kinds of analysis, i.e., ask more questions and be able to answer them by computation over that richer structure. Having a precise structure, and procedures - algorithms - for its analysis, is in many cases the only way to say something about relationships across terms in a complex innovation, or technical language; we saw that

even something that looks manageable, like the 150+ term Glossary of World Heritage Terms, involves almost ten times more dependency relationships. A related benefit of a formal structure, is that it tells us how to organize (at a conceptual level) digital storage of an innovation language.

The richer structure in question, is still a directed labeled graph. The difference now, relative to a term dependency network, is that there will be more relationship and node types, and from there, new things to say about the language by computing and looking at the graph's properties.

The chapter works as follows. The first section maps a set of definitions to a simple directed labeled graph. Each subsequent section adds new relationships and node types, shows how to map the content of the definition set to labeled edges and nodes, and which questions this helps us answer, and how about an innovation language.

20.1 Definition Networks

Suppose that you have a set of definitions. Each definition includes two parts, definiendum and definiens. This is the simplest structure. We will later look at how to deal with other items that can be found in glossaries, such as abbreviations of a definiendum, notes meant to complement the definiens, pointers to references outside the glossary, and so on.

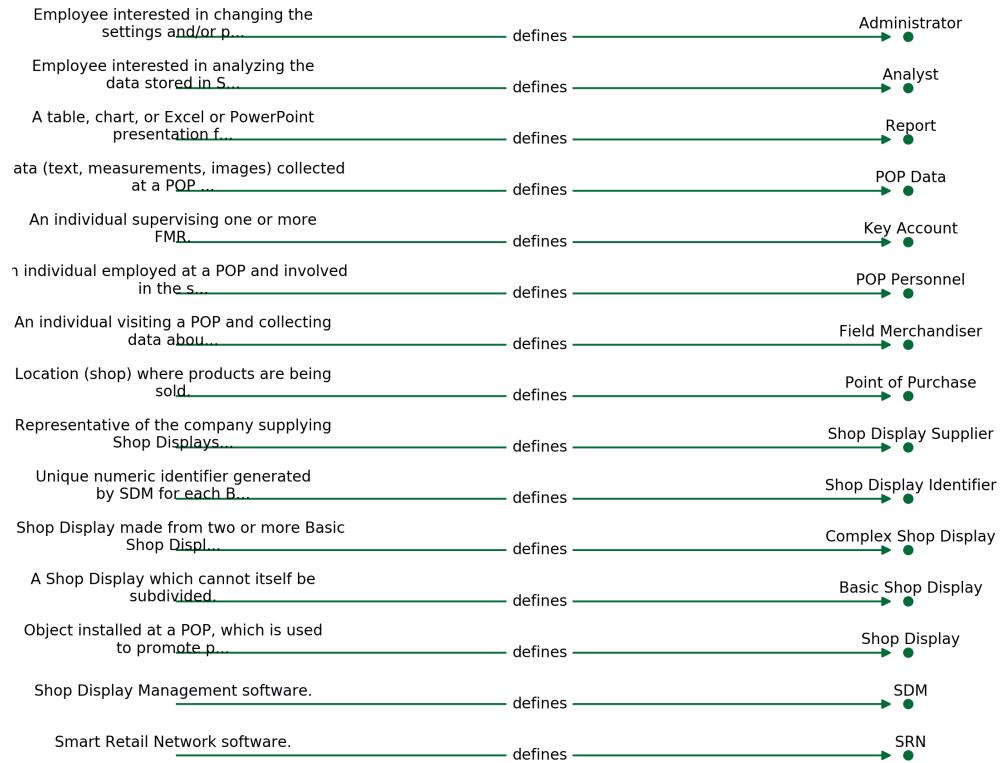
A Definition Network is used to answer the following questions, and do so without anyone having to visually inspect the underlying data:

- Which terms are defined in the data?
- For each term that is defined, what defines it?

To answer these questions, we need the definition network to be a directed graph, where:

- For labels, we have:
 - **Definienda** is the set of labels, each element is a definiendum;
 - **Definientia** is the set of definiens labels, each element is a definiens;
 - The Define relationship relates a definiens to its definiendum; for it, we need a label **Defines**, to be put on edges that instantiate the definition relationship, from a definiens to the corresponding definiendum;
- There is a set **Nodes** of nodes which will carry labels;
- There is a set of **Edges**, of edges to carry labels.

The following figure shows the Definition Network for the Smart Retail Network case.



In the Figure, there is no Definiendum which is defined by more than one Definiens, and vice-versa. This corresponds to what we would expect, that there is, for a given Definiendum, exactly one Definiens. But this is not always the case, which can be confusing.

In the Glossary of World Heritage Terms, the structure of the Glossary is such that there are cases of a Definiens defining more than one Definiendum. This can be identified by looking at the numbers of incoming and outgoing edges in the Definition Network. In the said Glossary, one node has more than one incoming edge, namely "International Council of Monuments and Sites".

Definiendum	Defined by
International Council of Monuments and Sites	2

The following nodes have more than one outgoing edge.

Definiens	Defines
-----------	---------

Definiens	Defines
see Authenticity	4
see Convention concerning the protection of the world cultural and natural heritage	3
see Groups of urban buildings, Inhabited historic towns	2
see Heritage route	2
see Operational Guidelines for the Implementation of the World Heritage Convention	2
see Secretariat	2
see World Heritage Committee	2
see World Heritage Fund	2
see World Heritage Youth Forum/Fora, Young People's Participation in World Heritage Preservation and Promotion	2
see World Heritage emblem	2

The tables suggest something odd in the Glossary, and leads to inspect all nodes above.

The node named "International Council of Monuments and Sites" is a Definiendum, and it has two incoming edges, from a node named "ICOMOS", and another named "Article 8 (3) of the Convention requires that a representative of ICOMOS attend meetings of the Committee...". The second is in fact the Definiens, while ICOMOS is an abbreviation. The incoming edges are a consequence of how the Glossary is structured, where both a term and its abbreviation are treated as two Definienda, and the equivalence between them is conveyed by having two definitions, one for "International Council of Monuments and Sites", and another for "ICOMOS".

What is defined by "see Authenticity"?

Definiens	Defines
see Authenticity	Conditions of authenticity
see Authenticity	Nara Conference on Authenticity in Relation to the World Heritage Convention
see Authenticity	Nara Document on Authenticity
see Authenticity	Test of authenticity

Table below shows the terms defined by "see Convention concerning the protection of the world cultural...".

Definiens	Defines
see Convention concerning ...	Convention

Definiens	Defines
see Convention concerning ...	Convention for the protection of the world cultural and natural heritage
see Convention concerning ...	World Heritage Convention

According to the above, all four terms defined by "see Authenticity" are defined under or in close relation to Authenticity. Since these four terms lack other definitions in the Glossary, the said relationships can also be seen as suggesting a Glossary that is either incomplete, or where Authenticity has an overloaded meaning.

In a summary, even a structure as simple as a Definition Network, can be used to answer a number of questions:

- Which terms are defined in the data?
- For each term that is defined, what defines it?
- Are there terms (Definienda) which have more than one definition?
- Are there Definientia that define more than one Definiendum?

These questions can easily be answered in small Definition Networks. Small is relative, and just how big it can get, to still allow visual inspection, depends on the capability of the procedures that position nodes, to produce a view of the network, in which incoming and outgoing edges of a node can be counter. This is the case for the Smart Retail Network case. But as soon as a visualization gets unclear, and the network is not small enough, it is more practical to compute the numbers of incoming and outgoing edges on each node, and only look at those where either of the values is not one or less.

The last two questions, as we saw above, can signal the need to make changes to the underlying set of definitions. Changes should avoid incompleteness, as in the case of Authenticity which indirectly defines several other terms. This begs, of course, the question of what an indirect definition may be, and why it makes the innovation language incomplete. To look more closely into that, we need an additional set of tools, for the classification of items in the set of definitions we started with.

20.2 Classification Networks

- A single label **Definiendum**, on the node which represents the Term class, each definiendum being its instance;
- A single label **Definiens**, on the node which represents the Definiens class, each definiens being its instance;
- A single label **Is-a**, for edges which relate an instance node to the class node it instantiates.

If definitions are distributed, then a set of definitions is a poor way to think of a structure of an innovation language. One richer structure was to relate definitions with dependencies, which gave us term dependency networks, and ensuing interpretation of basic graph statistics in the context of innovation languages.

The aim in this chapter is to be more precise about a structure that can be used to describe

In this Chapter, I ascribe a structure to innovation languages. I describe that structure and discuss it. I start from the simplest ones, present their structure, and explain why richer structures are needed, and what they look like. I discuss how these structures are related to established ideas in knowledge representation.

The simplest innovation languages are sets of definitions. Each definition has two parts, the definiendum, which is the term being defined, and the definiens, which is the definition. The purpose of a definition, is to tell its user

how to recognize instances of the defined term. Let, then, L be the innovation language, and make it a set of term and its definition tuples.

$$T = \{(t, d), \text{ such that } t \text{ is a definiendum and } d \text{ is the definiens of } t\}$$

We can represent the structure described so far as a directed labeled graph that has the following nodes, edges and labeling functions:

- Nodes and node labels:
 - For each tuple in T:
 - One node for t, labeled "t";
 - One node for d, labeled "d";
 - One node for the class Definiendum (a.k.a. Term), labeled "Definiendum"; we will use this to say that each t in a tuple is an instance of Definiendum;
 - One node for the class Definiens, labeled "Definiens"; this will be used to say that each d in a tuple is an instance of the class Definiens;
- Edges and edge labels:
 - For each tuple in T, one edge, directed from d to t, and labeled "Defines";
 - For each Definiendum instance node t, an edge from t to the Definiendum class node, labeled "Is a";
 - For each Definiens instance node d, an edge from d to the Definiens class node, labeled "Is a".

In a summary, the structure of a simple innovation language corresponds to a set. Each member is a tuple (t, d), t being the definiendum and d the definiens.

A definition can be a combination of an intensional and an extensional definition, or either of these. Respectively, it may define by pointing to properties an instance should have, or by pointing to instances that the user can observe, or otherwise can experience or has experienced.

So in a (t, d), d can be intensional, extensional, or a combination thereof.

This was good enough, until we needed to analyze the scope of an innovation language. We took individual definitions, blew them up to identify potential candidate terms for inclusion in, or exclusion from the innovation language.

Blowing up meant that we turn each term in each definiens, into a member of the set. If there is a term (t,d) in the set, and d includes terms t1,...,t4, then we add (t1,d1),...,(t4,d4) to the set; and we do the same to terms in d1,...,d4, and so on.

We blow terms up, in order to identify how the definition of each depends on the others. A set which only has (t,d) tuples will not include this information. The set needs to include a new set of tuples, of the form (tX, tY), where tX is the term which appears in the definition of tY.

Let that set be X. By now, it has two partitions, T and S, where T includes the terminology, i.e., all tuples of definiendum and definiens, and D includes dependencies.

$T = \{ (t, d) \text{ such that } t \text{ is a definiendum and } d \text{ is the definiens } \}$

$S = \{ (tX, tY) \text{ such that } tX \text{ is a term that appears in the definiens of } tY, \text{ and } tY \text{ is a term in a tuple in } T \}$

Let T_1 be the set of all terms in T . Then, T_1 and S together are a directed graph over terms, also called a definition network. Each node is labeled by a term, and each tuple from S defines a directed edge between terms.

Any measure that you compute on a definition network tells you only about the structure of that network. It cannot tell you if a term has a good or bad definiens, if it can be improved, how, and so on. Measures I mentioned in earlier chapters, do reflect the content of definitions, but those reflections are crude. This remains an important caveat, when analyzing a terminology by doing computations over its definition network.

Blowing up a definition into terms is different from identifying properties mentioned in it if the definition is intensional, or examples if it is extensional.

Let's say that a definiens assigns at least one property, and/or identifies at least one instance of the term. Let p denote an arbitrary property, and c an arbitrary term instance. With that, for every (t, d) in the terminology, we now have one set for properties mentioned in d and another set of instances mentioned too. Either, but not both of these sets can be empty ♦ what would a definition even be, if both were?

For a given (t, d) , we now have a set P_t of properties, and C_t of references to sample instances. A property is conveyed by a statement (or can be rewritten as a statement), such as ♦color is green♦ or ♦temperature is 95 centigrade♦. A member of C_t is also a statement, or more than one, for example, ♦BestBuy on West 4th Street is an example of a POP♦, as long as that one or many statements are necessary to point to an identifiable instance of the term t .

If you take, then, only one (t, d) , and its P_t and C_t , you still need to add something that captures, in the structure of the language, that each member of P_t is a property of t , and each member of C_t an instance of t . This can be done by adding two sets, one that puts a directed edge from each property in P_t to t , and another which does the same from each instance in C_t to t .

For each (t, d) , we now have two graphs. One graph includes, as nodes, t and all members in P_t , and has edges connecting the latter to the former. The second includes t and all members of C_t , with edges from each of the latter, to t . That is, you have, for each (t, d) a tree rooted in t , with leaves being, respectively, members of P_t and C_t .

In a summary, for every (t, d) in T , such that there are a set P_t of properties conveyed by d , and a set C_t of instances identified by C_t , there are:

- A tree $G(P_t)$, rooted in t , with directed edges from every member of P_t to t ; this is called a property tree of t ;
- A tree $G(C_t)$, rooted in t , with a directed edge from every member of C_t to t ; this is called a sample tree of t .

We add this to X , which gives a more complicated structure for an innovation language, namely

$X = T \cup S \cup T_P \cup T_C$,

Where T_P is the set of property trees, of all terms in T , and T_C is the set of all sample trees of all terms in T .

An alternative way to look at this structure, is to see it all as a single labeled graph.

Let there be a tuple $(N, E, L, \text{Lab}_N, \text{Lab}_E)$, where

- N is a set whose items are called nodes,
- E is a set of items called edges,
- L is a set of items called labels,
- Lab_N is a function that maps a member of N to a member L ,
- Lab_E is a function that maps a member of E to one in L .

Lab_N and Lab_E are called, respectively, the node and edge labeling functions.

The idea is simple, that every node in every graph in X , becomes a node in N , and that we label nodes and edges so as to preserve all information which X has or would have for an innovation language.

To have in G the information in the terminology T , consider what we need for a single (t,d) tuple:

- A node n_1 labeled $\diamond t \diamond$;
- A node n_2 labeled $\diamond \text{term} \diamond$;
- An edge from n_1 to n_2 , labeled $\diamond \text{is a} \diamond$, to say that n_1 is an instance of n_2 ;
- A node n_3 labeled $\diamond d \diamond$;
- A node n_4 labeled $\diamond \text{definiens} \diamond$;
- An edge from n_3 to n_4 labeled $\diamond \text{is a} \diamond$;
- An edge from n_3 to n_1 labeled $\diamond \text{defines} \diamond$, to say that the definiens on n_3 defines the term on n_1 .

This looks as follows.

We can now add dependencies between definitions of terms, to capture the information that was in the set S earlier. Recall that there was an edge in S , directed from t_1 to t_2 if the definiens of t_1 mentioned the term t_2 .

Suppose, for simplicity, that our innovation language has only one term, (t,d) , that d mentions a term t_1 , and that we want to add t_1 in the language. Let d_1 be the definition we made for t_1 . We get the following graph.

The relationship $\diamond \text{depends on} \diamond$ relates d to t_1 , to represent that t_1 is mentioned in d . In contrast to edges in S , dependency is now between a definiens which uses the term, and the term it depends on. In general, every definiens which mentions a term, also depends on that term. As d defines t , t depends on t_1 .

Next, we need to represent properties and sample instances. Let d assign t the properties p_1 and p_2 , and d_1 assign t_1 the properties p_3 and p_4 . Moreover, say that d mentions sample case c_1 , and d_1 mention c_2 . We get the following graph.

With this graph, we represent the same information as in The structure X defined earlier. We do not have the property and sample trees as they were defined earlier, but these can be obtained with simple transformations from the graph above. In the next Figure, the green boundary marks the information needed to make the property tree for t , and the blue boundary to define the sample tree for t_1 .

Let's return to the Smart Retail Network case. The following Figure shows that language as a network of term and definition nodes, with is-a and defines relationships.

The next Figure shows the same innovation language, now with dependencies, in orange, shown between definitions and terms they use.

XVI Epistemology of an Innovation Language On the relationship between knowledge, intentional states and ideas conveyed in definitions in an innovation language.

Is the definition of a term, its definiens, knowledge or assumption? Is it a reflection of something that you are sure about? Does it have to be? These questions are closely related with time. Does the term refer to something that does not exist now? Is it about something that is already there? Or, perhaps about something that should not happen at all? This is not only tied to time; it relates to observation, experience, experiment, and prediction as well.

Why do these questions matter? 

An innovation language develops as a team progresses through an innovation process. Regardless of how their innovation process is structured, or who does it, it will involve people making assumptions, transposing their experiences to new situations, trying, sometimes succeeding to close the gap with expectations, often failing. Some of what they learn along the way can be called knowledge, some of it remain assumptions. But their actions are not driven by assumptions or knowledge alone; they innovate because of motives to change what they know already, and intentions to do so.

How are, then, intentions, assumptions, and knowledge tied to an innovation language? The short story is this: intentions lead to the formation of assumptions about what the problem is, and how to resolve it, assumptions need to be validated through experience, and this happens through trial and error, which in turn lead us to observe evidence about the validity and relevance of assumptions, and from there, some of the conclusions can be called knowledge. As this learning process goes on, its outcomes will determine the scope of the innovation language and its content - the terms and their definitions.

But the innovation language is one of artifacts that will reflect and help the innovation process. It is neither its only artifact, nor the main one. So while it may not be central to learning that happens through innovation, we can keep track what, in definitions, are expectations, assumptions, knowledge. This makes a difference in two ways. One, put bluntly, is that it lets you have a judgment about the innovation language - there may be something to worry about if it is all assumptions and wishful thinking, late in the innovation process. In other words, you may want to have less speculation in definitions, as innovation matures. Two, it is a way to use the innovation language as that on which you track progress in learning.

Doing that means doing more analysis of the innovation language. Now, however, the structure we established in prior Chapters is not enough. The analysis has to be concerned with content, with how statements announced in definitions fit observation and experience, if they fit the present or if they tell us how to tailor the future, and how much evidence, if any, and how we got it, to support our claims that some definitions are built on solid grounds.

Let's again go back to one of the examples used so far, the lightweight case of the smart retail network. The first task is to distinguish content describing the present from that which describes future.  The former are in green, the latter in red.

SRN: Smart Retail Network software. SDM: Shop Display Management software. Shop Display: object installed at a POP, which is used to promote products; Shop Display is synonym with Fixture. Basic Shop Display (BSD): a Shop Display which cannot itself be subdivided. Complex Shop Display (CSD): a Shop Display made from two or more Basic Shop Displays. Shop Display Identifier (SDID): unique numeric identifier generated by SDM for each BSD, used to identify a BSD when a QR code scanner is not available. Shop Display Supplier: representative of

the company supplying Shop Displays. Point of Purchase (POP): location (shop) where products are being sold. Field Merchandiser (FMR): an individual visiting a POP and collecting data about the POP; an FMR typically gets a list of POP to visit, and upon visiting collects data about products being sold at POP, the characteristics of a POP, and any other data of use. POP personnel (POPP): an individual employed at a POP and involved in the selling of, and/or presentation of Samsung products to customers. Key Account (KA): an individual supervising one or more FMR. POP data: data (text, measurements, images) collected at a POP by an FMR. Report: a table, chart, or Excel or PowerPoint presentation file generated by SRN or SDM to facilitate the understanding of POP data. Analyst: employee interested in analyzing the data stored in SRN. Administrator: employee interested in changing the settings and/or performing maintenance of SRN.

There already exist shop displays, basic and complex, there are field merchandisers, staff at points of purchase. There are key accounts, and in general, there is staff interested in using SRN and SDM. But since these systems do not exist now, anything which describes STN and SDM is about a future in which they are expected to exist and run.

You can think of this as two murky pictures, both rough and partial, one of the present, the other of a potential future. Several things are missing, and the innovation process needs to produce them. A more complete picture of the present is needed, alongside a more elaborate one of the future. There needs to be an account of how to transition from the former to the latter.

What to do after you have the distinction between present and future? The decision to make is if you want to improve them, and if so, then when and how. That decision is closely tied to another distinction, between assumptions and knowledge.

You have two options.

One is to stop improving the innovation language. This would be a fair choice if there are no more resources to do so, or if you know no way to do it. Does this also mean that you stop the innovation process? It could; if not, then you are continuing innovation, but disconnecting it from the innovation language.

Two is to continue the innovation process, and to keep the innovation language up to date. The latter cannot happen without the former. The language will get updates, terms and definitions will change, be added and removed.

How do you decide which definitions to work on, and how? If the innovation language is only used as a record, then language changes will not be decided because, or be driven by an analysis of the language. But the language can do more. It can help the innovation process, by pointing to terms that require more attention, and that, as we will see below, can even suggest next steps for innovation.

Having distinguished present and future, the task now is to distinguish assumptions and knowledge. For that, we need working definitions of these two.

Usually, it takes the following three conditions for you to know something [SEP], let's call that X:

- X is true, and is so in the metaphysical sense, so to speak. So X fits actual conditions in the world, and it really does not matter if you observed or otherwise established that X indeed does fit the world. Someone else might have observed the opposite, for example, and tried to convince you X is not true, but that has nothing to do with X being true that, broadly speaking, the metaphysical part of X being true. It may be true that a white eagle landed just now on a rooftop of some house on Whistler, in British Columbia; I haven't seen this, and no one may have seen it really, but this does not preclude it from being true. This

also means that you can say that you know X, but in fact have no way of assuring that X is true in this sense. It means, in addition, that you may say that you know X, but if it is not true, than you are making a false statement.

- You believe X, but not in the lightweight sense of being confident that X is true. You need to believe X in the sense of being committed to it being true; this is not only being certain in its truth. It is also that you would defend it being true from arguments against it, and more generally, your choices and actions would be consistent with your belief that X is true (you would not choose and act as if X was not true).
- You need to have a justification for your belief that X is true. So you cannot choose, for no reason, to believe that X is true. You need reasons, and these reasons need to support your belief that X is true. A simple way to think about this is that you can explain why you believe that X is true; what exactly counts as a good explanation is a separate issue.

The term ♦knowledge♦ is a subject of many discussions, and the above is a common technical definition.

If assumptions are a record of beliefs, then you see how knowledge departs from assumptions. Assumptions need not be justified, and they need not be metaphysically true. An assumption is belief freed from constraints knowledge has.

But there is a problem here, with taking knowledge to be ♦knowledge♦ as defined above. This is because of what we want them for, when analyzing innovation languages.

The analysis to do, once we distinguished content about the present and future, is to identify content which we are least confident in. Such content begs attention, before we can take it seriously when deciding where next in the innovation process.

We should be less confident in assumptions, and more in knowledge. But while that is obvious, the practical side is that there is ♦ in innovation - rarely much content that is as good as we need it to be, to qualify as knowledge.

This is because innovation is about the new, and so we not only have a picture of the present that we need to be confident in, but also in a picture of the future, and in the account of how to transition from one to the other.

The problem is how to measure confidence in the content of an innovation language. The distinction knowledge-assumption is too crude for this. Metaphysical truth is to blame. Something can be true even if we have no way to establish if it is, in a reproducible way. So how, then, can we check if something is indeed metaphysically true? If we cannot, then we♦ll have little knowledge in an innovation language. It will be easy to say, and safer, that all in it are assumptions, which is equally uninteresting ♦ it tells us nothing about which of these assumptions we can take seriously, or what we risk if we do.

If we abandon metaphysical truth as a necessary condition for knowledge, we get a lesser kind, which requires justification and commitment to truth. The latter is really a condition that relates our actions to that lesser knowledge; we display commitment if our actions are consistent with the content of that knowledge.

While we certainly can propose an alternative weaker notion to knowledge, as defined above, there is another approach. It is less ambitious, and works. It consists of avoiding notions such as knowledge and assumption, and instead, of evaluating justifications that are given for the content of an innovation language. ♦Content♦ is a definition, or part of a definition, the inclusion of a term, or its absence, i.e., content that says something about the scope of the language.

In this approach, we take some self-contained content in an innovation language (however we decided to cut it up and make it self-contained), and we ask the following questions:

- Relevance, which is that our decisions in the innovation process would be different without that content?
- Is that content relevant, relative to the purpose of the innovation process?
- Is there an explanation for its relevance?
- If there is an explanation of relevance, is that explanation acceptable relative to all information we have?
- Accuracy, which is the correspondence between phenomena it describes and our experience of those phenomena:
- Is there an explanation, evidence for its accuracy?
- Can we check its accuracy, within available resources? Can we access and observe or otherwise experience the evidence that is available for its accuracy?
- If there is an explanation of accuracy, is that explanation acceptable relative to all information we have?

Ideally, all content in an innovation language would yield positive answers to all relevance and accuracy questions. When that fails to happen, and it usually does, negative answers help you decide what to improve next.

To answer relevance and accuracy questions, we need to align on a notion of explanation and when it is acceptable. We need to know what qualifies as evidence, when it is observable, and how it relates to explanations and their acceptability. Just like we did so far, the aim is not to have generic definitions, but those that work well enough for the purposes of this book. An explanation is a statement or account that makes something clear, a reason or justification given for an action or belief. In turn, a justification is the action of showing something to be right or reasonable. To have an explanation, then, we need statements given with the purpose to take actions and beliefs as reasonable. However we choose to put this, these statements are given to make something, actions or beliefs, accepted by an audience, those who for some reason need to be aware of, and not counter actions and beliefs.

The question of what explanations are, is one of the central concerns in philosophy of science. The common answer goes as follows.

According to the Deductive-Nomological Model, a scientific explanation consists of two major constituents: an explanandum, a sentence describing the phenomenon to be explained and an explanans, the class of those sentences which are adduced to account for the phenomenon. For the explanans to successfully explain the explanandum several conditions must be met. First, the explanandum must be a logical consequence of the explanans and the sentences constituting the explanans must be true. That is, the explanation should take the form of a sound deductive argument in which the explanandum follows as a conclusion from the premises in the explanans. This is the deductive component of the model. Second, the explanans must contain at least one law of nature and this must be an essential premise in the derivation in the sense that the derivation of the explanandum would not be valid if this premise were removed. This is the nomological component of the model - nomological being a philosophical term of art which, suppressing some niceties, means (roughly) lawful. In its most general formulation, the DN model is meant to apply both to the explanation of general regularities or laws such as [why light conforms to the law of refraction and also to the explanation of particular events, conceived as occurring at a particular time and place, such as the bent appearance of the partially submerged oars of a rowboat on a particular occasion of viewing. As an additional illustration of a DN explanation of a particular event, consider a derivation of the position of Mars at some future time from Newton's laws of motion, the Newtonian inverse square law governing gravity, and information about the mass of the sun, the mass of Mars and the present position and velocity of each. In this derivation the various Newtonian laws figure as essential premises and they are used, in conjunction with appropriate information about initial conditions (the masses of Mars and the sun and so on), to derive the explanandum (the future position of Mars) via a deductively valid argument. The DN criteria are thus satisfied.]

Two ideas above are important. The explanation are statements from which should conclude that which we are explaining. And the argument thus built, must include what they call laws of nature, that is, statements accepted as describing regularities which have been confirmed, via rigorous reproduction, as repeatable and true.

Now, in a less demanding context we have here, that chaos of innovation, we can retain two ideas as follows. One is that the explanandum should be the conclusion of an argument we build to support it. Two, that that argument should, as much as possible, rely on statements which have been accepted elsewhere, and do not have to be taken at face value, for the first time, in a given innovation process and within the realm of a specific innovation language. So they need not, unless that is possible, appeal to ♦natural laws♦, for us to have an argument supporting the explanandum.

With this, we drop below the demands of scientific explanations. But it makes little sense to expect established explanations as sole explanations in the context of innovation. The new will talk about an uncertain future, which can hardly be wholly accounted for by deduction from current conditions and natural laws.

But what is, then, the structure of that argument we can build? If we let go of natural laws, how low can we go and still aim for useful accuracy and relevance?

One level down, in some sense, is to let go of natural laws and accept statistical regularity. In other words, to forgo the idea that we need to be certain that the statements in the argument in fact are accounts of regularities without exceptions, and accept that they can be regularities that we have only high probability (whichever exact threshold applies for ♦high♦). Instead of saying that the argument needs to include generalizations that are certain, it can include generalizations which we will see more often than not.

As we slide down the slippery slope, and accept that much of the explaining in innovation will have to forgo appeals to certainty, we are merely acknowledging that innovation involves uncertainty. In my experience, that uncertainty is also hard to quantify. This makes it hard to give relevant estimates of probability for truth or falsity of specific statements which we build into explanations.

And as we slide down far enough, we get to a simple, but effective notion of local acceptability. It is local, in the sense that a statement in the explanandum is never thought of as true or false, but acceptable or not in the face of available information. And the availability of the information is bound by what can be offered by those who participate in the innovation process, i.e., is bound by their capacity to find information which supports or counters the statements in the explanation, thus making ♦ as we will make more precise below ♦ the explanation itself acceptable or not.

What matters, in the end, is that we should not appeal to scientific explanation where we cannot have it, and that is exactly what happens in innovation. A constructive and pragmatic reaction to this drawback is not despair, but a different approach, where we do not look for truth, but acceptability. And acceptability is local, in the sense that something is acceptable only relative to what we confronted it to; there may be something we forgot, which would have made it unacceptable, but too bad if we did ♦ it remains acceptable until we do mention it. This is far less ambitious from metaphysical truth.

Let♦s now give a formal structure to an explanation, by breaking it down and singling out special kinds of relationships between the parts we thus get.

We said that an explanation is first broken down into an explanandum and explanans, and that each is made of statements. Since we want explanans to be the conclusion one draws from explanandum, we need three relationship types between statements in an explanation.

We need to be able to group statements into explanandum or explanans, so that, in the structure we want to show, we can indeed show that an explanans follows from its explanandum.

Here is an example from the smart retail network case. Other than minor corrections for confidentiality, this is a reproduction of a requirement mentioned early on.

POPP would use SRN in order to:

- View/print product information, including:
- Product specifications
- Product images
- View available trainings
- Register for a training
- Make and track product orders
- Consult the Q&A database
- Post questions in the Q&A database
- Post answers in the Q&A database
- Participate in the SRN community by:
- Recommending answers
- Posting open-ended questions
- Answering open-ended question

You can ask different questions, depending on what you want to have explained. Explain how exactly a POPP would be accessing SRN. Explain how a POPP would register for training. And so on. But before any such details are interesting, it makes sense to ask what would draw, incentivize a POPP to access SRN anyway. In other words, please explain why a POPP would use SRN at all.

The explanation was as follows.

For a POPP, SRN provides a single approved source of product information and training content. As such, SRN is a tool for POPP to know products better, and thus sell better.

If this is a benefit for POPP, the owner of SRN needs to invest in order to realize this benefit, and the question then, is why that investment would make sense. This was the explanation.

When POPP participate in SRN, they have access to product-related content that is being generated by others on SRN. At the same time, product manufacturer (which here is the same as SRN owner) product support staff use SRN, and can answer any questions from POPP. This way, SRN community generates content which helps SRN owner understand the concerns and capabilities of POPP across points of purchase. This in turn helps SRN owner marketing to POPP, such as planning and shaping training.

We have two explanations above.

In the first, explanandum can be stated as ♦POPP will use SRN♦, and explanans is the paragraph that starts with ♦For a POPP♦♦.

In the second, explanandum is ♦SRN owner has interest to invest in SRN features targeted at POPP♦, and explanans is the paragraph that starts with ♦When POPP participates♦♦.

Following the conventions we had earlier, this structure corresponds to the following:

- Node n1 labeled ◇POPP will use SRN◇;
- Edge labeled ◇is a◇ from n1 to a node n2;
- Node n2 labeled ◇Explanandum◇;
- Node n3 labeled ◇For a POPP, SRN provides a single approved source of product information and training content. As such, SRN is a tool for POPP to know products better, and thus sell better.◇;
- Edge ◇is a◇ from n3 to n4;
- Node n4 labeled ◇Explanans◇;
- Node n5 labeled ◇Explanation◇;
- Node n6 without a label, to act as the explanation instance, which combines the explanans and explanandum instances on nodes n3 and n1;
- Edge ◇is a◇ from n6 to n5;
- Edge ◇in◇ from n3 to n6;
- Edge ◇in◇ from n1 to n6.

To simplify writing, the above can be stated as follows.

(n1, ◇POPP will use SRN◇) (n2, ◇Explanandum◇) (n1, n2, ◇is a◇) (n3, ◇For a POPP, SRN provides a single approved source of product information and training content. As such, SRN is a tool for POPP to know products better, and thus sell better.◇) (n4, ◇Explanans◇) (n3, n4, ◇is a◇) (n5, ◇Explanation◇) (n6) (n6, n5, ◇is a◇) (n3, n6, ◇in◇) (n1, n6, ◇in◇)

You can do the same for the explanation of ◇SRN owner has interest to invest in SRN features targeted at POPP◇. This gives the same structure, but now over other Explanandum, Explanans, and Explanation instances.

The relationship ◇in◇ can be used whenever we need to look at part of the content of a node's label. Consider the explanans node n3. There are four ideas in it.

1. For a POPP, SRN provides a single approved source of product information and training content.
2. As such, SRN is a tool for POPP to know products better.
3. If POPP know products better, POPP will sell them better.
4. If POPP use SRN to learn more about products, then POPP will sell products better.