From Playing with Constraints...
... to Thinking in Terms of Limits
Towards New Research Directions in CS

Florence Maraninchi
www-verimag.imag.fr/~maraninx

Verimag/Grenoble INP - Ensimag

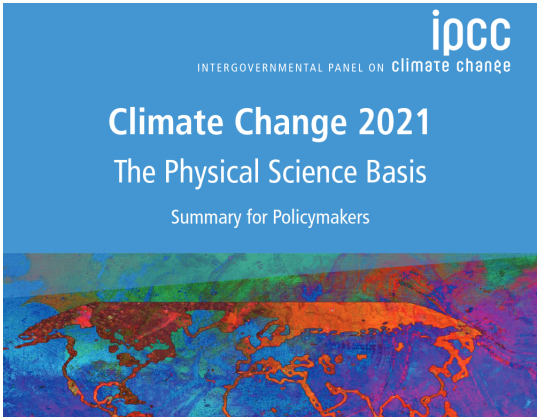
November, 2021 – SYNCHRON – La Rochette

Acknowledgments

CNRS/Ecoinfo, Verimag/ETICS, CITI/Phenix, Campus d'Après Grenoble, Séminaire transdisciplinaire sur l'anthropocène Grenoble, Inria/STEEP and Inria/Spades, Ensimag MEGA, GDR GPL, ...

And a lot of colleagues on Twitter...

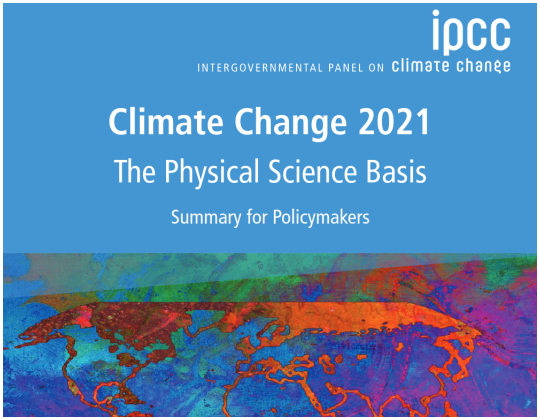
How Should a Researcher (in CS) React?



Possible answers:

[1] *https://cacm.acm.org/magazines/2020/1/241717-publish-and-perish/fulltext*

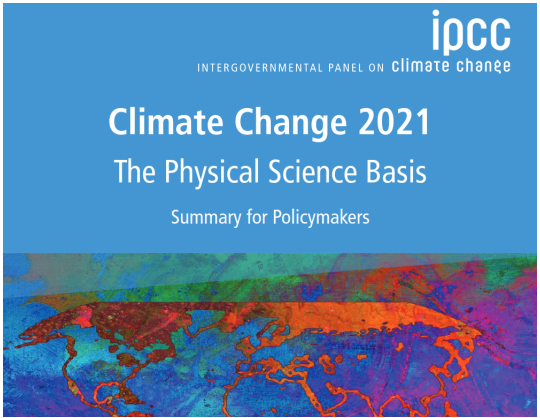
How Should a Researcher (in CS) React?



Possible answers:
— I don't care

[1] *https://cacm.acm.org/magazines/2020/1/241717-publish-and-perish/fulltext*

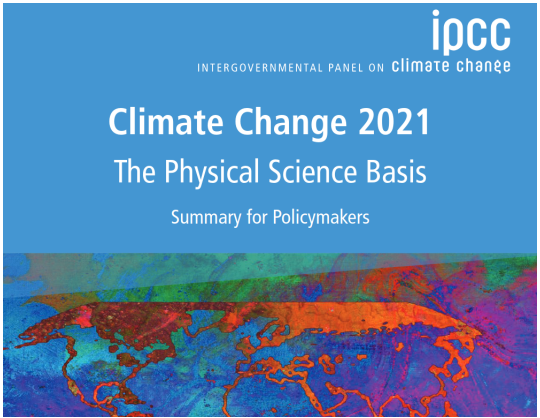
How Should a Researcher (in CS) React?



Possible answers:
— I don't care
— I do care, but not in my professional life

[1] *https://cacm.acm.org/magazines/2020/1/241717-publish-and-perish/fulltext*

How Should a Researcher (in CS) React?



Possible answers:
— I don't care
— I do care, but not in my professional life

— No research is neutral, what's my impact? I care also in my professional life: I stopped flying, and I started questioning my research objects.

See also *Publish and Perish - M. Vardi*[1].

[1] *https://cacm.acm.org/magazines/2020/1/241717-publish-and-perish/fulltext*

On Neutrality, by Howard Zinn:

This is not going to be a neutral class, I said. I don't believe in neutrality. I believe neutrality is impossible, because the world is already moving in certain directions. Wars are going on. Children are starving. And to be neutral, to pretend to neutrality, to not take a stand in a situation like that is to collaborate with whatever is going on, to allow it to happen. I did not want to be a collaborator with what was happening. I wanted to enter into history. I wanted to play a role. I wanted my students to play a role. I wanted us to intercede. I wanted my history to intercede and to take a stand on behalf of peace, on behalf of a racial equality or sexual equality, and so I wanted my students to know that right from the beginning, know **you can't be neutral on a moving train**[2].

[2] *http://firstrunfeatures.com/zinn.html*

The Current Green $\times$ IT Landscape

- Green IT:
  - Measures/estimations/modeling of (mainly) energy consumption
  - Optimization (SW, HW, communication)
- IT for Green:
  - Optimizations of existing non-IT domains (supply chain, smart-*)
  - New domains (car-sharing platforms)
- Climate Sciences and IT:
  - Instrumentation of physical phenomena
  - Modeling and simulation

This Talk

1. Motivations for a New Topic in the Green $\times$ IT Landscape: Limits
2. Two Domains and Lessons Learnt
3. Towards Thinking in Terms of Limits
4. This is Not a Conclusion

30 Years Playing with Constraints

- 30 years of research and teaching on (mainly critical and hard real-time) embedded systems; HW/SW interface, safety properties, high-level languages, model-driven implementations, virtual prototyping, constrained systems (time, memory, ...), long-term development, dealing with certification authorities, ...
- A domain where "try-a-bigger-machine" is not an option but there's no need to be faster than the music!

- *Collaborations with Airbus, STMicroelectronics, OrangeLabs, ...*
- *Application domains: avionics, railways, consumer electronics, sensor networks and smart cities, ...*

Current Personal Motivations: Questioning Sustainability of Digital Systems (and Research)

- Several sources of fragility: Security, Safety, Privacy, Loss of Expertise, Obsolescence, Dependency on HW Manufacturers, Democratic Choices
- How to stop the applications of facial recognition?
- Too many papers kill the publication system[3]
- **How to Estimate the Environmental Impacts of Digital Technologies? Questioning Optimizations vs Rebound Effects**

[3] *https://www.pnas.org/content/118/41/e2021636118*

How to Think in Terms of Limits?

REVIEW-ARTICLE

Computing within limits

Authors: Bonnie Nardi, Bill Tomlinson, Donald J. Patterson, Jay Chen, Daniel Pargman, Barath Raghavan, Birgit Penzenstadler Authors Info & Affiliations

Communications of the ACM, Volume 61, Issue 10 • October 2018 • pp 86–93 • https://doi.org/10.1145/3183582

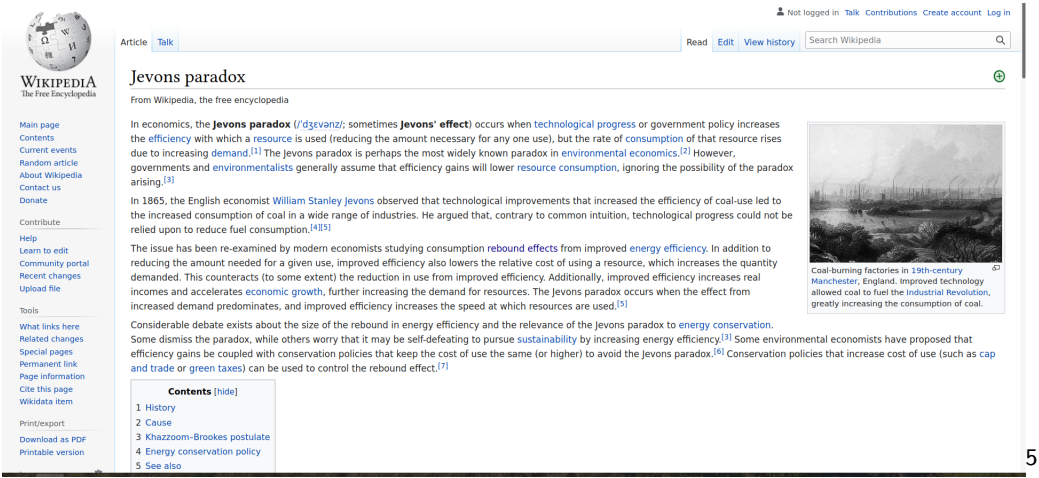
Published: 26 September 2018

Abstract

The future of computing research relies on addressing an array of limitations on a planetary scale.

4

The Jevons Paradox and Rebound Effects



Not logged in Talk Contributions Create account Log in

Article Talk | Read Edit View history | Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export
Download as PDF
Printable version

Jevons paradox

From Wikipedia, the free encyclopedia

In economics, the **Jevons paradox** (/ˈdʒɛvənz/; sometimes **Jevons' effect**) occurs when technological progress or government policy increases the efficiency with which a resource is used (reducing the amount necessary for any one use), but the rate of consumption of that resource rises due to increasing demand.[1] The Jevons paradox is perhaps the most widely known paradox in environmental economics.[2] However, governments and environmentalists generally assume that efficiency gains will lower resource consumption, ignoring the possibility of the paradox arising.[3]

In 1865, the English economist William Stanley Jevons observed that technological improvements that increased the efficiency of coal-use led to the increased consumption of coal in a wide range of industries. He argued that, contrary to common intuition, technological progress could not be relied upon to reduce fuel consumption.[4][5]

The issue has been re-examined by modern economists studying consumption rebound effects from improved energy efficiency. In addition to reducing the amount needed for a given use, improved efficiency also lowers the relative cost of using a resource, which increases the quantity demanded. This counteracts (to some extent) the reduction in use from improved efficiency. Additionally, improved efficiency increases real incomes and accelerates economic growth, further increasing the demand for resources. The Jevons paradox occurs when the effect from increased demand predominates, and improved efficiency increases the speed at which resources are used.[5]

Considerable debate exists about the size of the rebound in energy efficiency and the relevance of the Jevons paradox to energy conservation. Some dismiss the paradox, while others worry that it may be self-defeating to pursue sustainability by increasing energy efficiency.[3] Some environmental economists have proposed that efficiency gains be coupled with conservation policies that keep the cost of use the same (or higher) to avoid the Jevons paradox.[6] Conservation policies that increase cost of use (such as cap and trade or green taxes) can be used to control the rebound effect.[7]

Contents [hide]
1 History
2 Cause
3 Khazzoom-Brookes postulate
4 Energy conservation policy
5 See also

Coal-burning factories in 19th-century Manchester, England. Improved technology allowed coal to fuel the Industrial Revolution, greatly increasing the consumption of coal.

5

[5] *https://en.wikipedia.org/wiki/Jevons_paradox*

Typical Situation in 2005

Typical Situation in 2020

Mobile Communications 2005 - 2020

- 2005: Use them to place and receive calls "everywhere"; charge once a week; telephone booths remain;

Mobile Communications 2005 - 2020

- 2005: Use them to place and receive calls "everywhere"; charge once a week; telephone booths remain;
- 2005 ... 2020: Huge improvements of the devices (hardware, software, batteries, screens, casing, ...) + huge improvements of the infrastructure

Mobile Communications 2005 - 2020

- 2005: Use them to place and receive calls "everywhere"; charge once a week; telephone booths remain;
- 2005 ... 2020: Huge improvements of the devices (hardware, software, batteries, screens, casing, ...) + huge improvements of the infrastructure
- 2020: Use them mainly as portable always-connected computers; have allowed new services (Uber, maps+GPS, ...); charge twice-a-day or carry an external battery; telephone booths have disappeared; electric charging stations have appeared everywhere (bicycle-powered in railway stations, cafes, ...)

Evolution of the Global Impact of such Mobile Devices and the Underlying Infrastructure?

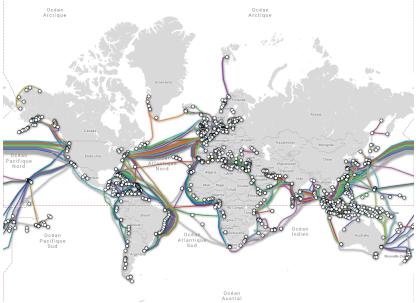
Both the potential uses and the environmental impacts increased a lot. Is it ok? Do smartphones replace (or rather add up to) something else that also has a very bad impact (laptops, cameras)? How to decide whether optimizations win over rebound effects?

These cannot be "tech-only" questions and answers. Currently reading:

6

[6] *https://www.sciencedirect.com/science/article/pii/S1364032121000769*

About the Infrastructure



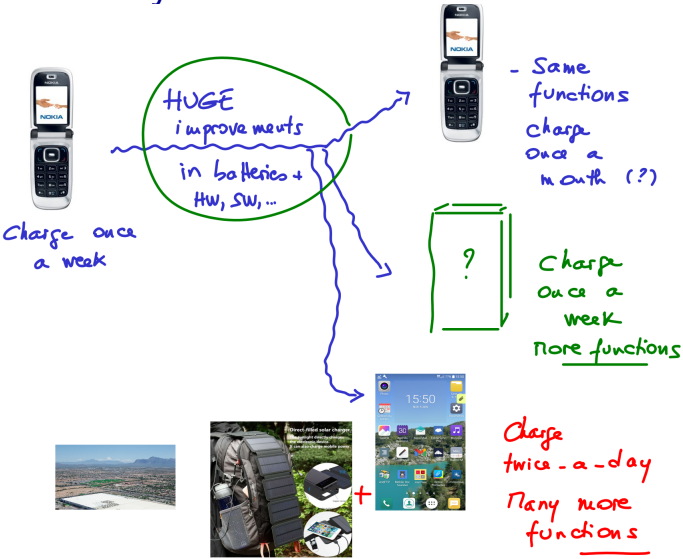
The internet of 2020 is represented by a large block indicating the weight of all the electrons running through it, with a much smaller block showing how it appeared in 2000.

78

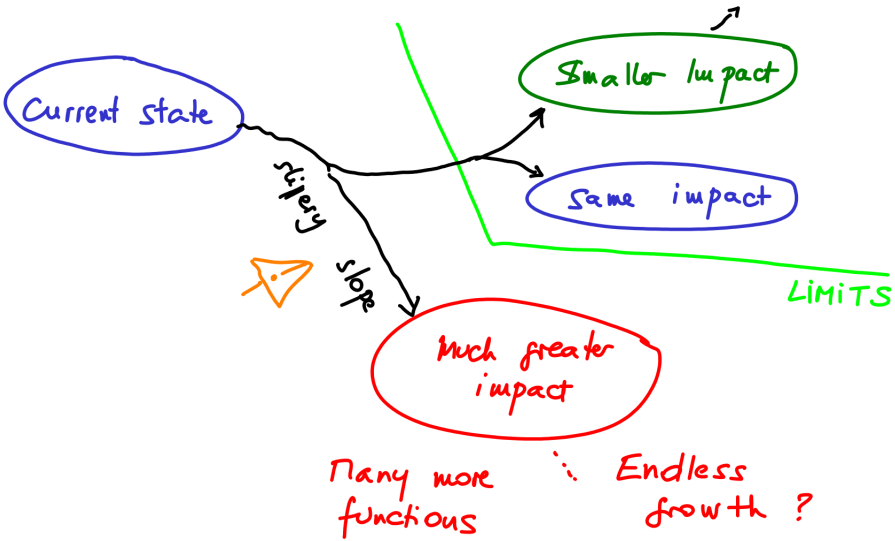
[7] *https://www.theguardian.com/artanddesign/2021/nov/03/art-shed-materialism-fragile-future-technology*

[8] *https://en.wikipedia.org/wiki/Submarine_communications_cable*

Theoretically Thinkable Paths



Several Paths...



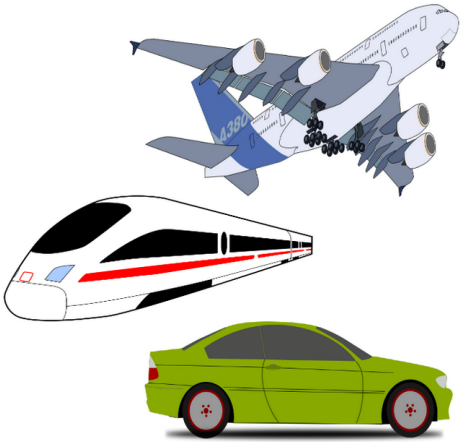
On the Need for a Better Research Coverage

Ensure not all research is devoted to the slippery slope.

Divert some energy and thoughts (hopefully money) to other paths...

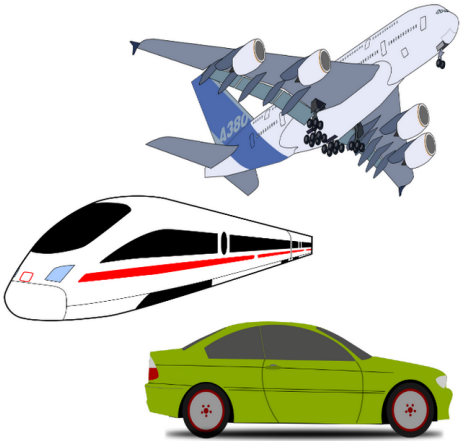
Just in case...

Hard Real-Time Critical Embedded Systems...



- Flight commands, ...
- Railways Signaling Systems, Onboard Systems
- ABS, Airbag, other car equipments...

Hard Real-Time Critical Embedded Systems...



- ~~*Flight* commands, ...~~
- **Railways Signaling Systems, Onboard Systems**
- ~~*ABS*, Airbag, other car equipments...~~

Some Characteristics of the Domain

- Hard real-time, Worst-Case-Execution-Time evaluation on modern HW
- Scarce resources (time, memory, heat, ...)
- Long-lived systems (obsolescence problem with HW platforms and OSes or compilers)
- Embedded: monitor-and-repair is not an option
- Flying: when an error occurs, no easy fail-safe behavior (contrary to trains)
- Certification authorities require predictability (through determinism)
- Domain-Specific-Languages like Lustre: no Dynamic Memory Allocations, Static Scheduling (no OS) approaches

Staying Within the Limits of Predictable Systems

- Do not use a general-purpose programming language
  A Domain-Specific-Language (DSL) can add a few things to a general-purpose language, but can be used to remove something from a GP language. Example: the total amount of needed memory should be statically computable.

Staying Within the Limits of Predictable Systems

- Do not use a general-purpose programming language
  A Domain-Specific-Language (DSL) can add a few things to a general-purpose language, but can be used to remove something from a GP language. Example: the total amount of needed memory should be statically computable.
- Do not use a general-purpose Operating System (OS), but a Real-Time OS

Staying Within the Limits of Predictable Systems

- Do not use a general-purpose programming language
  A Domain-Specific-Language (DSL) can add a few things to a general-purpose language, but can be used to remove something from a GP language. Example: the total amount of needed memory should be statically computable.
- Do not use a general-purpose Operating System (OS), but a Real-Time OS
- Select (or Redesign) HW that is not "too unpredictable"

Staying Within the Limits of Predictable Systems

- Do not use a general-purpose programming language
  A Domain-Specific-Language (DSL) can add a few things to a general-purpose language, but can be used to remove something from a GP language. Example: the total amount of needed memory should be statically computable.
- Do not use a general-purpose Operating System (OS), but a Real-Time OS
- Select (or Redesign) HW that is not "too unpredictable"
- Adopt a compilation point of view: automatic generation of code and automatic configuration of communications, from a single high level model.

Designing for 30+ years
Certification authorities introduce some friction

HW/SW Interface in Consumer Electronics Products

- Smartphones
- Set-top-boxes
- Cameras
- ...

Some Characteristics of the Domain

- Very hard time-2-market constraints (but more soft real-time constraints)
- Tyranny of the "new", "innovation" vs maintenance[9]
- Technically:
  - Communication between HW and SW designers is crucial because the HW keeps changing (hence the advent of concurrent engineering approaches like Transaction-Level Modeling (TLM), ...)
  - Aggressive optimizations for power consumption introduce "new" sources of bugs (aka "energy bugs", e.g., lock/unlock problems, battery drain, but also functional problems)

[9] *https://aeon.co/amp/essays/innovation-is-overvalued-maintenance-often-matters-more*

Limits? Where, and How?

- Intermittent Electricity?
- Individual "Ethical" and Economic choices ?
- Regulations?
- Chip shortage?

Limits? Where, and How?

- Intermittent Electricity?
- Individual "Ethical" and Economic choices ?
- Regulations?
- Chip shortage?

"We don't need ethics, we need regulations!" - Moshe Vardi

Lesson 1: Don't Forget the HW

- Building SW with constraints (e.g., timing predictability) if the HW is not built for that is a nightmare; a stack of very clever HW and SW layers that are meant to make your good against your own will is indistinguishable from magic

Lesson 1: Don't Forget the HW

- Building SW with constraints (e.g., timing predictability) if the HW is not built for that is a nightmare; a stack of very clever HW and SW layers that are meant to make your good against your own will is indistinguishable from magic
- Optimizing a device enables packing more functions within the same energy budget, but:
- If the HW is too complex, then the SW will not be able to exploit it fully; less "optimized" (or sophisticated) may actually lead to more efficient HW+SW systems!

Lesson 2: Designing for 30+ Years, or for Next Christmas

- Marketing constraints imply a lot of waste
  (even the Fairphone 4[10] presentation mentions *"ultra 5G"* and a *"48 MP camera"* along with advocating for durability)

- "Old" HW may become unavailable for long-lived systems, certification processes tend to freeze designs (for good reasons), hence obsolescence problems

[10] *https://shop.fairphone.com/fr/*

Lesson 3: Simplifying versus Optimizing?

- In critical systems: simplicity used to come first
- In Consumer Electronics: optimizations come first, but: new bugs, SW not always exploits complex HW fully, and there are (uncontrollable) rebound effects

The Current Green × IT Landscape

- Green IT:
  - Measures/estimations/modeling of energy consumption
  - Optimization (SW, HW, communication) vs Rebound effects?
- IT for Green:
  - Optimizations of existing non-IT domains (supply chain, smart-*)
  - New domains (car-sharing platforms)
- Climate Sciences and IT:
  - Instrumentation of physical phenomena
  - Modeling and simulation

The Current Green × IT Landscape

- Green IT:
  - Measures/estimations/modeling of energy consumption
  - Optimization (SW, HW, communication) vs Rebound effects?
- IT for Green:
  - Optimizations of existing non-IT domains (supply chain, smart-*)
  - New domains (car-sharing platforms)
- Climate Sciences and IT:
  - Instrumentation of physical phenomena
  - Modeling and simulation

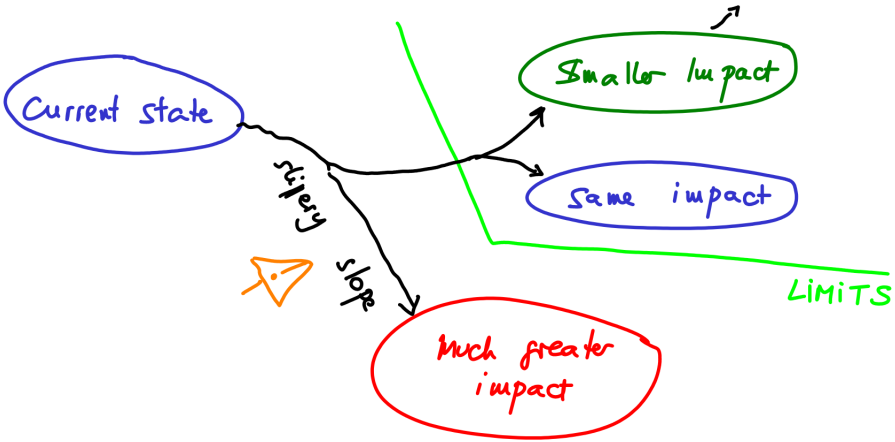
Thinking in terms of limits

Limits, Computing within Limits

- LIMITS series of workshops[11], Motto: *prepare a future of scarcity, in a world of abundant resources*
- Article "Computing Within Limits" [12]
- *Related approaches: collapse informatics* (see above paper)

[11] *https://computingwithinlimits.org/2021/*
[12] *https://dl.acm.org/doi/10.1145/3183582*

How to Avoid the Slippery Slope?



How to Define/Impose Limits for Digital Systems?

Technical approach — for people coming from already-constrained (and somewhat limited) contexts:

- De-Construct to Identify anti-limits (= intrinsically unbounded contexts)
- Rebuild from scratch as a thought-experiment (to escape the tyranny of the state of affairs), self-impose limits on the way

Not only technical:

- Decide beforehand what to do with the gains of optimized systems
- Think in terms of priorities and choices (but avoid moral judgements)

There Are Anti-Limits in a Digital System if it...

- Requires an increasing amount of resources globally (bitcoin alone, or with other crypto-currencies, Chia (proof of space)[13], PKT (proof of bandwidth)[14], NFTs, etc.)

[13] *https://en.wikipedia.org/wiki/Chia_(cryptocurrency)*
[14] *https://pkt.cash/*

There Are Anti-Limits in a Digital System if it...

- Requires an increasing amount of resources globally (bitcoin alone, or with other crypto-currencies, Chia (proof of space)[13], PKT (proof of bandwidth)[14], NFTs, etc.)
- Provisions resources for immediate service delivery, whatever the number of clients

[13] *https://en.wikipedia.org/wiki/Chia_(cryptocurrency)*
[14] *https://pkt.cash/*

There Are Anti-Limits in a Digital System if it...

- Requires an increasing amount of resources globally (bitcoin alone, or with other crypto-currencies, Chia (proof of space)[13], PKT (proof of bandwidth)[14], NFTs, etc.)
- Provisions resources for immediate service delivery, whatever the number of clients
- Assumes unlimited storage — in space and time (initial advertisement of gmail)

[13] *https://en.wikipedia.org/wiki/Chia_(cryptocurrency)*
[14] *https://pkt.cash/*

There Are Anti-Limits in a Digital System if it...

- Requires an increasing amount of resources globally (bitcoin alone, or with other crypto-currencies, Chia (proof of space)[13], PKT (proof of bandwidth)[14], NFTs, etc.)
- Provisions resources for immediate service delivery, whatever the number of clients
- Assumes unlimited storage — in space and time (initial advertisement of gmail)
- Assumes (HW+SW+vendor cloud) availability with no time limit (home automation)

[13] *https://en.wikipedia.org/wiki/Chia_(cryptocurrency)*
[14] *https://pkt.cash/*

There Are Anti-Limits in a Digital System if it...

- Requires an increasing amount of resources globally (bitcoin alone, or with other crypto-currencies, Chia (proof of space)[13], PKT (proof of bandwidth)[14], NFTs, etc.)
- Provisions resources for immediate service delivery, whatever the number of clients
- Assumes unlimited storage — in space and time (initial advertisement of gmail)
- Assumes (HW+SW+vendor cloud) availability with no time limit (home automation)
- (Needs 24/7 connectivity and cloud access) $\times$ (growing number of users)

[13] *https://en.wikipedia.org/wiki/Chia_(cryptocurrency)*
[14] *https://pkt.cash/*

There Are Anti-Limits in a Digital System if it...

- Requires an increasing amount of resources globally (bitcoin alone, or with other crypto-currencies, Chia (proof of space)[13], PKT (proof of bandwidth)[14], NFTs, etc.)
- Provisions resources for immediate service delivery, whatever the number of clients
- Assumes unlimited storage — in space and time (initial advertisement of gmail)
- Assumes (HW+SW+vendor cloud) availability with no time limit (home automation)
- (Needs 24/7 connectivity and cloud access) $\times$ (growing number of users)
- Built to allow for "unlimited" functional extensions (web)

[13] *https://en.wikipedia.org/wiki/Chia_(cryptocurrency)*
[14] *https://pkt.cash/*

There Are Anti-Limits in a Digital System if it...

- Requires an increasing amount of resources globally (bitcoin alone, or with other crypto-currencies, Chia (proof of space)[13], PKT (proof of bandwidth)[14], NFTs, etc.)
- Provisions resources for immediate service delivery, whatever the number of clients
- Assumes unlimited storage — in space and time (initial advertisement of gmail)
- Assumes (HW+SW+vendor cloud) availability with no time limit (home automation)
- (Needs 24/7 connectivity and cloud access) $\times$ (growing number of users)
- Built to allow for "unlimited" functional extensions (web)
- Bets on the availability of a better/bigger/more efficient machine, next year (SW obesity)

[13] *https://en.wikipedia.org/wiki/Chia_(cryptocurrency)*
[14] *https://pkt.cash/*

There Are Anti-Limits in a Digital System if it...

- Requires an increasing amount of resources globally (bitcoin alone, or with other crypto-currencies, Chia (proof of space)[13], PKT (proof of bandwidth)[14], NFTs, etc.)
- Provisions resources for immediate service delivery, whatever the number of clients
- Assumes unlimited storage — in space and time (initial advertisement of gmail)
- Assumes (HW+SW+vendor cloud) availability with no time limit (home automation)
- (Needs 24/7 connectivity and cloud access) $\times$ (growing number of users)
- Built to allow for "unlimited" functional extensions (web)
- Bets on the availability of a better/bigger/more efficient machine, next year (SW obesity)
- Deployment is profitable only if there are more users and/or increasing usage per user (5G)

[13] *https://en.wikipedia.org/wiki/Chia_(cryptocurrency)*
[14] *https://pkt.cash/*

Candidate Limits (and Induced Constraints)

- Gemini (heavier than gopher, lighter than the web, ...)[15] - same idea as restricted DSLs, no images, no extensions, ...

[15] *https://gemini.circumlunar.space/*
[16] *https://collapseos.org/*

Candidate Limits (and Induced Constraints)

- Gemini (heavier than gopher, lighter than the web, ...)[15] - same idea as restricted DSLs, no images, no extensions, ...
- All configurations in:
  $\{$ intermittent, quotas $\} \times \{$ power, connection, memory, computing power $\}$

[15] *https://gemini.circumlunar.space/*
[16] *https://collapseos.org/*

Candidate Limits (and Induced Constraints)

- Gemini (heavier than gopher, lighter than the web, ...)[15] - same idea as restricted DSLs, no images, no extensions, ...
- All configurations in:
  $\{$ intermittent, quotas $\} \times \{$ power, connection, memory, computing power $\}$
- no centralized architecture, no cloud, no network, no immediate service delivery

[15] *https://gemini.circumlunar.space/*
[16] *https://collapseos.org/*

Candidate Limits (and Induced Constraints)

- Gemini (heavier than gopher, lighter than the web, ...)[15] - same idea as restricted DSLs, no images, no extensions, ...
- All configurations in:
  $\{$ intermittent, quotas $\} \times \{$ power, connection, memory, computing power $\}$
- no centralized architecture, no cloud, no network, no immediate service delivery
- Carefully chosen DSLs (no dynamic allocation, ...) everywhere

[15] *https://gemini.circumlunar.space/*
[16] *https://collapseos.org/*

Candidate Limits (and Induced Constraints)

- Gemini (heavier than gopher, lighter than the web, ...)[15] - same idea as restricted DSLs, no images, no extensions, ...
- All configurations in:
  $\{$ intermittent, quotas $\} \times \{$ power, connection, memory, computing power $\}$
- no centralized architecture, no cloud, no network, no immediate service delivery
- Carefully chosen DSLs (no dynamic allocation, ...) everywhere
- The Ultimate Limit: What if we Stopped Manufacturing New HW Now?

[15] *https://gemini.circumlunar.space/*
[16] *https://collapseos.org/*

Candidate Limits (and Induced Constraints)

- Gemini (heavier than gopher, lighter than the web, ...)[15] - same idea as restricted DSLs, no images, no extensions, ...
- All configurations in:
  $\{$ intermittent, quotas $\} \times \{$ power, connection, memory, computing power $\}$
- no centralized architecture, no cloud, no network, no immediate service delivery
- Carefully chosen DSLs (no dynamic allocation, ...) everywhere
- The Ultimate Limit: What if we Stopped Manufacturing New HW Now? *See also "collapse informatics", e.g., CollapseOS*[16]

[15] *https://gemini.circumlunar.space/*
[16] *https://collapseos.org/*

An Example Question: is Extensibility a Desirable Property?

Extensibility[17] *is a software engineering and systems design principle that provides for* **future growth.** *Extensibility is a measure of the ability to extend a system and the level of effort required to implement the extension. Extensions can be through the addition of new functionality or through modification of existing functionality. The principle provides for enhancements without impairing existing system functions.*

An extensible system is one whose internal structure and dataflow are minimally or not affected by new or modified functionality, for example recompiling or changing the original source code might be unnecessary when changing a system's behavior, either by the creator or other programmers. (...)

Isn't it a slippery slope towards overshoot solutions?

[17] *https://en.wikipedia.org/wiki/Extensibility*

Let us Think of Alternate Futures

- Optimizing digital systems is ok only if we can control rebound effects
- What if we designed non-extensible systems, on purpose?
- Start thinking in terms of limits, just in case...
- Be prepared to trade convenience for a guarantee on the absence of anti-limits

The End. Thank you.
Questions ?