AZERTY amélioré: computational design on a national scale

Anna Feit, Mathieu Nancel, Maximilian John, Andreas Karrenbauer, Daryl Weir, Antti Oulasvirta

A new French keyboard standard is the first designed with the help of computational methods.

BY ANNA MARIA FEIT, MATHIEU NANCEL, MAXIMILIAN JOHN, ANDREAS KARRENBAUER, DARYL WEIR, AND ANTTI OULASVIRTA

AZERTY amélioré: Computational Design on a National Scale

IN 2015, FRANCE'S Ministry of Culture wrote to the French Parliament[4] criticizing the lack of standards for a keyboard layout. It pointed out that AZERTY, the traditional layout, lacks special characters needed for "proper" French and that many variants exist. The national organization for standardization, AFNOR, was tasked with producing a standard.[5] We joined this project in 2016 as experts in text entry and optimization.

THE FRENCH LANGUAGE uses accents (for example, é, à, î), ligatures (œ and æ), and specific apostrophes and quotation marks (for example,' « » " "). Some are awkward to reach or even unavailable with AZERTY (Figure 1), and many characters used in French dialects

are unsupported. Similar-looking characters can be used in place of some missing ones, as with " for ", or ae for æ. Users often rely on software-driven autocompletion or autocorrection for these. Also, they insert rarely used characters via Alt codes, from menus, or by copy-pasting from elsewhere. The ministry was concerned that this hinders proper use of the language. For example, some French people were taught, incorrectly, that accents for capital letters (for example, É, À) are optional, a belief sometimes justified by reference to their absence from AZERTY.

This article reports experiences and insights from a national-scale effort at redesigning and standardizing the spe-

cial-character layout of AZERTY with the aid of *combinatorial optimization.* Coming from computer science, our starting point was the known formulation of keyboard design as a classical optimization problem,[2] although no computationally designed keyboard thus far has been adopted as a nationwide standard. The specific design task is shown in Figure 2. Going beyond prior work, our goal was not only to ensure high typing performance but also to consider ergonomics and learnability factors.

However, the typical "one-shot" view of optimization, in which a user defines a problem and selects a solver, offers poor support for such complex socio-technical endeavors. The goals and decisions evolved considerably throughout the three-year project. Many stakeholders were involved, with various fields of expertise, and the public was consulted.[19,20] A key takeaway from this case is that algorithmic methods must operate in an interactive, iterative, and participatory manner, aiding in defining, exploring, deciding, and finalizing the design in a multi-stakeholder project.

In this article, we discuss how interactive tools were used to find a jointly agreed definition of what makes a good keyboard layout: familiarity versus user performance, expanded character sets versus discoverability, and support for

» key insights

- **France is the first country in the world to adopt a keyboard standard informed by computational methods, improving the performance, ergonomics, and intuitiveness of the keyboard while enabling input of many more characters.**

- **We describe a human-centric approach developed jointly with stakeholders to utilize computational methods in the decision process not only to solve a well-defined problem but also to understand the design requirements, to inform subjective views, or to communicate the outcomes.**

- **To be more broadly useful, research must develop computational methods that can be used in a participatory and inclusive fashion respecting the different needs and roles of stakeholders.**

everyday language versus programming or regional dialects. Interactive tools are also needed to elicit subjective preferences[11] and to help stakeholders understand the consequences of their choices.

Although only time will tell whether the new layout is adopted, one can draw several lessons from this case. Computational methods become essential not only to solve a well-stated design problem but also to understand it and to communicate and appreciate its final outcomes. They are needed to elicit and inform subjective views, and to resolve conflicts and support consensus by presenting the best compromises achievable. This yields a vastly different picture of optimization and algorithmic tools, revealing important opportunities for research to better support participatory use.

Goals for Revising AZERTY

The AFNOR committee concerned with the development of a standard for the French keyboard was composed of Ministry of Culture representatives


Figure 1. The old AZERTY layout. Try typing « À l'évidence, l'œnologie est plus qu'un 'hobby'. » ("Evidently, wine-making is more than a 'hobby'.") Hint: the underlined characters are not present, such as nonbreaking spaces and curved apostrophes.
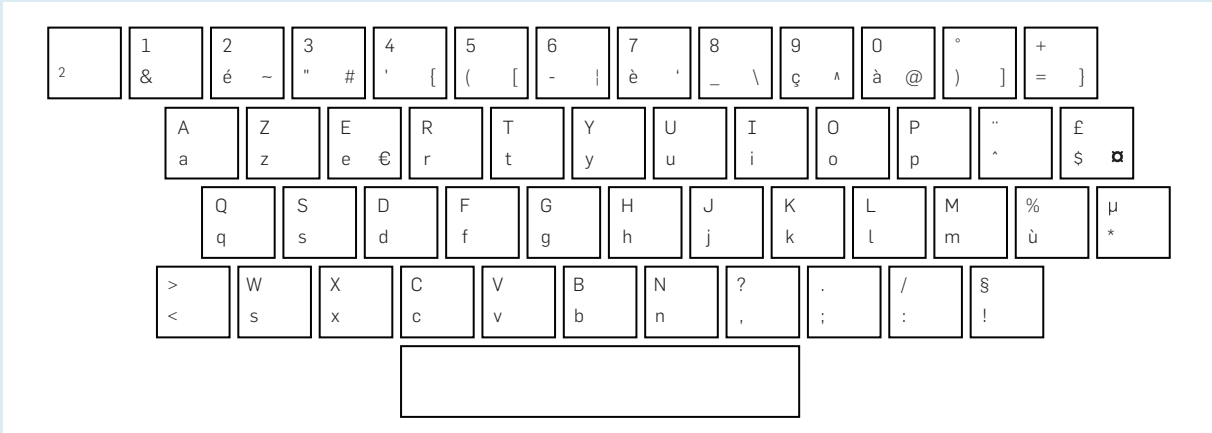
Figure 2. The computational goal was to assign the special characters to the available keyslots such that the keyboard is easy to use and typing French is fast and ergonomic. The process saw the set of characters change frequently; in (a), the set in the final layout is shown (the last 24 characters displayed were not part of the optimization problem but added later).
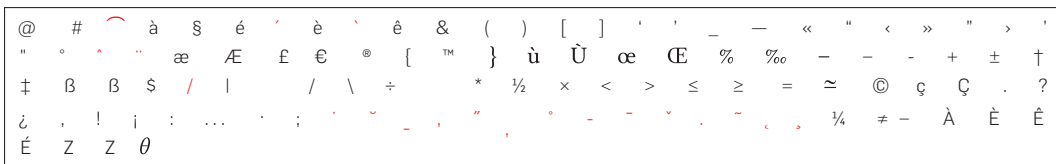
(a) Example set of special characters (107). In red are diacritic marks; entered via dead keys; see pane d.
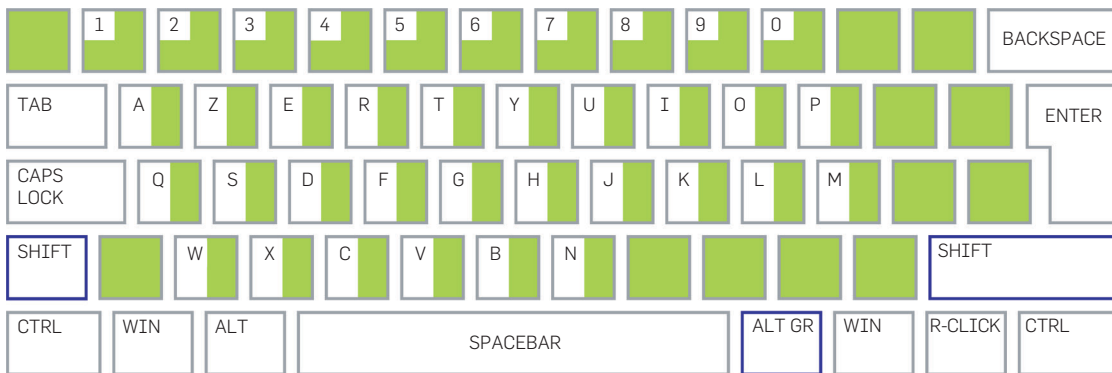
(b) The set of keyslots (129), with green for free keyslots and blue for modifier keys to access each of the four slots per key.
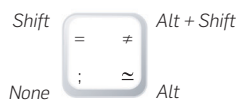
(c) Each key offers four keyslots accessible via modifier keys.

(d) Diacritics work as dead keys: visible output is produced upon subsequent input of a letter.

and experts in ergonomics, typography, human-computer interaction, linguistics, and keyboard manufacturing. A typical standardization process involves meetings to iterate over each aspect of the standard and its wording. Final drafts are opened to public comment on which the committee then iterates if need be. At the start of the project, we took these meetings as an opportunity to understand the requirements of the design problem from a human-centered perspective. We then formulated them in a way that enabled modeling and solving the problem using optimization.

Our task was to develop an improved layout for all so-called "special characters", that is, every character that is not a nonaccented letter of the Latin alphabet ("AZERTYUIOP..."), a digit, or the space bar. The list of special characters to be made accessible was greatly augmented compared to the traditional AZERTY layout, to facilitate the typing of all characters used in the French language and its dialects,[a] modern computer use (especially programming and social media), and scientific and mathematical characters (for example, Greek letters), alongside major currency symbols and all characters in Europe's other Latin-alphabet languages. Despite having to add many new characters, we strove to keep the layout usable, ergonomic, and easy to learn.

There were several challenging requirements (Figure 2). The physical layout follows the alphanumeric section of the ISO/IEC 9995-112 standard. Each key can hold up to four characters, using combinations of the Shift and AltGr modifiers (Figure 2c). For nonaccented letters, digits, and the space, the layout had to remain as in traditional azerty, leaving 129 keyslots (see Figure 2b). The only characters that could be added or moved were the special characters described in Figure 2a; their number, up to 122, varied throughout the project as new suggestions were made and discussed. Combining diacritical characters, like accents, are entered via "dead keys," as explained in Figure 2d.

Note that the requirements and constraints of this project evolved dramatically as it progressed, depending on

---

a   https://bit.ly/32ZGnQh

Despite having to add many new characters, we strove to keep the layout usable, ergonomic, and easy to use.

intermediate solutions, priorities updates, public requests, and so on. We detail these changes in the later text.

Keyboard Design as an Optimization Problem

The arrangement of characters in a layout is a very challenging computational problem. Formally, one must assign characters to the keyboard keys and to keyslots accessible via modifier keys. Each assignment involves three challenging considerations. We here discuss the computational problem before opening up approaches to making them useful in a multi-stakeholder design project.

Firstly, what is a "good" placement? Ergonomics and motor performance should be central goals. More common characters should be assigned to keys that minimize risks of health issues such as repetitive strain injury and that are quickly accessed. However, people differ in how they type.[7] There is no standard model that can be used as an objective function. Also, time spent visually seeking a character should be minimized through, for example, placing characters where people assume they are,[14] and grouping characters that are considered similar.

Secondly, which level of language to favor is tricky to know in advance and, as we learned, a politically loaded question. To decide where to put #, we must weigh the importance of programming or social-media-type language in which that character might be common, against "proper" literary French in which it is rare. Decisions on character positions mean trading off many such factors for a large range of users and typing tasks.

Finally, there is a very large number of possible designs, up to 10^{213} distinct combinations for assigning characters to keyslots in our case. Text input is a sequential process wherein entering a character depends on the previously typed one. Therefore, finding the best layout for typing is an instance of the quadratic assignment problem (QAP).[2,6,16] These are not only hard to solve in theory (NP-hard to approximate within any constant factor[22]); there still exist unsolved instances of QAPs, published as benchmarks decades ago, with only 30 items,[3] a far cry from 129.

An optimization model for typing special characters. The design problem was formulated as an integer program (IP), which lets us use effective solvers that provide intermediate solutions with bounds on their distance from optimality. We use binary decision variables x_{ik} to capture whether character i is assigned to keyslot k or not. The criteria, and corresponding IP constraints, are formulated in Table 1. Every feasible binary solution corresponds to a keyboard layout. An objective function measures the goodness of each layout according to each of the criteria. The parameters, constraints, and objectives of the integer program reflect the standardization committee's goals: facilitate typing of correct French, enable the input of certain characters not supported by the current keyboard, and minimize learning time by guaranteeing an intuitive to use keyboard that is sufficiently similar to the previous AZERTY.

The challenge for us was to translate goals such as "facilitate typing and learning" into quantifiable objective functions. We ended up defining four objective criteria, which were combined in a weighted sum to yield a single objective function:[6,19] **Performance** (minimizing movement time), **Ergonomics** (minimizing risks of strain), **Intuitiveness** (grouping similar characters together), and **Familiarity** (minimizing differences from AZERTY). Table 1 presents our formulation of the integer program and articulates the intuition behind each criterion.

The criteria here rely on input data that reflect the real-world typing of tens of millions of French users. Therefore, we gathered large text corpora, with varied topics and writing styles, and weighted them in accordance with the committee's requests. We focused on three typical uses. Formal text is well-written, curated text with correct French and proper use of

special characters. Sources include the French Wikipedia, official policy documents, and professionally transcribed radio shows. Informal text (for example, in social-media or personal communication) has lower standards of orthographic, grammar, and typographic correctness. The material includes anonymized email and popular accounts' Facebook posts and Tweets. The Programming corpora comprise content representative of common programming and description languages: Python, C++, Java, JavaScript, HTML, and CSS, with comments removed. Several of our Formal-and Popular-class corpora were provided by the ELDA.[8,b] Frequencies were computed by corpus, then averaged per character and class, and finally assigned weights subject to committee discussion (Formal: 0.7, In-

---

b   The Evaluations and Language resources Distribution Agency; see http://www.elra.info/en/about/elda/.

Table 1. The integer programming formulation of the keyboard design problem. The objective function is a weighted sum over four normalized criteria. Only basic assignment constraints are shown. Throughout the project, additional constraints were added or removed for particular instances.[6] For the instance that led to the standardized layout ($N = 85$, $M = 129$), the following weights were chosen: $w_P = 0.3$, $w_E = 0.25$, $w_I = 0.35$, $w_F = 0.1$.

min $\quad w_P \sum_{i=1}^{N}\sum_{k=1}^{M}\sum_{c=1}^{27}(p_{ci}T_{ck} + p_{ic}T_{kc})x_{ik}$

Performance (P):
Guarantees that frequent special characters can be quickly entered in combination with the fixed letters. It is quantified by computing the average time to type a special character before or after any of the regular letters (T_{ck}, T_{kc}), weighted by the special-character–regular-letter pair (p_{ci}, p_{ic}). The corresponding data were gathered in a crowdsourcing-based study.

$+ \quad w_E \sum_{i=1}^{N}\sum_{k=1}^{M} p_i(W_k + F_k + M_k)x_{ik}$

Ergonomics (E):
Penalizes keyslots that require extreme movements putting strain on tendons and joints, which are empirically associated with repetitive strain injuries[24]: extreme outward or inward movements of the wrist ($W_k \in \{0, 1\}$), extreme extension of fingers ($F_k \in \{0, 1\}$), and use of one or two modifier keys ($M_i \in \{0, 1, 2\}$). The score is weighted by the frequency (p_i) of the character assigned to the keyslot.

$+ \quad w_I\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{M}\sum_{l=1}^{M}s_{ij}D_{kl}x_{ik}x_{jl} + \sum_{i=1}^{N}\sum_{k=1}^{M}\sum_{c=1}^{27}s_{ic}D_{kc}x_{ik}\right)$

Intuitiveness (I):
Minimizes the distance between similar special characters (D_{kl}) and between special characters and similar letters (D_{kc}), to facilitate discovery and learning.[17] This similarity can be syntactic or semantic and is captured by the scores s_{ij}, s_{ic}. All characters are considered equally important for grouping.

$+ \quad w_F \sum_{i=1}^{N}\sum_{k=1}^{M} p_i D_{k,\mathcal{A}(i)}x_{ik}$

Familiarity (F):
Places frequent characters near the position familiar from traditional AZERTY, to facilitate visual search with the new layout.[14] $D_{k,\mathcal{A}(i)}$ quantifies the distance between the keyslot k assigned to the character i and its AZERTY position, weighted by that character's frequency (p_i).

subject to

$\sum_{k=1}^{M} x_{ik} = 1 \quad \forall i \in \{1, \dots N\}$

Ensures each character is assigned to one keyslot.

$\sum_{i=1}^{N} x_{ik} \leq 1 \quad \forall k \in \{1, \dots M\}$

Ensures no keyslot is assigned to multiple characters.

$x_{ik} \in \{0, 1\} \quad \forall i \in \{1, \dots N\}, k \in \{1, \dots M\}$

formal: 0.15, Programming: 0.15). Table 2 shows the most common characters in each category.

For estimating key-selection times, we gathered an extensive dataset of key-to-key typing durations to capture how people type in terms of the Performance objective. In particular, we were interested in how soon a special character keyslot (in green in Figure 2) could be accessed before or after a regular letter. In a crowdsourcing-based study, we asked about 900 participants to type word-like sequences of nonaccented letters that each had one special character slot in the middle,[6,19] for example, "buve Alt+Shift+2 ihup." We gathered time data for all combinations of letters and special character slots (7560 distinct key pairs).

For the Intuitiveness objective, we defined a similarity score between characters as a scalar in the range [0, 1], depending on visual resemblance (for example, R and ®, _ and -), semantic proximity (for example, × and *, or ÷ and /), inclusion of other letters (for example, ç and c, œ and o), frequent association in practice (for example, n and ~, e and ´), or use-based criteria such as lowercase/uppercase and opening/closing character pairs. These weights, and the similarities to consider and give priority, were discussed at length with the committee and frequently updated throughout the project, especially after the public comment.

Solving the QAP. Branch-and-bound[1] is a standard approach to solve integer optimization problems. It relies on relaxations that can be solved efficiently (for example, by linearizing the quadratic terms and dropping the integrality constraints). In the powerful RLT1 approach,[10] every quadratic term of the form $x_{ik} \cdot x_{k\ell}$ is replaced with a new linear variable $y_{ijk\ell}$. Although this linearization produces very good lower bounds, it introduces $\mathcal{O}(n^4)$ additional variables, leading to a vast increase in problem size. We observed, however, that, although we have 100+ characters to place, our quadratic form is very sparse. Our approach exploits this sparsity, leading to a framework that synthesizes the concepts and benefits of powerful (but complex) linearization and column-generation technique.[13]

In our adaptation, only a subset of variables is part of the initial instance, and further variables are generated iteratively "on the fly" as they become relevant. The idea is as follows: we start with the easy-to-solve linear part of the objective and ignore any quadratic terms at first. Iteratively, we generate the RLT1 relaxation of those quadratic terms $a_{ijk\ell} \cdot x_{ik}x_{j\ell}$ where at least one of the variables x_{ik} and $x_{j\ell}$ is set to 1 in the previous optimal solution and where $a_{ijk\ell}$ has a substantial contribution to the objective value; in particular, we do not generate any $y_{ijk\ell}$ where $a_{ijk\ell} = 0$. We thereby take advantage of the sparsity of the quadratic objective function, which allows us to introduce only a few additional variables in every iteration. After enhancing our model with these variables, we reoptimize until the addition of further terms does not significantly increase the objective value and the desired optimality gap is reached. This algorithm provides a hierarchy of lower bounds with every iteration producing a bound that is at least as good as the one from the previous iteration. For the problem instance that led to the final standardized layout, we could thus demonstrate that a very small gap (<2%) exists between the computed and the optimal solution after only five days computation time. Note that, thanks to the sparsity, the formulations used in every iteration stay relatively small, enabling us to solve larger problem instances with less time and memory than the traditional complete RLT1 relaxation.

Introducing Optimization Tools in the Standardization Process

The optimization approach described permits a principled approach to solving the keyboard layout problem. However, we quickly learned that a one-shot approach to optimization is not actionable in a complex, multi-stakeholder design project. The problem definition and expectations from stakeholders were ill-defined in the beginning and constantly evolving: definitions, parameters, and objectives changed, and decisions often hinged on subjective opinions, public feedback, or cultural norms, making them hard to express mathematically. We therefore ended up developing several approaches that helped integrate computational methods into the operational mode of the standardization committee.

When we first joined the project, the committee was debating each

Table 2. The highest-frequency special characters, by category of French text.

Formal		Informal		Programming	
Char.	**Freq. (%)**	**Char.**	**Freq. (%)**	**Char.**	**Freq. (%)**
é	1.883	#	1.139	.	1.584
,	0.896	é	1.074	-	1.315
.	0.796	/	0.895	(	1.310
'	0.765	.	0.805	)	1.309
à	0.332	!	0.712	;	1.158
-	0.262	@	0.648	=	1.035
è	0.241	:	0.497	_	1.002
)	0.156	'	0.457	,	0.926
(	0.141	,	0.447	:	0.922
:	0.135	à	0.269	"	0.918
'	0.118	-	0.209	>	0.527
ê	0.098	"	0.185	/	0.459
/	0.078	è	0.155	<	0.445
!	0.075	'	0.129	[	0.444
;	0.058	ê	0.099	]	0.443
"	0.047	_	0.079	'	0.292
»	0.041	;	0.075	[	0.186
ç	0.041	&	0.068	]	0.186
«	0.041	)	0.063	%	0.150
?	0.040	«	0.059	+	0.144

Interesting differences are visible. For instance, the mostly Internet-related characters # and @ appear in the table only for the Informal class. The common accented letters é, à, è, and ê are less frequent in Informal text than in the Formal corpora, although retaining the same relative order. Interestingly, / is present in all three columns, because of its wide range of uses.

character in hand-crafted layouts designed by individual members, with rationales such as

- "ê is **frequent**, so I gave it direct access because it's **faster**."
- "The guillemets (« ») are **important**, so they should be **easy to find**."
- "@ **looks like** a, so I've put them **close together**."
- "We should leave ç and ù where they are; otherwise, they will be **hard to find**."

Many of these rationales were based on intuition, even when the objective measurement (of frequency, speed, and so on) was possible. Our first challenge in defining the optimization problem was to turn these rationales into well-defined quantified objectives. These hand-crafted proposals were typically good in one sense (for example, aiming for speed) but compromising other objectives. They often generated ideas following a greedy approach: starting with what seemed important and then having to make do with the remaining free slots and characters. The outcome of such a process depends greatly on the choice order, and on the subjective weights given to each rationale, which could vary hugely between characters and stakeholders.

Our first task was to explain how a combinatorial approach can assist with such complex, multi-criterion problems. In contrast to *ad-hoc* designs, formulating the problem in quantifiable objective metrics allows algorithms to consider all objectives at once and explore all possible solutions. It also enables stakeholders to assign understandable weights to the task's many parameters, permitting exact control of their priorities. Also, the objective metrics scan be evaluated separately for assessing effects of manual changes; room is left for design decisions based on subjective criteria that cannot be formalized.

We built an evaluation tool that replicated the objective criteria calculations used by the optimizer and used it to quickly compare competing layouts for different objectives. This allowed us to illustrate how easily character-by-character layout design can lead to suboptimal results. For example, evaluating one of the handmade layouts

The challenge for us was to translate goals such as "facilitate typing and learning" into quantifiable objective functions.

with the objective functions described above, we found that typing special characters was 47% slower, 48% less ergonomic, and 17% less similar to the traditional AZERTY than was our final optimized solution, which formed the basis for the new layout.

Over the course of the project, there were two cycles of *optimizations* and *adjustments*, separated by public consultation (see Figure 3). Before that, over nine months, we defined and iterated the optimization model with the committee, formulating objective functions that matched members' intuitions and expectations (Table 1) and collected the text corpora. As we collected the input data, more subjective choices, such as character similarity, were discussed with the committee members and continuously adjusted over the course of the project. The first *optimization* phase entailed a five-month back-and-forth process between optimization and committee discussions. The optimizer computed solutions to numerous instances of the design problem, which we presented to the committee, explaining how inputs and constraints affected aspects of the designs. Members then suggested particular parameter settings or adding or removing constraints (for example, keeping capital and lowercase letters on the same key, changing the character sets, or weighting specific text corpora differently). We then optimized new layouts for these new parameters. After several such iterations, the committee agreed on the layout and parameter set it deemed best with regard to the optimization objectives.

Then, in the first *adjustments* phase, we used the optimizer to evaluate manual changes proposed by the experts. It was argued that these adjustments capture exceptions to the objectives, such as individuals' expectations and preferences, cultural norms, or character-specific political decisions that frequently changed with every iteration and could not be formally modeled. For example, the traditional position for the underscore was preferred for some solutions, thanks in part to nomenclature: it is colloquially called the "8's dash" (*tiret du 8*), for its location on the 8 key in AZERTY. The aforementioned evaluation tool helped us assess the consequences of these character moves or swaps on the four objective criteria.

Figure 3. Project timeline: computational methods were involved in all phases but the public comment and were governed by interactions with stakeholders and various computational methods developed by the researchers.
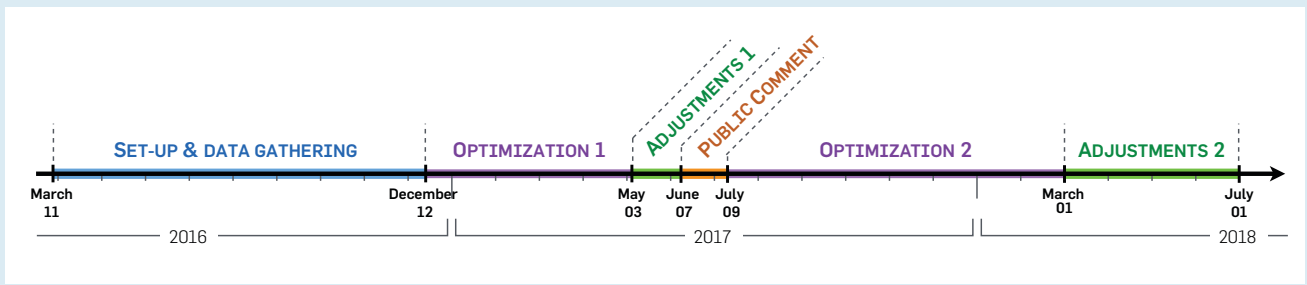
The committee hence could make better-informed decisions about trade-offs between adjustments. This led to the first release candidate.

In June 2017, this layout was presented to the public, which had 1 month, per AFNOR's standard procedure, to respond to the proposed standard and offer comments and suggestions. An unprecedented number of responses (over 3,700) were submitted, including numerous suggestions. Feedback was strongly divided on some matters, such as how strongly computer-programming-related characters should be favored, or where accentuated characters should be placed. The committee compiled the feedback into themes and tried to identify consensual topics. In some cases, there were opposing sides with no clear majority. For example, some people insisted that all pre-marked characters (for example, é à ç) be removed from the layout to make other characters more accessible, because the former could be entered using combining accents, whereas others argued for having even more pre-accentuated letters accessible directly. Consensus itself could also be difficult to assess: a subset of people argued that digits should be accessible without the Shift modifier, but it was not clear whether all of the remaining commenters were positive, neutral, or even gave any thought to keeping them "shifted." In such cases, the committee referred to the Ministry of Culture's stated objectives as well as to the experts' opinions on the available options: digits in our text corpora are much less frequent than some of the most used accentuated characters, and the change from the traditional AZERTY was deemed too large.

Consensual trends in the comments directly led to updates of the optimization model, its inputs, or parameters: characters were added or removed from the initial set; some associations were added to the Intuitiveness criterion and the weights of the criteria and corpora were updated. Hard constraints were added to the optimizer, such as having opening and closing character pairs (for example, [] {} "" «») placed on consecutive keys on the same row and with the same modifiers. Finally, the positions of @ and # were fixed to more accessible slots already used in alternative AZERTY layouts.

The second cycle then began, consisting of a (seven-month) *optimization* and (four-month) *adjustment* phase, similar to the ones described above.
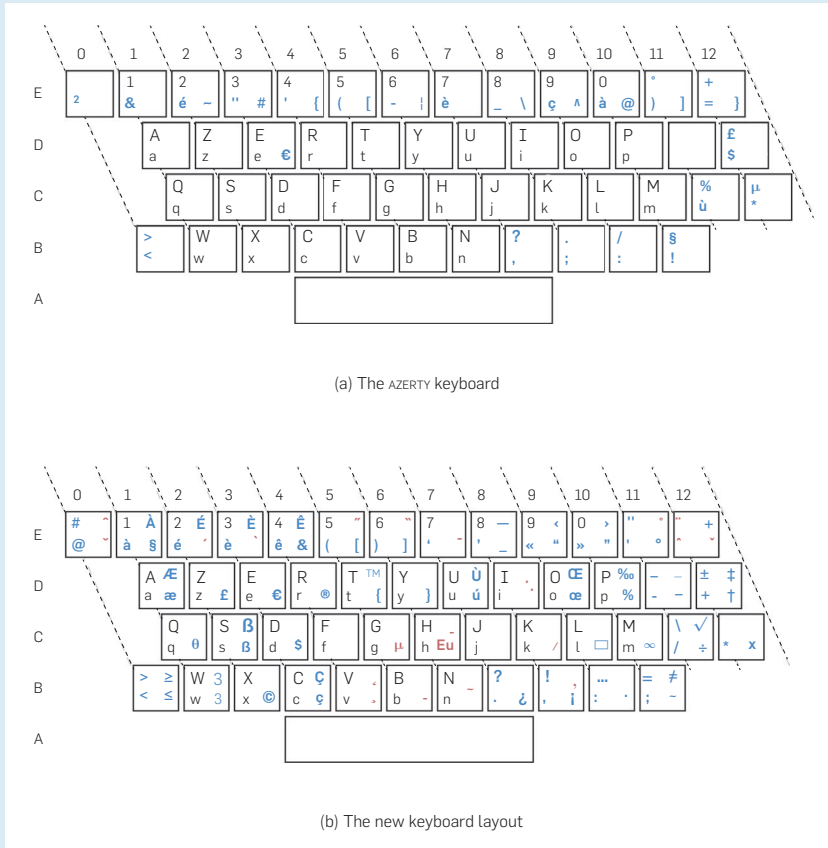
Figure 4 summarizes our approach to integrate computational methods into the standardization of the French keyboard and diagrams the interactions we developed. Our optimization tools proposed solutions and could be used to evaluate suggestions, which enabled an efficient exploration of the very large design space. On the other side, the experts steered the definition of objectives, set weights, and adjusted the input data. They used the optimized designs to explore the solution space and tweaked the computed layouts to consider tacit criteria too, such as political objectives and cultural norms. The evaluation tool could be used to study the consequences of conflicting views, for example, by quickly checking what happens to objective scores when a character is moved. Simultaneously, both sides were informed by comments from the public, whose expectations and wishes led the experts to question their assumptions and criteria and were directly implemented as

Figure 4. Diagram of our participatory optimization process wherein experts on a standardization committee define objectives and inputs to an optimizer, which, in turn, supplies concrete layouts, accompanied with feedback on quality (performance and intuitiveness, among others). The process was informed by feedback from the public.
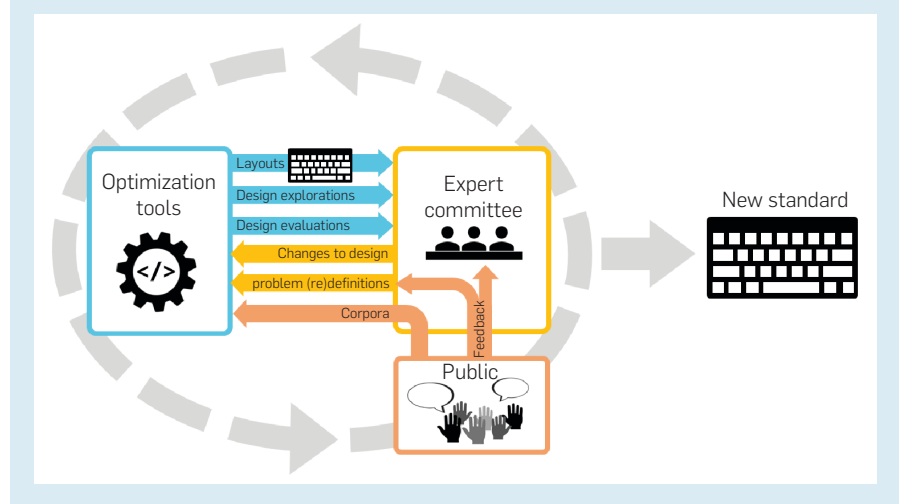
Figure 5. Comparison of the AZERTY and the new standardized layout. The characters included in the design problem are in boldface and color. Marked in red are dead keys, which require pressing a subsequent key before a symbol is produced (diacritical marks and mode keys for accessing non-French-language Latin characters, Greek letters, and currency symbols).

(a) The AZERTY keyboard

(b) The new keyboard layout

AZERTY.[c] It allows accessing all characters used in French without relying on software-side corrections. Frequently used French characters are accessible without any modifier (é, à, «, », and so on), or intuitively positioned where users can expect them (for example, œ on the o key). All accented capital letters (À and É, among others) can be entered directly or using a dead key. The main layout offers almost 60 characters not available in AZERTY for entering symbols used in math, linguistics, economics, programming, and other fields. Some programming characters, which often have alternative uses, were given more prominent slots; for instance, / became accessible without modifiers and \ is on the same key but in a shifted slot. According to the metrics described above, the performance and ergonomics of typing the special characters already present in traditional AZERTY are improved by 18.4% and 8.4%, respectively, even though the new layout had to accommodate 60 additional characters.

The keyboard offers three additional layers accessed via special mode keys. These are dedicated to European characters not used in French (via the Eu key from Alt+H in Figure 5b), currency symbols (via ¤ with Alt+F), and Greek letters (via Alt+G's μ), more than 80 additional characters in all. Their placement was beyond the scope of the optimization process, being near-nonexistent in our text corpora.

Its many changes notwithstanding, the layout maintains similarity to the traditional AZERTY, making the transition for users simple. Of the 45 special characters previously available, 8 retained their original location and 12 moved by less than three keys. In particular, frequently used characters were kept near their original position. For instance, the most common special character (é) is not in the fastest spot to access on average (B07 in our study) but stayed at E02 for similarity although maintaining good performance. Many punctuation characters (slots B7–B10) were moved slightly by the optimizer to better reflect character and character-pair frequencies (see Table 2) although remaining in the expected area of the keyboard. Comparing the final design to AZER-

changes in the weight and constraint definitions within the optimization model.

In summary, customized tools applying an established optimization approach allowed fast iteration and explainable results, and provided monitoring tools that enabled stakeholders to test and assess the effects of their ideas for every measurable objective goal, yielding transparent results. We arrived at the final layout by combining objective and subjective criteria weighted and refined through several iterations with computational tools. This involved hard facts whenever possible and factoring in numerous opinions not only from diverse experts but also from the public, the primary target of the new standard.

The New French Keyboard Standard

The outcome, shown in Figure 5b, makes it easier to type French and

enables accessing a larger set of characters. Despite the problem's computational complexity, we were able to propose a solution for which we could computationally verify that it is within 1.98% of the best achievable design with regard to the overall objective function and the final choice of parameters presented in Table 1. This means that it is either optimal or, if suboptimal, at most 1.98% worse than an unknown optimal design.[13] This solution was taken as a design basis, to which the committee added 24 further, rarer characters. Manual changes were made to accommodate these and locally optimize the layout's intuitiveness. All decisions were informed by our evaluation tool, allowing the committee to finely control the consequences of each manual change to the initial four objectives.

The new layout enables direct input of more than 190 special characters, a significant increase from the 47 of the current

---

c Not including accented characters that can be created using dead keys, such as ^ + I = Î.

TY based on our objective functions, we can see that all larger moves of characters had a clear justification, be it better performance, ergonomics, discoverability, or consistency. Most noteworthy was bringing paired characters such as parentheses and brackets closer together, a direct result of the public consultation.

Finally, substantial effort was devoted to forming semantic regions for characters, such as mathematical characters (C11–D12 and B12), common currency symbols (C02–D03), or quotation marks (E07–E11). Many of these groupings emerged during the optimization process, thanks to the Intuitiveness objective. Others resulted from manual changes when the committee decided to prioritize semantic grouping over performance or ergonomics (for example, following a calculator metaphor for mathematical characters). The Intuitiveness score improved more than fourfold (434.4%) relative to the traditional AZERTY.

Communication and adoption. We cannot predict the success of the new standard, nor how quickly users will adapt it. Being voluntary, its publication does not bind users nor manufacturers. We can, however, report first indicators of interest, as well as the French Ministry of Culture's plans to promote the new layout.

At least two manufacturers started producing physical keyboards engraved according to the new standard, of which already one was marketed by the end of 2019. We were also informed that Microsoft will integrate an official driver to Windows 10. Importantly, as an attempt to promote the use of the new layout, the French Ministry of Culture reported that they will replace the entire "fleet" of its employees' keyboards. We also received numerous emails from individuals motivated to write their own keyboard drivers and key-stickers, so they and others could use the layout before it is effectively commercialized. Only few months after the release of the standard, several drivers were available for Mac OSX, Windows 10, and Linux; some of them listed on our webpage.[d] These measures indicate the will and potential for nationwide adoption.

To inform users and encourage pub-

d See http://norme-AZERTY.fr/

lic acceptance, we published an interactive visualization of the keyboard online,[d] in which people can explore to discover the new layout and learn the reasoning behind it. It received more than 74,800 page views in the week following the official release event on April 2, 2019, and counted more than 122,000 views 5 months after the standard was published. For people interested in finer details, we also published an open-access document in French and English explaining the essence of our method in layman's terms.[8,19] This details the impact of the various corpora and weights involved in the calculations and in the committee's later deliberations.

Learnings and Outlook
The design of keyboards is a matter of economic, societal, and even medical interest. However, as most complex artifacts involving software do, they evolve by stacking layers on layers. Most keyboard layouts were designed decades ago or more. To respond to chang-

ing uses of language, from programming to social media, they have evolved incrementally via adding characters to unused keyboard slots. The absence of appropriate layout standards negatively affects the preservation and evolution of these languages. Indeed, it is startling that some of the world's most spoken languages[21] lack any government-approved keyboard standard: Punjabi (10th), Telegu (15th), and Marathi (19th).

Similarly, virtual (software) keyboards mostly follow agreed-on standards for alphanumeric characters, but special characters can be company-specific and vary greatly. Computational design methods could play a role in helping regulators improve quality and respond more swiftly to changes in computing and language, even "shaking up" a design if needed. The optimization methods and tools proposed here can be applied to other languages and input methods (for example, touchscreens) with adaptations to the input data and corresponding weights.

For keyboards and beyond, we be-

Table 3. Opportunities for improving the use of computational methods in large-scale design projects, identified on the basis of our experience in using combinatorial optimization for designing the French keyboard standard.

Facilitating participatory optimization
To support multistakeholder design projects, computational methods should be interactive, iterative, and participatory. Therefore we need tools that allow:

(1) **Fast (re)definition of the problem:**
In an iterative design process, the problem definition is constantly evolving. To speed up computation in cases of only slight changes in definition or instances, standard solvers should find a way to reuse information about previously explored solutions, as with the pruning decisions in a branch-and-bound tree, and adapt them to the modified constraints and objectives.

(2) **Online exploration of the design space:**
Manual exploration is essential for stakeholders' understanding of the design problem and speculation such as "what if we group all math characters on the right side of the keyboard?" General-purpose solvers lack interfaces for manually exploring the design space.[9, 15] A two-way interface is needed that lets stakeholders change solutions or propose new ones and enables the optimization process to communicate the outcome from the assessment in human-readable format.

(3) **Learning and visualizing hidden "subjective" criteria:**
Our stakeholders made manual adjustments to a proposed solution, applying tacit criteria such as assumptions about users' habits, cultural specificities, subjective preferences, and political agendas. Optimizers should offer interfaces for making such local changes. From the interactions, a "subjective function" could be learned that can be shown and used as an additional objective in optimization of future solutions.

(4) **Justifications for design choices:**
When presented with a solution, the committee and the public often asked questions of the form "why is this character placed here?" and wanted to understand how a change in the objective weight or parameters would affect the optimal solution. Developing effective visualizations that show how changes to optimization parameters impact the design and *vice versa* could aid users in navigating the design space and make the optimization more predictable.

lieve that much of the potential of computational methods remains un-exploited. The power of algorithms lies in their problem-solving capability. They can explore design spaces and obtain suggestions that would be hard to find by intuition or trial-and-error. This element is often missing from present-day mainstream interaction design, which leaves the generation of new designs to humans.

However, the case of the French keyboard has revealed important challenges in integrating computational methods into large-scale multi-stakeholder design projects. Starting from a well-defined optimization problem, our approach evolved toward something one could call *participatory optimization*. This is inspired by participatory design, which originated with labor unions and was developed as a co-design method aimed at democratic inclusion of stakeholders.[23] Equal representation and resolving conflicts were two key aims. For such optimization, the stakeholders must be brought together at a level where they can inform and influence each other interactively and iteratively, engaging directly with the optimizer and model to arrive at a good solution collaboratively.

There is growing interest in optimization research employing methods that actively include the user in the process. However, the notion of participatory optimization goes beyond previous efforts to simply open up the search- and model-building process for input by the end-user.[18] It focuses particularly on including stakeholders at every step in the process, for which state-of-the-art optimization methods provide limited support. The case of the French keyboard reveals 4 avenues for future work as especially important to address for enabling active participation of stakeholders *and* optimizer in an iterative human-centered design process supported by computational methods (see Table 3).

We envision such demonstrations as ours encouraging establishment of new, human-centered objectives in algorithm research. Considering interactive and participatory properties of algorithms also opens new questions and paths to new, societally important uses. How well can we stop, refine, and resume an algorithm? Can we define

task instances in different ways and leave some variables open? Can we visualize the search landscape meaningfully, or learn "subjective functions" from interactions? Can we use fast approximations in lieu of full-fledged solvers in interactive design sessions? We believe that when designed from a participatory perspective, algorithms could more directly support not only problem-solving but also considering multiple perspectives, making refinements, and learning about a problem.

The code and data presented in this article are documented and open-sourced,[e] alongside instructions for optimizing a layout for any language. ⓒ

e See http://norme-AZERTY.fr.

References
1. Bertsimas, D., Tsitsiklis, J. *Introduction to Linear Optimization*, 1st edn. Athena Scientific, Belmont, MA, USA, 1997. ISBN 1886529191.
2. Burkard, R.E., Offermann, D.M.J. Entwurf von Schreibmaschinentastaturen mittels quadratischer Zuordnungsprobleme. *Zeitschrift für Opt. Res. 21*, 4 (1977), B121–B132.
3. Burkard, R.E., Karisch, S.E., Rendl, F. Qaplib - a quadratic assignment problemlibrary. *J. Global Optim. 10*, 4 (June 1997), 391–403. ISSN 0925-5001. doi: 10.1023/A:1008293323270. http://dx.doi.org/10.1023/A:1008293323270.
4. DGLFLF. *Rapport au Parlement sur l'emploi de la langue française*. Government Report, 2015. http://www.culture.gouv.fr/Thematiques/Langue-francaise-et-langues-de-France/La-DGLFLF/Nos-priorites/Rapport-au-Parlement-sur-l-emploi-de-la-langue-francaise-2015. From the Délégation générale à la langue française et aux langues de France du Ministère de la Culture et de la Communication. In French.
5. DGLFLF. *Vers une norme française pour les claviers informatiques*. Government Publication, 2016. http://www.culture.gouv.fr/Thematiques/Langue-francaise-et-langues-de-France/Politiques-de-la-langue/Langues-et-numerique/Les-technologies-de-la-langue-et-la-normalisation/Vers-une-norme-francaise-pour-les-claviers-informatiques. From the Délégation générale à la langue française et aux langues de France du Ministère de la Culture et de la Communication. In French.
6. Feit, A.M. Assignment problems for optimizing text input. PhD thesis, Aalto University, 2018. http://urn.fi/URN:ISBN:978-952-60-8016-1.
7. Feit, A.M., Weir, D., Oulasvirta, A. How we type: Movement strategies and performance in everyday typing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16, 2016). ACM, New York, NY, USA, 4262–4273. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858233. http://doi.acm.org/10.1145/2858036.2858233.
8. Feit, A.M., Nancel, M., Weir, D., Bailly, G., John, M., Karrenbauer, A., Oulasvirta, A. *Élaboration de la disposition AZERTY modernisée*. Technical report, Inria, Lille, France 2018. https://hal.inria.fr/hal-01826476.
9. Fisher, M.L., Rosenwein, M.B. An interactive optimization system for bulk-cargo ship scheduling. *Nav. Res. Logist. 36*, 1 (1989), 27–42.
10. Frieze, A.M., Yadegar, J. On the quadratic assignment problem. *Discrete Appl. Math. 5*, 1 (1983), 89–98. ISSN 0166-218X. doi: http://dx.doi.org/10.1016/0166-218X(83)90018-5. http://www.sciencedirect.com/science/article/pii/0166218X83900185.
11. Gajos, K., Weld, D.S. Preference elicitation for interface optimization. In *Proceedings of the 18th annual ACM symposium on User interface software and technology* (2005), ACM, New York, NY, USA, 173–182.
12. ISO/IEC 9995-1. *ISO/IEC 9995-1:2009 Information technology – Keyboard layouts for text and office systems – Part 1: General principles governing keyboard layouts*. Standard, International Organization for Standardization, Geneva, CH, October 2009.
13. John, M., Karrenbauer, A. Dynamic sparsification for quadratic assignment problems. In *Mathematical Optimization Theory and Operations* Research. M. Khachay, Y. Kochetov, P. Pardalos, eds. Springer International Publishing, Cham, 2019, 232–246. ISBN 978-3-030-22629-9.
14. Jokinen, J.P.P., Sarcar, S., Oulasvirta, A., Silpasuwanchai, C., Wang, Z., Ren, X. Modelling learning of new keyboard layouts. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17, 2017), ACM Press, New York, New York, USA, 4203–4215. ISBN 9781450346559. doi: 10.1145/3025453.3025580. http://dl.acm.org/citation.cfm?doid=3025453.3025580.
15. Kapoor, A., Lee, B., Tan, D., Horvitz, E. Interactive optimization for steering machine classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10, 2010), ACM, New York, NY, USA, 1343–1352. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753529. http://doi.acm.org/10.1145/1753326.1753529.
16. Karrenbauer, A., Oulasvirta, A. Improvements to keyboard optimization with integer programming. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (UIST '14, 2014), ACM, New York, NY, USA, 621–626. ISBN 978-1-4503-3069-5. doi: 10.1145/2642918.2647382. http://doi.acm.org/10.1145/2642918.2647382.
17. Lee, P.U-J., Zhai, S. Top-down learning strategies: can they facilitate stylus keyboard learning? *Int. J. Human-Computer Stud. 60*, 5-6 (May 2004), 585–598. ISSN 1071-5819. doi: 10.1016/J.IJHCS.2003.10.009. http://www.sciencedirect.com/science/article/pii/S1071581903001794.
18. Meignan, D., Knust, S., Frayret, J-M., Pesant, G., Gaud, N. A review and taxonomy of interactive optimization methods in operations research. *ACM Trans. Interact. Intell. Syst.*, 5(3):17:1–17:43, September 2015. ISSN 2160-6455. doi: 10.1145/2808234. URL http://doi.acm.org/10.1145/2808234.
19. NF Z 71-300. *User interfaces - French keyboard layouts for office*. Standard, AFNOR, 03, 2019a.
20. NF Z 71-300. *Interfaces utilisateurs – Dispositions de clavier bureautique français*. Standard, AFNOR, 03, 2019b.
21. Parkvall, M. Världens 100 största språk 2007. *Nationalencyklopedin*, 2007. http://www.ne.se/.
22. Queyranne, M. Performance ratio of polynomial heuristics for triangle inequality quadratic assignment problems. *Oper. Res. Lett. 4*, 5 (1986), 231–234. ISSN 0167-6377. doi: 10.1016/0167-6377(86)90007-6. http://www.sciencedirect.com/science/article/pii/0167637786900076.
23. Simonsen, J., Robertson, T. *Routledge International Handbook of Participatory Design*. Routledge, New York, NY, USA, 2012.
24. Yassi, A. Repetitive strain injuries. *The Lancet 349*, 9056 (March 1997), 943–947. doi: 10.1016/S0140-6736(96)07221-2. http://www.sciencedirect.com/science/article/pii/S0140673696072212. Manuscript submitted to ACM

Anna Maria Feit (feit@cs.uni-saarland.de) is a professor at Saarland University, Germany. This work was done while a researcher at Aalto University and ETH Zurich, Switzerland.

Mathieu Nancel is a research scientist in the Loki research group at Inria Lille–Nord Europe; Lille, France.

Maximilian John is a researcher at Max Planck Institute for Informatics, Saarbrücken, Germany.

Andreas Karrenbauer is a senior researcher at Max Planck Institute for Informatics. Saarbrücken, Germany

Daryl Weir is a researcher at Aalto University. Espoo, Finland.

Antti Oulasvirta is an associate professor at Aalto University, Espoo, Finland.

Watch the authors discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/azerty-ameliore