ABSTRACT
To cope with the wicked problems and exploit the opportunities of the digital age requires socio-technical environments that integrate social and technical systems. The framework presented in the paper explores architectural principles that makes socio-technical environments more usable, more useful, more engaging and contributes to the quality of life of all their users. The framework focuses on two basic principles by differentiating between (1) adaptive systems that change their behavior by themselves driven by context-aware mechanisms including models of their users and specific task contexts, and (2) adaptable systems that can be adjusted, modified, and extended by their users in order to capture unforeseen and important aspects of problems.

Grounded in specific examples the need for the two different approaches is analyzed, the design trade-offs between them are articulated and specific system developments are documented that we have developed over the last two decades to address specific challenges. Arguments and examples for creating a desirable symbiosis between adaptive and adaptable systems are described and the impact of these developments for socio-technical environments is discussed.

Keywords
adaptive systems, adaptable systems, context-aware interactions, intelligent interfaces, user modeling, task modeling, collaborative filtering, domain-oriented design environments, meta-design, information overload, information delivery

1 Introduction
Research efforts to argue for the need and desirability of socio-technical environments that support users of all ages from a broad spectrum to think, learn, work, and collaborate in more productive and more creative ways have been the objective of numerous research disciplines including computer science, cognitive science, and the learning sciences. More specifically, the research has been pursued in areas such as human-computer interaction [1], artificial intelligence [2], user modeling [3], computer-supported cooperative work [4], and computer supported collaborative learning [5]. Many of these research efforts have not clearly differentiated between adaptable and adaptive systems.

Grounded in the assumption that such a differentiation is important and useful, the objective of this paper identifies the design trade-offs [6] between them, demonstrates the possibility for a successful integration [7], and analyzes the impact of these developments [8]. Throughout the paper, examples are described that illustrating the need for adaptive and adaptable systems in specific contexts that have guided a variety of research efforts over the last two decades.

2 Brief Characterization of Adaptive and Adaptable Systems
Adaptive systems change their behavior by themselves driven by context-aware mechanisms [9] including models of their users and of specific tasks. Adaptive systems components are important because the “typical” user of a system does not exist; there are many different kinds of users, and the requirements of an individual user usually change with experience [10]. Simple classification schemes based on stereotypes, such as novice, intermediate, or expert users, are inadequate for complex knowledge-based systems because they do not take particular contexts into account. One of the central objectives of user modeling in HCI [3] is to address the problem that systems will be unable to interact with users cooperatively unless they have some means of finding out what individual users really know, do and intend. Techniques to achieve that systems know important aspects of users and tasks are: (1) being told by the users (e.g., by questionnaires, setting preferences, or specification components), and (2) being able to infer it from the user’s actions (e.g., by analyzing usage data).

A simple example of an adaptive system is the “Auto-Correct” feature of Microsoft Word. Auto-Correct does a lot of things automatically, and it does them by default. In case a user types “hte” or “EHR” into a Word document (see Figure 1), the system will automatically change the words to “the” and “HER” (in many cases users do not even want to be aware of these changes). The underlying knowledge for these modifications includes (1) a task model of correct spelling, (2) the recognition that “hte” or “ehr” are no English words, and (3) the fact that transposition errors are mistakes that people make frequently.
This letter is written to the National Science Foundation Agency EHR.

Encountering a great variety of related phenomena with intelligent support systems, we have developed a taxonomy to differentiate between adaptive and adaptable systems that is summarized in Table 1 and elaborated in the remaining sections of the paper.

3 Adaptive Systems

This section will describe some of more prevalent challenges, opportunities and pitfalls associated with the development and use of adaptive systems.

3.1 Benefit: Reducing Information Overload with Personalization

The growth of digital technologies has provided the foundations for more information being available at our fingertips and more opportunities for learning and collaboration for all citizens in the digital age. The exploding amount of information available is due to the dramatically increased possibilities for many more people to create and easily share information [13].

The mismatch between these developments has caused and contributed to information, participation, and collaboration overload problems.

Figure 3 illustrates that information production has increased dramatically (represented by the red curve) whereas people’s capabilities (represented by the blue curve) are limited in keeping up with the demands of perceiving, sense-making, organizing, utilizing, and managing all this information [1].

Personalization [14] supported by adaptive system components (represented by the green curve) is seen as the major development to address the fundamental issue that the scarce resource in the digital age is not information but human attention [15]. Simon [16] illustrates with convincing examples that design representations suitable for a world in which the scarce factor is information may be exactly the wrong ones for a world in which the scarce factor is attention. The movie “The Social Dilemma” (https://www.thesocialdilemma.com) provides evidence how many social media companies succeed by capturing as much of human attention as they can, then selling that attention to the highest bidders.
Personalization techniques are widely used in major technological developments; some prominent examples are:

- **Tesla cars** support individual driver profiles, remembers their preferences, and support dynamic personalization by recognizing occupants with interior-facing cameras ([https://www.forbes.com/sites/blakemorgan/2021/05/10/3-ways-tesla-creates-a-personalized-customer-experience/?sh=627b531923b3](https://www.forbes.com/sites/blakemorgan/2021/05/10/3-ways-tesla-creates-a-personalized-customer-experience/?sh=627b531923b3));
- **voice assistants** (such as Amazon’s Alexa) can distinguish and recognize speakers sharing an account ([https://www.theguardian.com/technology/2019/oct/09/alexa-are-you-invading-my-privacy-the-dark-side-of-our-voice-assistants](https://www.theguardian.com/technology/2019/oct/09/alexa-are-you-invading-my-privacy-the-dark-side-of-our-voice-assistants));
- the **Epic Mix app** ([https://www.epicpass.com/benefits/epicmix](https://www.epicpass.com/benefits/epicmix)) provides skiers with substantial amount of information for 34 ski areas owned by Vail Associates including current snowfall, waiting time for lift lines, grooming information and total vertical feet and distance skied;
- **Recommender systems** [17] assisting customer buying books or selecting movies of special interest for them. The basic idea of these systems is to present a selection of items to users which corresponds closely to their specific interests. The collected data is based on items users have bought, recently viewed, and rated, and suggestions of interesting collaborations as identified by “big data” analyses [18]. An indication of the importance of creating better predictions (corresponding to lowering the red to the green curve in Figure 3) is documented by Netflix offering a $1 Million prize for the best collaborative filtering algorithm [19] (which it ended up not using [https://www.wired.com/2012/04/netflix-prize-costs/](https://www.wired.com/2012/04/netflix-prize-costs)).
- **Collaborative filtering** [19] filtering or evaluating items using the opinions of other people. To maximize human attention, the technique computes predictions about the interests of users by collecting preferences, taste, or decision information from many users. Collaborative filtering keeps people in the loop by analyzing the information and condense their opinions into ratings that can be easily processed by software.
- **Intelligent Tutoring Systems** [20] have employed adaptive components by dynamically adjusting the level or type of course content based on an individual student's abilities contained in user models. They identify zones of learnability [21] by determining the gap between what a student knows and what a student is supposed to learn (e.g. according to defined learning goals) by moving students through a personalized learning path to prescribed learning outcomes and skill mastery.

### Table 1: A Comparison and Differentiation between Adaptive and Adaptable Systems

<table>
<thead>
<tr>
<th></th>
<th>Adaptive Systems</th>
<th>Adaptable Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Definition</strong></td>
<td>modifications and suggestions generate by the systems for specific tasks and users</td>
<td>users actively change the functionality of the system</td>
</tr>
<tr>
<td><strong>Knowledge</strong></td>
<td>contained in the system; projected in different ways</td>
<td>knowledge is curated, modified, and extended by users</td>
</tr>
<tr>
<td><strong>Strengths</strong></td>
<td>little (or no) effort by users; no special user knowledge is required; work for people</td>
<td>users are in control; users know their tasks best; work with people</td>
</tr>
<tr>
<td><strong>Weaknesses</strong></td>
<td>users lack control; common understanding is reduced resulting in filters bubbles; lack of explainability</td>
<td>users must do substantial work; require a learning effort; create a tool mastery burden; systems may become incompatible</td>
</tr>
<tr>
<td><strong>Mechanisms required</strong></td>
<td>models of users, tasks, and dialogs; big data resources; intelligent agents</td>
<td>meta-design environments supporting modifiability, tailorability, and evolution</td>
</tr>
<tr>
<td><strong>Application domains</strong></td>
<td>active help systems, critiquing systems, recommender systems</td>
<td>open systems, co-designed systems, end-user development</td>
</tr>
<tr>
<td><strong>Primary Techniques</strong></td>
<td>automation grounded in Artificial Intelligence (AI) approaches</td>
<td>human involvement grounded in Intelligence Augmentation (IA) approaches</td>
</tr>
</tbody>
</table>

3.2 Benefit: Becoming Aware of Unknown Things with Information Delivery

*Information access* (based on “pull” approaches) and *information delivery* (based on “push” approaches) are two
approaches for obtaining information [22]. Information access relies on user-initiated searches, while information delivery is a system-initiated presentation of information that is intended to be relevant to the user’s task. Information access schemes help the designer articulate information needs, while information delivery schemes infer information needs. Support for information access is indispensable since designers must have support to search for information when they perceive the need. Information delivery is a complementary approach that is particularly important when the designer is not motivated to look for information or is not aware of the existence of relevant information.

The fundamental challenge for information delivery systems is not to provide more information “to anyone, at any time, and from anywhere,” but to keep them quiet most of the time by exploiting context-aware mechanisms for identifying “the ‘right’ information, at the ‘right’ time, in the ‘right’ place, in the ‘right’ way, to the ‘right’ person.”[9].

Research has explored the opportunities and developed support mechanisms for adaptive system components in numerous domains including active help systems for high-functionality environments, large software reuse libraries, and domain-oriented design environments [23].

These research activities resulted in the generic architecture shown in Figure 4 that can be widely employed for adaptive system components to reduce the intrusiveness of information delivery systems.

![Figure 4: Computing user- and task-relevant for information delivery systems](image)

The different domains can be briefly explained:

- The ovals L1, L2, and L3 represent the models held by the user: L1 is the domain of functionality that the user uses frequently (therefore being familiar to the user); L2 is the domain of functionality that the user uses occasionally (passive help systems can provide adequate support for this domain); and L3 is the mental model of the system held by the user. The rectangle L4 represents the actual system.
- The diagram indicates two areas of particular interest: (1) L3 ∧ ¬L4, representing functionality that a user assumes exists in the system, but it does not; and (2) L4 ∧ ¬L3, representing functionality existing in the system that the user has no awareness of.
- The cloud represents the information needed for the inferred task at hand (shown with fuzzy boundaries because the system may have only an incomplete understanding of it).

Identifying these domains improves adaptive systems as follows: (1) the black dots are not relevant for the task at hand and should therefore not be delivered, (2) the white dots inside the cloud are already known and should also not be delivered; and (3) the shaded dots are task-relevant and not known and therefore should be delivered to the user.

### 3.3 Drawback: Lack of Meaningful Explanations

Many current systems (specifically AI systems with adaptive components [8]) are unable to provide understandable explanations about their behavior. The inner workings of such systems are “black boxes”: they provide recommendations, but they are unable to explain the underlying rationale needed for empowering users to adapt systems to their needs. An example (encountered a couple of years ago by the author) occurred in the context of using the navigation system of a rental car. The fact that it was a rental car limited our familiarity with the functionality of the navigation system. We programmed the system to guide us the quickest way to our destination. Continuing our trip with the navigation system in operation, it advised us at every exit to leave the freeway. Without any way to query our navigation system and because we were familiar with the geography, we decided not to follow the advice that the system repeated at every exit. Unable to explore the differences between the system and us for choosing the most
preference route, we followed our plan and shut the system
off.

The navigation system acting as an adaptive system was
unable to describe its intention in a way that we were able to
understand. These systems are not capable of meaningful
explanations. Drivers do not have any way to query these
systems and they cannot adopt our perspective to determine
what statement would satisfy us. They cannot convey
confidence in the route they have selected, other than giving
a probabilistic estimate of the time differential for alternative
routes, whereas we wanted them to explain the assumptions
they are making.

When we returned the car and described the encountered
problem the human rental car agent provided us with an
explanation: somewhere deep down in a complex web of
options there was a flag “do not use freeways” — and the
previous renter had selected this option as part of the
adaptable features. An intelligent navigation system would
have allowed us to ask the question “why should we leave
the freeway” and then guided us to find the flag that was
responsible for this behavior.

3.4 Drawback: Polarization based on Lack of
Shared Experiences and Common
Understanding

Reducing the information overload (by reducing the amount
of information from the red to the green curve in Figure 3
and thereby only delivering information represented by the
shaded dots in Figure 4) is highly desirable but it represents
a design trade-off [6] that needs to be taken into account, and
the pros and cons need to be carefully evaluated. One of
the things that binds a culture together is that people are exposed
to the same relevant information presented from different
points of view. One of the pitfalls of adaptive systems and
user models is that they can and will create filter bubbles,
echo chambers [14]. The polarization based on individual
universes of information has become a defining
characteristic in our world of today. Grounded in interests
captured by big data, citizens will be grouped into types and
be confronted with information that corresponds to their
preferences and conforms to their beliefs leading to group
think [24]. There are numerous prominent and far-reaching
elements including:

- people in the USA watching CNN or Fox News
  will form beliefs and act accordingly as they would live
  in two different worlds;
- the Corona virus has split societies around the
  world in totally opposed camps believing in vaccination
  or totally opposing it.

To address global problems such as Corona, climate change,
digitalization, fake news, and the widening economic divide
require the efforts of large communities and massive
coordination to create a common understanding based on
shared experience. Groupthink represents a serious obstacle
for a future world in which people make conscious efforts to
create common ground and avoid further polarization.

4 Adaptable Systems

Designers of socio-technical environments face the
formidable task of writing software for millions of users (at
design time) while making it work as if it were designed for
each individual user (at use time). New design
methodologies [25] explore that the once sharp distinction
between users and developers of software is fading away,
and many users are starting to take control of shaping
software with adaptable systems to their needs through their
own development activities. Adaptable systems provide
foundations for “democratizing innovation” as argued by
von Hippel [26]: “Users that innovate can develop exactly
what they want, rather than relying on manufacturers to act
as their (often very imperfect) agents. Moreover, individual
users do not have to develop everything they need on their
own: they can benefit from innovations developed and freely
shared by others”.

This section will describe some of the more prevalent
challenges, opportunities, and pitfalls associated with the
development and use of adaptive systems.

Adaptive systems analyze existing information and
functionality and allow users actively change and transcend
the functionality of an existing system. Figure 2 illustrates
the adaptable component of Auto-Correct to overwrite the
changes made by the adaptive component. Another simple
but useful adaptable system component is the development
of Macros in MS-Word.

On a global scale, adaptable systems can be supported by:
(1) offering task-specific languages supporting human
problem-domain interaction [1]; (2) provide programming
environments that protect users from low-level
computational drudgery [5]; (3) support customization,
reuse, and redesign effectively [27]; (4) tailor software
applications at use time with component-based approaches
[28]; and (5) advance construction kits to domain-oriented
design environments with intelligent support systems [29].

4.1 Benefit: Creating Supportive Environments
for Adaptable Systems with Meta-Design

Meta-design (“design for designers”) is a theoretical
framework to conceptualize and to cope in unique ways with
design problems. It is focused on open-ended co-design
processes [30] in which all the involved actors actively
participate in different ways. It is grounded in the
fundamental assumption that design is not a matter of getting
rid of the emergent, but rather of including it and making it
an opportunity for more creative and more adequate
solutions to problems by supporting adaptable systems [31].

Research resulted in the following design requirements for
encouraging end-users to engage in adaptable extensions
[32]: (1) making changes must seem possible; (2) changes
must be technically feasible; (3) benefits must be perceived
by the stakeholders who do the work associated with the
adaptations; (4) low barriers must exist for sharing changes;
and (5) the original designers acting as meta-designers must
be willing to share control of how systems will be used,
which content will be contained, and which functionality
will be supported.
4.2 Benefit: Putting the Problem Owners in Charge

A challenge for many software systems is the growing importance of application domain knowledge held by domain experts rather than by software developers, who suffer from a “thin spread of application domain knowledge” [33]. Another challenge is the need for open, evolvable systems that can adjust to fluctuating and conflicting requirements.

A interview with a geoscientist highlights the importance of these challenges that can be addressed by adaptable components [34]. He uses several existing domain-specific software systems to analyze his research data. However, those systems cannot provide complete solutions to his problems as his research unfolds and his understanding of the problem, data, and results progresses. He said,

I spend on average an hour every day developing software for myself to analyze the data I collected because there is not any available software. Even if there is a software developer sitting next to me, it would not be of much help because my needs vary as my research progresses and I cannot clearly explain what I want to do at any moment. Even if the software developer can manage to write a program for me, I will not know if he or she has done it right without looking at the code.

He continued,

So I spent three months to gain enough programming knowledge to get by. Software development has now become an essential task of my research, but I do not consider myself a software developer, and I don’t know many other things about software development.

This example provides evidence for democratizing innovation (see comment from von Hippel [26] earlier in the paper) and that software development is no longer the exclusive activity of professional software engineers. Domain experts being the owners of problems such as this geoscientist are engaged in intensive software development and adaptable systems provide important support environments to address this challenge.

4.3 Drawback: Participation Overload

Adaptable systems open up unique new opportunities for mass collaboration and social production [13], but these engagements are not without drawbacks. One such drawback is that humans may be forced to cope with the burden of being active contributors in personally irrelevant activities leading to a participation overload [35]. “Do-it-yourself” societies empower humans with powerful tools; however, they force them to perform many tasks themselves that were done previously by skilled domain workers serving as agents and intermediaries. Although this shift provides power, freedom, and control to users and customers, it also has urged people to act as contributors in contexts for which they lack the experience that professionals have at their disposal.

4.4 Drawback: The Tension between Standardization and Improvisation

Meta-design creates inherent tensions between standardization and improvisation. The SAP Info (from July 2003; [34]) argues to reduce the number of customer modifications: “every customer modification implies costs because it has to be maintained by the customer. Each time a support package is imported there is a risk that the customer modification may have to be adjusted or re-implemented. To reduce the costs of such on-going maintenance of customer-specific changes, one of the key targets during an up-grade should be to return to the SAP standard wherever this is possible.” Finding the right balance between standardization (which can suppress innovation and creativity) and improvisation (which can lead to a Babel of different and incompatible versions) represents a design trade-off which has been noted as a challenge in open-source environments, in which forking has often led developers in different directions.

5 Symbiosis

Adaptive and adaptable system components represent design objectives to cope with the challenges and opportunities of the digital world. They have to cope with the fundamental property of design [16] that design is choice. It is an argumentative process with no optimal solutions. In design, trade-offs are universal and unavoidable because there are no best solutions and no decontextualized sweet spots independent of specific goals, objectives, and values pursued by specific users for specific tasks. An in-depth analysis of design trade-offs to emphasize an “and” instead of an “or” relationship can provide a symbiosis [7] between the two approaches and should include the following trade-offs:

- **Automation versus augmentation:** adaptive features rely on autonomous features and are grounded in different Artificial Intelligence techniques requiring no human effort but humans have little control whereas adaptable features inform, require human work and engagement whereby providing humans the control over the changes (see Table 1 and [2, 36, 37]). Automation works for people; augmentation works with people.
- **Task- and user-relevant information versus serendipity:** adaptive features can reduce the information overload and bring us closer to people with shared interests, but they can lead to filter bubbles, echo chambers, and privacy violations (see Figure 3 and [14, 38]);
- **too many customized versions versus fitting individuals’ needs and task demands:** adaptable features allow users to be creative by adding new capabilities, but they can lead to incompatible versions of systems and lack of coherent voices (see section 4.4).

These design trade-offs will be used to frame the design possibilities for creating symbiotic relationships. The examples are grounded in real experiences and served as anchor points for a variety of research activities.
5.1 Example: Complementing Generic Critics with Specific Critics

Critiquing is a dialog in which the interjection of a reasoned opinion about a product or an action triggers further reflection on or changes to the artifact being designed. An agent (human or machine) capable of critiquing in this sense is a critic.

Critiquing systems can be especially supportive for design tasks that have the following characteristics: (a) knowledge about the design domain is incomplete and evolving, (b) the problem requirements can be specified only partially, and (c) the necessary design knowledge is distributed among many design participants. Critiquing messages rely on adaptive system components and the human actions in response require support for adaptable system components.

Research over the last two decades has explored different aspects of critiquing systems with a focus on embedding these systems in domain-oriented design environments. Critiquing systems are well suited for design tasks in complex problem domains in which expert systems or automated design approaches have proven inadequate. In representative approaches the human's primary role is to generate and modify solutions; the computer's role is to analyze these solutions and produce a critique for the human to consider in the next iteration of this process. Generic critics reflect knowledge that applies to all designs, such as accepted standards, building codes, and domain knowledge based on physical principles. In a specific domain-oriented design environment for kitchen design, an example of a critiquing message analyzing the design of a user could be “The dishwasher is too far from the sink.” The critiquing message relies on an adaptive system component. These systems integrated a construction with an argumentation component to support “reflection-in-action” as a problem solving approach. The human actions in response require support for adaptable system components including exploring the design rationale behind the critiquing rule and possible modifying the critiquing rule and the associated design rationale.

The complement generic critics with specific critics may be further refined by integrating task-relevant knowledge with characteristics of individual users contained in a user modeling component (see Figure 4 and [3, 42]). The generic rule “the dishwasher should be on the right side of the sink” (because the majority of people is right-handed) should be complemented with the specific rule “If the primary kitchen user is left-handed, the dishwasher should be on the left side of the sink.” Without a specification component the system will not know that the user in a specific design is left-handed. Specific critics increase the personalization to situation-specific characteristics thereby reducing the information overload by computing the most essential aspects of a design (see Figure 3).

Another important adaptable system component (supported by interface concepts such as sliders) allowed users to control the intrusiveness of the adaptive component of the critiquing system. The situation-specific rules ranging from “do not critique unless explicitly requested” to “critique after every action” represent another design trade-off.

5.2 Example: The Importance of Local Information Environments in a Globalized World

Massive Open Online Courses (MOOCs) are (partially) free courses with massive enrollments that promise education for everyone and for all interests. A basic aspect of MOOCs is that they can be taken by students all over the world. The need for adaptive and adaptable components arise from the different cultural backgrounds of students. For example: courses taught by instructors in the USA might use statistics from Baseball that are familiar to people in the USA whereas in other parts of the world statistics from soccer would be much more meaningful.

Another culturally dependent consideration is the gas consumption of cars (being an important factor for climate change). In the USA it is measured in gallons and miles whereas in other parts of the world people use liters and kilometers. This example illustrates an additional difference beyond the measurement difference of gallons/miles versus liters/km: the gas consumption of cars is measured in the USA by how many miles a car can go with one gallon (fixed amount of gas and variable distance) whereas in Europe it is measured by the numbers of liters for 100km (variable amount of gas and fixed distance).

To make global MOOCs fit local contexts can be addressed by several adaptive and adaptable components. Some of the modifications can be done automatically with adaptive components (using the location information of specific users) whereas other modifications could be localized by teachers (who share the same cultural backgrounds with the learner).

5.3 Example: Self-driving Cars versus Intelligent Driver Support Systems

A deep understanding of the respective contributions of adaptive and adaptable components will contribute to analyze the design trade-offs between self-driving cars and intelligent driver support systems. Adaptive system components will play a prominent role in supporting self-driving cars (taking control away from the human driver) whereas adaptable components are critical for intelligent driver support systems. The two approaches are already successfully combined in the recent generation of cars. An example for a symbiosis is dynamic cruise control where (1) adaptive components keep the car at a certain speed, make it slow down if an obstacle is discovered, and automatically resume speed to the original setting if the obstacle is out of the way and (2) adaptable components allow drivers to set the speed and the distance to the car in front of them that they consider personally safe.

6 Design Guidelines

Adaptive and adaptable system will be important components of future socio-technical environments. This
section summarizes design guidelines grounded in the frameworks and examples discussed in the paper.

**Identify the design trade-offs associated with different approaches** including: In design there are no decontextualized sweet spots. To understand the benefits and pitfalls of adaptive and adaptable systems their uses must be situated and explored in specific contexts (see Table 1).

**Strengthen the Benefits of Adaptive System** including: reduce information overload, deliver unknown functionality and information, support personalization to focus attention, complement the tool approach with intelligent agents (see Figure 3 and section 3.1 and 3.2).

**Strengthen the Benefits of Adaptable System** including: create sociotechnical environments that empower domain experts to engage actively in the continuous development of systems rather than restricting them to using existing systems; support meta-design at design time for creating solution spaces in which users can create their own solutions to fit their needs at use time (see section 4.3 and 4.4).

**Be aware of the Drawbacks of Adaptive System** including: being enclosed in filter bubbles and echo chambers; lack of explainability of the algorithmic decisions; privacy intrusions (see section 3.3 and 3.4).

**Be aware of the Drawbacks of Adaptable System** including: participation overload; incompatible version of systems; lack of rewards and recognition of contributions (see section 4.3 and section 4.4).

**Give Humans Control over Technology** including: identify the right mix between computer-based automation (replacing human beings) and human control (empowering human beings); strike a balance between relevance and serendipity (see Figure 4).

**Explore the Opportunities for Creating Symbiotic Relationships between Adaptive and Adaptable Systems** including: adaptive systems analyze what exists and they can provide foundations how new functionality can be added with adaptable components (see Figure 1 and Figure 2 and section 5); support user-controlled adaptation with system-generated adaptive suggestions.

7 Conclusions

To enrich rather than limit human lives with adaptive and adaptable technologies, discourses and investigations must not only be focused on technological issues but explore motivation, control, ownership, autonomy, and quality of life. A better understanding is required whether the technologies of the future (1) will provide us with more time, less stress, more control, and enhance human creativity or (2) will they cause a shift in authority from humans to algorithms (especially in case of tools that we do not understand and that cannot provide us with explanations about their actions).

Changes in complex environments are not primarily dictated by technology, but they are the result of an incremental shift in human behavior and social organization. They require the co-design of social and technical systems and need to use models and concepts that focus not only on the artifact but exploit the social context in which the systems will be used. Giving users more control to evolve systems with adaptable system components may lead to a cultural transformation towards democratizing innovations and life in general. The paper documents frameworks and illustrates with real world examples identified by empirical research how to differentiate and integrate adaptive and adaptable systems components in socio-technical environments towards achieving some of these objectives.

ACKNOWLEDGMENTS

References


44. Fischer, G., Beyond Hype and Underestimation: Identifying Research Challenges for the Future of MOOCs. Distance Education Journal (Commentary for a Special Issue “MOOCs: Emerging Research”), 2014. 35(2): p. 149-158.