

Framing Design for Appropriation with Zones of Proximal Evolution: email for PIM

Link to the formal publication: <https://doi.org/10.1016/j.ijhcs.2018.11.004>

Citation:

Tchounikine P. (2019). Framing Design for Appropriation with Zones of Proximal Evolution: Email for PIM. *International Journal of Human-Computer Studies* 123:18–28.

© 2019. This manuscript version is made available under the CC-BY-NC-ND 4.0 license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Abstract

Software appropriation is the process by which users adopt applications and make them their own. It is an important concern for HCI because it impacts what users use software for and/or how they use it, which in turn may cause applications to be unsuitable mediators for users' effective activity. We address appropriation as the process by which users, while interacting with the effective tasks at hand, attribute functional values to software artifacts. Taking as an illustration the way that email clients are appropriated as Personal Information Management (PIM) devices, we study why users should be offered adaptation means; how to identify adaptation means that are likely to allow users to solve their problems themselves; and how to address users' scaffolding.

1. Introduction

Software appropriation is the process by which users adopt software and make it their own (Dourish, 2003; Mackay, 1990). It is an important concern for HCI because it impacts what users use software for, and/or how they use it. As a consequence, applications may be unsuitable mediators for users' effective activity, causing them to be abandoned and/or create breakdowns (Bødker and Klokmoose, 2011). A typical example is how email applications (also referred to as email clients) are used as Personal Information Management (PIM) devices, e.g. to remember tasks. A large set of literature (see (Whittaker et al., 2006) for an early summary) shows how, since email clients are not meant for such purposes, users develop unexpected uses and workarounds, and also face breakdowns.

All authors examining appropriation agree that users must be offered adaptation means (Kaptelinin and Bannon, 2012; Mørch, 2011; Dix, 2007; Dourish, 2003; Mackay, 1990). Appropriation develops from effective usage, hence between software releases. Participative and iterative UCD are crucial, but not sufficient. Users must be offered means to adapt applications to their effective usages themselves.

This article studies the following research question: how can designing for appropriation be

supported, i.e. how can designers be helped to reflect on users' appropriation processes and offer adaptation means that may enable users to adapt the system to the effective usages they develop?

The contribution we propose is based on a theoretical account of software appropriation that adopts a developmental perspective: appropriation is addressed as the way users, while interacting with tasks, attribute functional values to artifacts and turn them into instruments for themselves (Tchounikine, 2017). Furthering this perspective, we introduce two notions to help designers consider adaptation issues: (1) instrumental direction, which focuses on the particular direction in which usage develops, and (2) Zone of Proximal Evolution, which focuses on the zone within which users may act productively.

This research question and this theoretical contribution are addressed and presented by studying how email clients are appropriated to remember tasks, and how designers may be helped to take such processes in account. The specific interest of this case study is that it builds on a review of a large amount of empirical analyses, conducted over a long period (three decades). Rather than considering recent technologies and hypothesizing how their usage will evolve, and in which context, considering email makes it possible to build on evidence. Moreover, although use of email (as such and for PIM) may seem outdated, it is worthwhile to remember that usages continue (Müller et al., 2015). Email is, and will likely continue to be, used, either alone and/or in conjunction with other more recent technologies; this is directly related to appropriation processes (see *infra*). Finally, studying email also helps us understand and anticipate more recent technology usages.

The article is organized as follows. In Section 2, we summarize the theoretical background we build on. In Section 3, we study how users take advantage of the general and adaptation features of email clients to build instruments. In Section 4, we study how design may respond to this, and highlight the specific interest of a designing-for-appropriation perspective to address inter- and intra-user variability. In Section 5, we introduce the notion of instrumental direction and show how it provides a framework to identify potentially relevant adaptation means. In Section 6, we introduce the notion of Zone of Proximal Evolution and show how it further frames design and scaffolding issues. This is illustrated with plugin and meta-plugin examples. Finally, Section 7 provides additional information and discussions.

2. General theoretical background

As raised by Dix (2007) designing for appropriation, i.e. intentionally providing the means to enable users to adapt software applications to the effective usages they develop, may be considered an oxymoron ("planning for the unexpected"). However, although usages are not fully predictable, we believe that designers may be prepared for action by a theoretical model of appropriation phenomena, productive conceptual notions, and a strategy framework.

The theoretical framework proposed in (Tchounikine, 2017) may be summarized as follows.

In line with Activity Theory, software applications may be regarded as *artifacts* which are mobilized by users in the context of the tasks at hand (Kaptelinin and Nardi, 2006; Bødker and Klokose, 2011). These artifacts become *instruments* for users, in the context of these activities, when they allow users to achieve these tasks. The development of instruments by users is a dual process comprising users' adaptation to the artifact constraints and transformation of the artifact by users (Rabardel and Waern 2003). This process is related to users' invariant organizations of behavior for a certain class of situations (schemes), which combine technical and conceptual aspects. It is also related to the evolution of these invariants, since users develop stable ways of addressing tasks and mobilize them as such, yet these stable ways do evolve.

Within this general perspective, software appropriation may be defined as the process by which users, while interacting with the effective tasks at hand, attribute *functional values* to software artifacts and, by considering them as mediators of their activities, turn them into instruments for themselves (Tchounikine, 2017). The notion of functional value of an artifact for a user (or group of users) is defined as the utility of the artifact for achieving a given task or goal *as perceived* by the user. A software application as such does not have a functional value. A user attributes one or several

functional values to it when and as he/she perceives it as a means to achieve the effective tasks that are undertaken. From a methodological perspective, usage analyses in terms of functional values may be conducted using different tools from which analysis methodologies originating in Activity Theory or in the genre approach; see a review in (Tchounikine, 2017).

Appropriation is related to the ontological characteristic of human beings to accomplish tasks and, dialectically, to elaborate means to do so. It may relate to users undertaking other tasks than the ones for which the application is designed, or undertaking them differently. In terms of analysis, emphasis should thus be on the way users interact with tasks rather than the way they interact with software applications. Tensions between the instruments developed by users and the existing features of software applications may natively exist, in which case there is an initial requirement analysis or design issue. However whatever the design, good or poor, tensions may also arise in relation to user development. Different users may turn the same applications into different instruments. Additionally, features at one time appropriate for a particular user may, later on, only partially support this user and/or become obstacles.

Taking this understanding, designing for appropriation should be thought of as allowing users to solve their instrument-artifact tensions themselves. Adaptation of software by users is both an indicator of, and a means for, software appropriation. Users adopt software and, as a part of this process, take advantage of adaptation features to render applications convenient given the functional values they attribute to them. Users should thus be offered adaptation means to help them align the software applications they use with their developed or developing instruments.

This theoretical framework provides a conceptualization of appropriation. However, it lacks support for designers to reflect on users' appropriation processes and offer pertinent adaptation means. In the next sections we show how designers may be helped to address these issues.

3. An analysis of why and how users take advantage of email client features

In this section, we present and discuss different works related to email for PIM. Let us mention that, in order to keep the article streamlined, we focus on the use of email. However, a proper analysis of how users interact with tasks such as PIM would require a holistic view. Users act within an ecology of artifacts, and appropriation of an application such as email is influenced by the usage of other devices and/or applications. We will come back to this in the discussion section.

3.1. Usage of email clients for PIM

Although email clients are designed to exchange messages with other people, users also send emails to themselves (emails-to-self). This practice is far from anecdotal. A recent study conducted in a large technology corporation found that 92% of users sent emails-to-self (Bota et al., 2017).

Why do users send emails-to-self? Since pioneering work in the 1980's, it has been known that email acts as more than just a communication system (Mackay, 1988). In particular, email clients act as PIM devices for task management, e.g. providing reminders for current tasks, tracking task status or maintaining information relevant to those tasks (Whittaker et al., 2006, Haraty et al., 2016). Emails-to-self occur within PIM usages of email: the inbox is used as a to-do list.

Let us frame emails-to-self in our theoretical background. Users, in addition to communicating with other people, also consider other goals from which to remember tasks. As many tasks are referred to in emails and/or they always have an eye on the inbox, the emailing application is perceived as helping to remember tasks. It is progressively attributed a *to-do* functional value, which is in turn reinforced by its success. This leads users to ask correspondents to send an email "to remind me that I should (...)", and/or to send emails-to-self. In Bota et al.'s (2017) analysis, 53% of emails-to-self concern reminders or to-do lists. Keeping emails from correspondents or from self as task reminders (let us call them "to-do-emails") is not a recent usage. In Ducheneaut et al.'s (2001) study, 83% of respondents left messages in their inbox as reminders, and 72% sent emails-to-self. Actually, if the email client has been appropriated as a to-do list instrument, it does not only make sense for users to include new important tasks in this list via emails-to-self. It becomes mandatory.

3.2. How email client features enable and/or support appropriation as to-do list instruments

Basically, email clients act as to-do lists because users know that they often have a look at the inbox (Whittaker et al., 2006). However, email client features also play an important role.

A first example is the folder feature. Users develop different strategies such as moving all to-do-emails to a specific folder or, on the contrary, putting all emails in folders except the to-do-emails (Belloti et al., 2005). Users also build from the alphabetical ordering feature to develop workarounds such as using "AAA-my-to-do-list" names (Ducheneaut et al., 2001). Some users manage self-addressed to-do-emails by editing the email and moving it from drafts to the inbox, not even using the basic purpose of the application (sending/receiving emails). Very interestingly, users also change their strategies. For instance, Whittaker's studies (2005) show that a majority of users who set up dedicated to-do folders abandon this strategy because of its cognitive cost.

Other examples are the use of sorting and searching features. Users use associative reminding based on indirect social and temporal cues such as sender, recipient and date of the message (Whittaker et al., 2006). They implement these strategies by taking advantage of the tabular interfaces of email clients and their sorting features, which make it possible to sort emails by characteristics such as date or subject. Searching by criteria makes it possible to consider similar concerns with different means such as considering the full text of an email or searching across folders. Early works showed that sorting was more popular because it is technically and cognitively simpler (Ducheneaut et al., 2001). Comparison within subjects shows there is no clear best strategy: each model has its strengths and weaknesses, including a cognitive cost (Civan et al., 2008). Very interestingly, this study reveals that users have different perceptions of these means.

3.3. How email client features enable and/or support users to solve their problems themselves

Let us now consider some issues raised by the fact that email clients are not meant to be used as to-do lists, and how users take advantage of adaptation means to solve their problems themselves.

An inbox used as a to-do list is polluted by the constant flow of incoming emails and/or emails kept for other reasons such as archiving (Whittaker et al., 2006). Another difficulty is the necessity to prioritize tasks (Belloti et al., 2005). This may lead to breakdowns, in particular because to-do-emails may be kept for long periods. Bota et al.'s (2017) survey shows self-addressed to-do-emails may find a utility for hours (48%), days (31%), weeks (5%), months (10%) or even years (2%). More generally, management of the temporal information of emails is an issue. The relevant information is more likely to be in the email message than reflected by the email date (Singh et al., 2013).

Breakdowns are typical reasons for users to switch to other means, for instance in this case a specific to-do list application. However, switching to other means is cognitively expensive, and not necessarily possible. Therefore, if possible, users often rather develop a new resource to deal with their needs within their currently used technical context.

A first way users manage the mismatch between email date and urgency is to cheat the system. For example, users refresh old emails by forwarding them to themselves or mark them as unread (Whittaker, 2005; Cecchinato et al., 2016). Another strategy is to use a tagging feature. Most email clients propose ready-to-use tags, "to-do" being a classical one. Using a to-do tag and sorting emails by tag lists displays the relevant emails and tasks. Some email clients, however, also make it possible to create personal tags, which act as a secondary notation system. This makes it possible to deal with the date/urgency mismatch by developing as a new resource an idiosyncratic system of semantic labels (e.g. "urgent" and "very urgent", or specific dates). Tag folksonomies (in the original sense, i.e. personal) may also include information of other types such as "to be managed by phone" or "awaiting third-party action". Using tags to find items more easily than in the default time-and-date-based view is a recurrent pattern (Belloti et al., 2005). Some email clients introduce or enforce predefined pervasive semantics (e.g. predefined "categories" which are referred to in other features), which may in turn support or constrain user needs.

As another example, the way email clients create threads by subject may raise issues, and require workarounds. When users reply to an email mentioning a task, the thread (visualized as a tree) well denotes the task-related exchanges. However, if someone sends a "new" email rather than

using "reply", it will not be threaded. Actually, the issue is due to the potential mismatch between the container (an email is a technical entity) and the content (the message text) with respect to viewing topics (Reyes and Tchounikine, 2003). Tags may be used to indicate the task at stake, but are useless to view threads. Cheating the system is not straightforward. Forwarding the not-threaded email to oneself while changing its "subject" with the one of the threaded emails does not work (or, at least, not for all email clients). Actually, cheating the system may require replying to one of the threaded emails, copying in this email the text one wants to attach to the thread, and sending it as an email-to-self. Users may pay this price, or not, depending on the importance of the functional value "task-related exchanges" they have attributed to threads.

3.4. Discussion

Sections 3.2 and 3.3 illustrate how user appropriation develops from, and consists of, users building their own instruments. These instruments originate in the effective needs of users: the tasks at hand; how they interact with these tasks; and how, in this context, they take advantage of software applications and features, which may lead to unexpected usages and workarounds. They illustrate how usages are related to perceptions (typically, if comparative analyses revealed the searching or filing strategy was more efficient, this would not imply its adoption). Additionally they show that design choices may come into conflict with user instruments (for example, some email clients suggest a flat organization of emails and the use of search tools).

A possible perspective is to consider unexpected usages arising from appropriation processes as a guidance and/or training issue only. Typically, email clients are arguably not very efficient in managing to-do lists and, more generally, in PIM, in particular due to fragmentation issues (Whittaker et al., 2006). In light of this perspective, efforts should be made to encourage use of other more convenient applications.

However, this perspective does not necessarily hold. Users do have good reasons to continue using emails for PIM. Email is often more like a habitat than an application, within which workers spend most of their work day (Ducheneaut and Bellotti, 2001). An analysis of present US mobile users reveals that emails come 2nd in the top smartphone activities (after text messages, but before phone calls) and 1st for tablet activities (Müller et al., 2015). A recent Delphi study (a forecasting method that calls on a panel of experts) was conducted to identify which PIM practices should be recommended or discouraged (Jones et al., 2015). "Email yourself notes, thoughts, tasks/to dos" appears as the second most valuable practice for "Information Capture and Retention for Later Use", just behind "Take a picture with smartphone". For "Reminding, Managing Attention, Tasks/To-dos", the option to "Use email inbox as to-do list (including flags as reminders, or maintaining a single "to-do" email)" comes second after "Use calendar events".

Acknowledging usage and helping users manage PIM via their email clients is thus a must.

4. Responding to appropriation by design

Appropriation of email clients as PIM devices may be responded to by design in two non-exclusive ways. The first is to embed in email clients dedicated features. The second, which we advocate, is to offer users adaptation means to solve their problems themselves.

4.1. Embedding PIM features in email clients

Acknowledging usage and helping users manage PIM via their email clients may be addressed by embedding PIM features in email clients, or transforming email clients into proper PIM devices. Research projects have explored different perspectives. As examples, identifying usage patterns and/or breakdown causes (e.g. difficulty viewing threads or information fragmentation) has led to task-centric or project/activity-centric interfaces (Whittaker 2005; Bellotti et al., 2005; Kaptelinin, 2003). Other works suggest sticking to traditional email client interfaces while offering services such as automatic categorization of emails and associated processing (e.g. filing in specific folders or tagging), including attempts for automatic identification of tasks mentioned in emails (Sappelli et al., 2016). As a matter of fact, most if not all standard market email clients offer specific ready-to-use

features to define and manage tasks, events and calendars.

Embedding PIM features in email clients, however, faces the issue of the heterogeneity of usages resulting from appropriation processes. Since very early research, it has been shown that people tend to adopt varied tactics to deal with task management via email (Whittaker et al., 2006; Mackay, 1988). Recent usage studies confirm this usage heterogeneity.

Among other influences, work practices play an important role. Dabbish and Kraut (2006) showed the influence of broad aspects of work such as level of responsibilities, dependence on other individuals' activities, or number of on-going projects. Recent analyses confirm this and point out that situations have become even more complex, for different reasons. Personal email has developed, and users manage personal and work email differently (Grevet et al., 2014). Another factor is that users now use multiple devices (e.g. laptops and smartphones), and develop different workarounds depending on the device (Cecchinato et al., 2016). Depending on their job, users also have different autonomy for selecting the applications they use, tasks are more less structured, and work-home boundaries differ (Haraty et al., 2016; Cecchinato et. al., 2015). A recent qualitative study (Hanrahan et al., 2014) also showed that users develop different strategies in relation to what draws their attention, e.g. anxiety, notification, number of unread emails or, of particular importance, participating in ongoing conversations. This study shows that most of the tasks referred to in emails are accomplished through communication, and individuals' email usages are shaped by a wider context of on-going relationships and activities.

As Belloti et al. (2005) presciently mentioned, *"From a design perspective, and more generally speaking, a user-centered design approach of trying to support every possible tactic may only lead to "feature creep." It is easy for the analyst to get bogged down studying the infinitely variable (and often fascinating) practices of e-mail users (...)." Although categories and usage patterns may be identified, most users fall into a middle ground and use combinations of strategies (Haraty et al., 2016). Categories are useful as analytical lenses, but not necessarily predictive of the adequacy and use of features that are designed based on them. It is not surprising that different qualitative studies show how some users, while using email for PIM and aware of ready-to-use applications or features, do not switch to these means or abandon them (Haraty et al., 2016; Singh et al., 2013).*

4.2. Designing for appropriation as offering users adaptation means

Taking our theoretical perspective, the heterogeneity resulting from the specificities of the socio-technical context and practices of users mentioned in Section 4.1 is reinforced by users' developmental processes. Users continuously create or improve resources allowing them to accomplish their tasks, within a competence improvement process or in response to breakdowns.

Therefore, the issue at stake may be conceptualized as follows. For interrelated collective and idiosyncratic reasons (e.g. personal characteristics, work practice or ecology of artifacts), two dual aspects must be addressed: inter-user variability (different users / different instruments, and thus different technical needs to build efficient idiosyncratic instruments); and intra-user variability (one user / evolving instruments, and thus evolving technical needs).

Ready-to-use features alone cannot respond to inter- and intra-user variability. Whatever the design of a feature to define and remember tasks, it is one response, which may or may not fit exact users' needs, or may do so only at a given developmental stage, in a given context, or for a given task. Moreover, although it may be based on a careful iterative and participative UCD process, it is a top-down response (designers invent something and explain to users how they should use it and thus, implicitly, how they should act and behave).

In direct contrast, the designing for appropriation strategy consists of intentionally providing means that are likely to enable users to adapt the system (here, the email client) to the effective usages they develop. Whether offering ready-to-use PIM features or not, email clients must support users who develop personal strategies by giving specific attention to the adaptation means offered.

Two different non-exclusive strategies may be considered (Tchounikine, 2017). One of these is to continuously adapt or extend current design to support smart emergent usages and/or make them available to other users. This may be seen as both design *for* appropriation (the objective is that users'

effective usages of the application as appropriated are supported) and *from* appropriation (designers build on users' appropriation process outputs). Another strategy is to offer users adaptation means to solve their problems themselves. If, as we suggest, usage analyses guide the identification of the adaptation means to be offered, this may also be viewed as design *from* appropriation, although in a significantly different way. In the next sections, we study how these strategies may be framed by considering instrumental directions (Section 5) and Zones of Proximal Evolutions (Section 6).

Before we skip to these sections, let us stress again the general perspective. Appropriation is not an issue to be solved, but a phenomenon to be taken into account. What design needs to respond to is not appropriation, but the potential technological obstacles to appropriation and/or the difficulties users face due to their appropriation process outputs, if any. Designing for appropriation consists of opening up new usage possibilities, and not in replacing original design (in which case it may just replace one constraint by another) or in disparaging users' workarounds for managing their problems (which may continue to prove their value). Users will decide what is good for them. Changing usages may be an objective, e.g. for a company considering employee efficiency, but this is another kind of objective.

5. Framing design with instrumental directions

5.1. Matter of concern

The issue we consider in this section is: how can email client designers be helped to identify which adaptation features or design alternatives should be offered? As most email clients are designed as native applications enhanced by plugins, we will mainly focus on this plugin technical approach. (Plugin is used here as a generic term. In some systems or communities, the terms plugin, add-on or extension may refer to slightly different notions.)

5.2. Notion of instrumental direction

As we saw, the way users take advantage of email clients to manage a given task may evolve and lead to other concerns. Intra-user variability thus requires considering more than individual functional values.

To frame this perspective we propose the notion of instrumental direction, which we define as a particular direction in which usage develops. Typically, the "to-do" functional value may be seen as part of the more general "task management" instrumental direction. Together with functional values, instrumental directions provide an analytic lens for analyzing usages and how design may respond to them. However, instrumental directions may prove more stable than functional values, and more predictive of future usages.

Let us illustrate this idea of users engaging in a direction. Usage studies reveal some users attribute a to-do functional value to their inbox. Taking a "task management" instrumental direction perspective shifts the focus from the underlying immediate need to the usage dynamics: what is likely to come together, or to come next? Since email clients have been studied for decades, we know what ensued. Using emails as a to-do instrument paved the way for prioritizing urgency or importance via adaptation means and workarounds, and also to personal archiving. One example of an emerging issue for which users now develop different workarounds is the way they deal with emails for PIM across different devices (Cecchinato et al., 2016). How these multi-device aspects will impact usage evolution, and what will come next, is not necessary foreseeable. The constant evolution of the socio-technical context creates new needs for which new means must be found. However, we do know usages will evolve, and individual tactics vary.

5.3. Using the notion of instrumental direction to analyze and enhance the plugin offer

Instrumental directions may be used as analytical lenses to review the current adaptation substratum and, on this basis, consider how it may be documented, improved and/or enhanced.

Let us first illustrate the review phase with the email client Mozilla Thunderbird. In ready-to-use task management features, Thunderbird offers the Lightning plugin, which makes it possible to define tasks, events and calendars. (Actually, Lightning has now been embedded in Thunderbird.) As

for task management resources, Thunderbird offers general features such as folders, tags or filters. However, using the instrumental direction "task management" (and the practices revealed by usage analyses mentioned in Section 3) as an analytical lens, a review of Thunderbird plugins identifies many different plugins that may be used for task management. See Table 1.

Table 1. Thunderbird adaptation substratum for task management¹.

Current features (*) and plugins (+)		Examples of how they may be used for task management
*Tags	Define tags and associated colors, and assign one or several tags to emails.	Denote tasks; denote urgency; denote personal thoughts related to the email/task; etc.
*Folders	Create folders and sub-folders, and file emails out of the inbox.	Store emails as related to some semantics; limit inbox content to tasks to be remembered only; etc.
*Filters	Automatically implement an action on received or sent emails.	Create filters to automatically tag emails and/or move them into a specific folder according to their characteristics (e.g. sender, subject or content).
*Quick Filter Toolbar	Easily create filters for basic characteristics, including tags.	Double-click access to a subset of emails from their semantic tags. Interestingly, allows constructions such as "all tags" or "one of these tags", i.e. the premises of an algebraic tool.
+Color Folder	Color folder icons.	Use colors to denote common semantics of emails (via tags) and folders (via this plugin). Using colors to differentiate between types or importance of tasks is a known pattern (Haraty et al., 2016).
+Tag Toolbar	Extend the tagging native feature in different ways, such as structuring tags into categories.	Develop a complex system of tags/tasks. As a first example: create a "task" category, and then different tags "task#1", "task#2", etc.; when tagging an email, selecting a category (e.g. "task") prompts for "task#1", "task#2", etc. As a second example: create an "urgency" category and different levels of urgency, or different reasons why the task is urgent.
+Mailbox Alert	Specify an action (e.g. prompt message) that will be executed when a new mail is found in a given folder.	Create semantic folders (e.g. a folder by task), create filters automatically filing emails in folders, and associate a prompt message to these folders. When a new email associated with the "observed" tasks is received, an alert is given.
+SubSwitch	Define a set of prefixes: when writing a new email, one may then insert a prefix chosen from the list, which will appear in the email subject.	Create a structure set of semantic prefixes. May be used as tags (e.g. using folders and filters; see above), with an interesting difference: the prefix is sent with the email, which makes it possible to share the semantics with recipients and, possibly, create a collective semantics. May also be used to prompt and scaffold a reflexive analysis of one's email content, similarly to the sentence openers used in some communication systems.
+RemindIt	Associate a message with the deadline date for responding to it: the message is colored in green, will change to orange when the expiration date is close, and to red when the date is expired.	Turn the semantics from the intended usage "deadline for responding the message" to "deadline for dealing with the task". The email and its colors now act as an automatic task reminder. In particular, this partially addresses the discrepancy between the date of reception and the task urgency.
+Send Later	Specify when a message will be sent.	Edit an email-to-self to remember a task and set a date to send/receive the reminder when useful (here again, this addresses different issues of emails-to-self as reminders mentioned in Section 3).
+Thunderlink	Create links to emails into personal or collective wikis.	Use email as a reminder and document the task in the wiki, which is a way to address one of the important disadvantages of using emails for PIM.

¹ <https://mozilla.org/thunderbird> (last consulted October 2017). This list is based on the plugin descriptions, and is for illustration only. It is the result of an external analysis (we are not engaged in Thunderbird or Mozilla projects).

+XNote	Add notes (post-its) to messages.	Document tasks. Usage studies showed that one of the issues when using email as to-dos is the cognitive cost of rebuilding one's thoughts each time one reconsiders the emails. As received emails cannot be edited, a workaround is to edit one specific ad-hoc tag mentioning one's thoughts. A proper notepad is of course more efficient, and addresses the intrinsic length limit of tags.
+Task Me	Periodically display a text box for users to describe the tasks they are currently performing	Promote reflection on one's activity in terms of tasks.
+Follow Up	Associate a message with a reply date and create a Lightning event.	Provide a reminder of email/task deadlines in a way which makes a connection with PIM dedicated features.

As a second step, the notion of instrumental direction and a review such as the one presented in Table 1 may be used to guide design in three different ways.

First, it may be used for indexation and knowledge sharing. Thunderbird's plugins are presently organized in "Categories" such as "Appearance and Customization", "Calendar and Date/Time", "Folders and Filters", "Import/Export" or "Tags". Table 1 plugins are widespread into different categories. How and why they may be used for task management (i.e. an instrumental perspective) may be made more obvious. This may be enhanced by application-specific forums where users productively exchange tips and workarounds. Early research already emphasized that software adaptation cannot be considered a primarily individual activity, and showed how some users act as leaders and help others (Trigg and Bødker, 1994; Gantt and Nardi, 1992; Mackay, 1990). More recently, to better understand how to make users aware of adaptation features, Haraty et al. (2016) asked users how they became aware of the features of their applications. Accidental discovery came first. However, regarding their preferred method, recommendations from other users came first (61%), before intentional browsing (49%).

Second, instrumental direction may be used to improve the coverage and coherency of the plugin offer. Taking a task management perspective draws attention to limits and potential improvements. Some examples are sending or receiving tags with emails as a means to develop usages based on collective folksonomies; enabling articulated use of semantic tags and colors on both emails and folders; or addressing email-task granularity issues. This may be addressed by creating new plugins and/or designing meta-plugins allowing inter-operation of other plugins (see Section 6).

Finally, instrumental direction may be used to inform the integration of coherent sets of features in new releases (which is what happened with the integration of Lightning in Thunderbird).

Before we skip to the next section, let us emphasize that the task management instrumental direction we used in this section is only an example. Other examples of instrumental direction revealed by usage studies are personal archiving and contact management (Whittaker et al., 2006). Adaptation means may find simultaneous utility for several of these. For instance, Bota's et al. (2017) study shows emails-to-self are used as reminders (53%) and also personal archiving (e.g. passwords, photos of receipts or backups). Actually, while using email to save or access data ubiquitously was a new and efficient strategy in the 2000's, cloud infrastructures have changed the technical offer. Nevertheless, practices co-exist (inter-user and intra-user), and also require workarounds. As examples: turn the email client from a "pop" to an "imap" configuration; use a "detach" feature to store large files attached to emails in specific folders, which requires a specific folder organization. Here again, taking an instrumental direction perspective may inspire plugin design.

6. Framing design with Zones of Proximal Evolution

6.1. Matter of concern

The issue we consider in this section is: how can designers be helped in considering what means users may productively use? While not perceived as an affordance, any set of plugins remains a smart response to a non-existent (or not yet existent) question. Actually, although standard market email clients offer specific features to define and manage tasks, what could have become an outdated and abandoned usage continues, including for users aware of these features.

6.2. Importance of conceptual issues

Notions and conceptualizations (i.e. differential systems of notions) are the substratum of reasoning processes and actions. The productive use of means such as plugins requires, first and foremost, mastery of the underlying notions.

While external analysts use the notion of task to make sense of usages, users do not necessarily conceptualize their activity in terms of tasks, at least not in an explicit and/or precise way. The empirical study by Copic Pucihar et al. (2016) reveals how, while notions as tasks or projects may be clearly defined/structured in corporate or institutional contexts, this is usually very different on a personal level. As another example, Singh et al. (2013) highlight that there is a gulf between inbox and calendar. As a completely different example, notice how many word processor users face difficulties when attempting to define personal styles. Using this adaptation means requires mastering an array of more or less complex notions (from paragraphs or fonts to style dependencies). The gap between the concepts considered by users and those underlying system implementation is a well-known issue of works considering end-user tailoring (Stevens et al., 2006).

With respect to design, this analysis draws attention to the notions underlying the plugin offer, and whether these notions are or may become meaningful to users. Of course, there is little doubt that most users may be explained the task notion, taught how to use task-management features, and succeed in defining a set of tasks from their to-do emails. This, however, is not the issue at stake. What is at stake is whether the notion may become a useful psychological tool in action.

In order to help designers address this issue we propose the notion of "Zone of Proximal Evolution".

6.3. Considering Zones of Proximal Evolutions (ZPEs)

In line with our developmental perspective, considering which notions and features users may productively use may be addressed in the light of the Zone of Proximal Development (ZPD) notion, i.e. "the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance, or in collaboration with more capable peers" (Vygotsky, 1978). ZPD is a concept that has been used as a starting point for a variety of studies in developmental and educational research.

With respect to HCI concerns, the ZPD notion may be used in different ways. Bryant et al. (2005) use it to make sense of how Wikipedia authors develop different usages in relation to different conceptualizations. The ZPD notion is referred to in order to analyze how novices, who address Wikipedia as a set of articles only, are aided by other more knowledgeable users to handle new activities (e.g. maintaining the integrity of the articles or the respect of some social rules), a process within which features not used by novices become meaningful to them (e.g. the watchlist). Bardram and Bertelsen (1995) frame the objective of allowing "transparent interactions" by offering conditions for creating a ZPD of operations. Norros et al. (2011) use the concepts of "zone" and "measurement" to evaluate the potential of a new technology by measuring the difference between the outcomes and the ways of acting of users aided or not aided by technology.

The notion of ZPD is inspiring but raises theoretical (and practical) difficulties. In particular, defining a precise ZPD requires explicit criteria (and practical possibilities) for analyzing not-aided and aided problem-solving levels, as Norros et al. (2011) do. With respect to appropriation as the progressive development of instruments to solve the problems one considers in the way one considers them, this opens up a range of open questions. Therefore, in order to avoid conceptual mismatches, we propose consideration of a notion which is inspired from the ZPD notion only.

We define a Zone of Proximal Evolution (ZPE) as a zone of potential development of users (potential development of new instruments), hypothesized from users' present practices and underlying notions, and made technologically possible by a given technical substratum. Proximal is used to refer to small/incremental developments.

The notion of ZPE draws attention to (1) the actual set of instruments users have constructed by (i) attributing task-management related functional values to email and (ii) taking advantage of the

available adaptation means; (2) the accessible instruments that may offer users a broader level of achievement of task management, which may include (i) considering new (or modified) goals and thus attributing new functional values to the technology, or (ii) developing more efficient technical implementations of present instruments, i.e. more efficient ways of taking advantage of the technology; and (3) the technical substratum that users should be offered, as well as scaffolding.

With respect to design, the ZPE notion is a way to conceptualize and frame the following best-effort strategy. (1) From usage studies, draw attention to functional values and instrumental directions. (2) Identify or hypothesize the notions underlying users' actions, i.e. the conceptual and technical zone within which they currently act. (3) Identify notions which may help users extend this zone and improve how they achieve the tasks at hand. (4) Scaffold users in considering these notions, in particular by offering adaptation means which mobilize them and connect them with the ones currently used. This rejoins the way Bryant et al. consider what notions make sense of users' present and possible future activities; the developmental concerns underlying Bardram's and Bertelsen's perspective; and, as a general concern rather than precise measurements as Norros et al., paying attention to the extent to which conceptualizations and practices may evolve.

Let us apply this strategy to our email example. Usage studies reveal (1) a task management instrumental direction and (2) the use of notions such as tags or notes for task-management concerns. The notion of *task* may arguably help users struggling with their present task management process and/or involved in a competence improvement process. Although available to users (via Lightning), it is not necessarily used. In order to enhance the chances that technology does not hinder but, rather, supports personal development, users should be offered as affordances different means that mobilize the notion of task. In particular, specific attention should be drawn to how users may close the gap between the conceptualizations underlying their present practices and the notion of task.

In the next sections we illustrate how, drawing on the limits/issues of the present plugin offer as highlighted by the usage analyses, this strategy may be implemented to design new plugins (Section 6.4) and meta-plugins (Section 6.5).

6.4. Enhancing the offer with designed-from-appropriation plugins: a task management example

The Lightning task management feature makes it possible to create tasks from emails using a right click. A task is featured by fields such as start/end dates, status or category. Such a ready-to-use application arguably helps connect email and task notions.

When taking an appropriation perspective, and as for any ready-to-use application, different aspects of Lightning's design may be questioned. For instance, the informal conceptualization of tasks revealed by Copic Pucihar et al.'s (2016) study opens space for both inter-user and intra-user variability. While using emails to manage tasks is compatible with an informal and/or situated task notion, using a predefined fixed fields may raise issues. The fact that notions that users use in practice only apparently correspond to those of the application, and are difficult to project onto the technical device, is often one of the explanations why the device is not used. Another potential issue is the 1-1 relation between emails and tasks. While creating a communication thread requires answering a topic, email clients only make it possible to answer emails as technical entities, which may cause confusion and breakdowns (Reyes and Tchounikine, 2003). In particular, the 1-1 relation between emails and tasks may become an issue for tasks denoted by several messages.

Specific design may support users to connect email and the task notion (and use the latter) in a way that avoids such potentially hindering characteristics. As an example, MyTask is a proof of concept plugin prototype designed for such a purpose. Flexibility issues (not imposing a predefined representation nor uniformity) are addressed by allowing definition of one or more task types, i.e. the fields describing tasks. This allows definition of basic structures (e.g. a task is associated with a name and an editable text zone) and/or specific structures for specific concerns. Email-task granularity and cardinality concerns are addressed by allowing creation of a task from an email portion only and, more importantly, by adding messages to an existing task (see Figure 1).

MyTask is an illustration of how adopting an instrumental direction and ZPE lens provides insights for offering users new means. It mobilizes the notion of task and its link with email with a

design rationale that increases users' chances of finding/building a technological substratum for their perspective to tasks. As such, it may help users close the gap between the conceptualizations underlying their present practices and the notion of task.

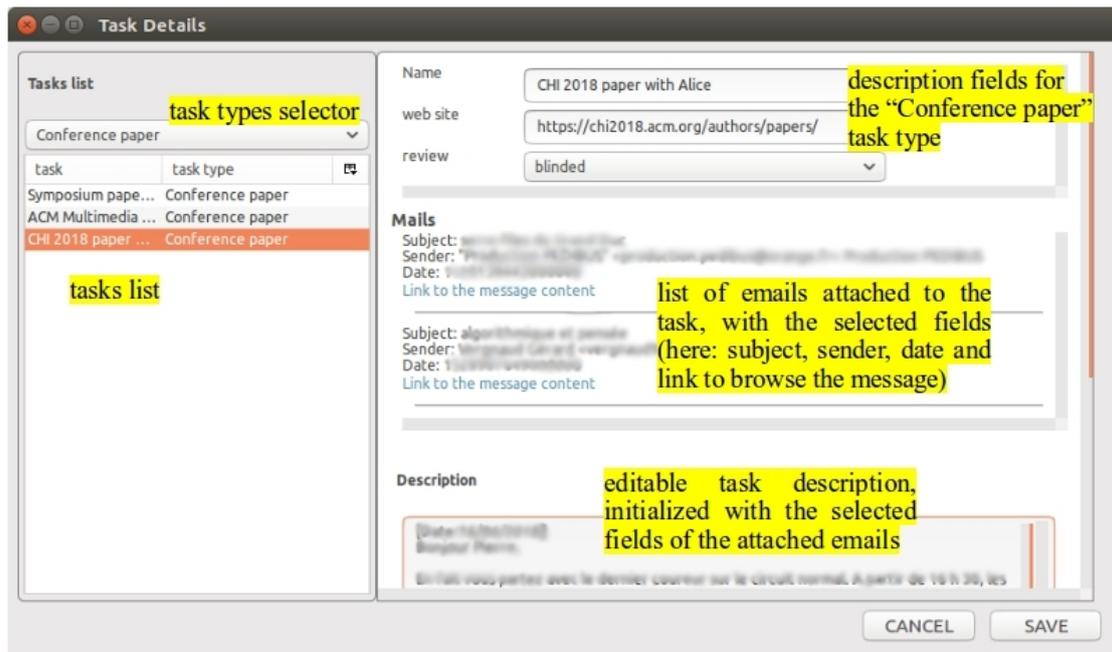


Figure 1. Snapshot of a plugin offering a flexible task structure and n-n email-task relations

Such a design is both design *from* appropriation (designers respond to needs as they emerge from usage studies) and *for* appropriation (high flexibility). Once again, such plugins are to be offered as alternatives. As a matter of fact, MyTask also implements specific design decisions which may not fit the developed/developing instruments of some users. Users may also continue not to find any additional value in the task notion, and/or find specific value in their previous ways of acting.

6.5. Enhancing the offer with meta-plugins: connecting emails, tags, notes and tasks

Another strategy for paving conceptual development is to create connections between the notions at stake via meta-plugins, which we define as plugins allowing inter-operation of other plugins without modifying them (epiphyte technical approach).

Let us illustrate this. Usage studies show that users engaged in task management via emails use different workarounds, some of which take advantage of tags and notes. Similarly to MyTask, a plugin combining features related to emails, tags, notes and tasks may be designed. This has the pros of designed integration, and the cons of possibly hindering unexpected usages. As an alternative, offering basic inter-operation of existing plugins via meta-plugins switches pros and cons. Table 2 synthesizes meta-plugin prototypes that were considered and implemented as proofs-of-concept.

Table 2. Meta-plugins

Features		Potential uses for task management
TagToNote	Given a tag, creates a note (via Xnote) and attaches it to all the tagged emails.	Step from "tasks denoted by tags" to "tasks denoted by notes", i.e. in a way that allows for personal annotation.
TagToTask	Given a tag, creates a task from the tagged emails in MyTask.	Step from "tasks denoted by tags" to a proper task description. Thanks to MyTask, the created task features the link to the different messages (1-n cardinality).
NoteToTask	Given a message with an attached note, creates a task whose description is the text of the note.	Edit and attach personal annotations to an email and, when deciding to skip to a proper task description, avoid information loss.

TaskToTag	Given a (MyTask) task, creates a tag (by default, the name of the task) and tags all the linked emails.	Highlight the different emails linked to a given task and/or use the tag for sorting or filtering actions.
TaskToNote	Given a (MyTask) task, creates a note and attaches it to the different emails linked to the task.	Manage tasks as such in MyTask (global description and links to associated emails) and add specific information to each email.

The specific interest of meta-plugins is to loosely link different concepts which present different structural and functional properties but, also, are related to one another via different cardinalities. Meta-plugins acknowledge that users may manage tasks via different notions (emails, notes, tags and/or proper task descriptions) and leave users flexibility to create relations between these notions via their practices. For instance, given a set of emails tagged to mention they are related to a given task, this task may be further documented using TagToNote (to attach specific information to each email) or TagToTask (to edit a single comprehensive piece of information). These possibilities may be combined, and practices may vary from case to case. In other words, by allowing the orchestration of other plugins, meta-plugins open up possibilities for different usages, including unexpected ones. Later on, usage analyses may identify particular workflows for which it may make sense to design a packaged streamlined version (design *from* appropriation as for MyTask), though still offering the basic meta-plugins for possible idiosyncratic and/or innovative usages to emerge.

As for MyTask, meta-plugins may be seen as both design *from* and *for* appropriation with, however, a significant difference with respect to the *from* (identifying notions to be inter-operated vs. designing an integrative feature). More precisely, the design rationale is to connect the different plugins or features mobilizing an identified set of notions. In order to allow unexpected usages, this should not be limited to considering the plugins whose features may appear to be linked pertinently. What is at stake is not whether or how combinations seem efficient or desirable for the designer, but to offer users a technical substratum. As an example of inter-operation with no immediate, obvious advantage, yet which may be offered, consider the meta-plugin we did not list in Table 2: NoteToTag.

6.6. Discussion

In Section 4.2, we defined designing for appropriation as opening up new usage possibilities via adaptation means. The plugin and meta-plugins presented above illustrate how instrumental directions and ZPE may act as conceptual tools to frame the design of such means. With respect to scaffolding, they also illustrate how new means may extend and/or inter-operate with existing ones.

From a design perspective, and although "neutral design" is an oxymoron, the objective of leaving different usages possible suggests keeping plugins and meta-plugins as simple as possible. When considering meta-plugins, different "smart" features come to mind, implementation of which would, however, lead to an integrative plugin similar to MyTask². Although we believe that designers must be supported in considering what adaptation means should be offered to users, if only to focus efforts, one may attempt to generalize the process and systematically connect notions and plugins, independently from any usage analysis (no more design *from*). Some notions underlying different plugins that it could make sense to connect would be: clock, invitation, monitoring, contacts, archive or instant messaging. Finally, although we focus on conceptual difficulties, offering plugins requiring different technical levels may also increase the chance that some of them will be in the user's ZPD, i.e. in the space within which he/she may act productively. As we reflected in terms of plugin offer, the negative impact of offering useless means remains limited. This would be very different if the available technology were replaced.

From a software engineering perspective, design must be framed into a strategy acknowledging that appropriation is a long-term process, and that usages and (needs for) adaptations progressively co-develop. The SER meta-design model ("Seeding, Evolutionary growth, and Reseeding", proposed

² When considering end-users, some meta-plugins require basic yet designed user interfaces. For instance, turning a tag or a note into a MyTask task requires selecting the task type. Pushing the meta-plugin idea to its limits would lead to considering raw data pipelines (in the Unix sense).

by Fischer et al., 2004), appears to be a pertinent framework. Once seeded with a priori adaptation means, evolutionary growth may be accompanied by continuous design or modifications of plugins and meta-plugins in line with the instrumental directions as they emerge, the usage of which will inform the next release (re-seed).

Finally, from a technical perspective, meta-plugin implementation shows that it is possible to avoid the "not invented here" trap and, using an epiphyte approach, take advantage of other plugins whilst not modifying them. However, although (and, actually, because they are) conceptually simple, some aspects of MyTask and meta-plugins implementation were made frustratingly difficult due to lack of documentation and/or the ad hoc implementation of the original code to which they connect. Designing for appropriation requires a community perspective facilitating inter-operation (APIs, systematic use of databases, documentation, etc.).

7. Discussion and additional information

7.1. Importance of considering users' ecology of artifacts

As mentioned in Section 1, in this article we used email and longitudinal analyses of its usage to illustrate our analyses and proposals. However, if studying an application (e.g. email) and/or a given task (e.g. PIM), analyzing usage of one artifact in isolation is misleading. It is mandatory to consider users' ecology of artifacts (Bødker and Klokmoose, 2012; Kaptelinin and Bannon, 2012).

Users draw on all available means. For instance, the study by Haraty et al. (2016) shows how users distribute to-do items on email, e-calendars, paper agendas, e-agendas or web browsers, with email being used in most combinations. These devices may or may not be contextually available, and have both overlapping and different properties. Moreover, email is used on multiple devices such as laptops and smartphones, which raises specific issues. As an example, Cecchinato et al. (2015) mention that it is often useful to keep track of the device with which email was first opened, and suggest tagging emails with device icons to do so (yet another functional value for tags). More generally, users develop different workarounds depending on the device (Cecchinato et al., 2016).

Taking our theoretical perspective, it may be said that users' instruments may build on the coordinated use of different artifacts. Through practice, users may thus develop new instruments in different ways: new artifacts may be given a functional value; functional values may evolve from already-present artifacts, in relation to the evolution of the artifact and/or of the usage schemes; and the way instruments are related one to the other may also change to form successive systems of instruments (Tchounikine, 2017). As a consequence, adaptations or requests for adaptation of one application may be difficult to interpret, appear contradictory and/or unstable when considered individually, but may gain coherence if considered in terms of ecology of artifacts and instrumental directions. Similarly, different uses of artifacts may be explained by shared or related functional values. Analyses should thus consider users' systems of instruments. This raises the research question of considering instrumental directions distributed over several artifacts.

A recent study of today's communication applications well illustrates how appropriation cannot be regarded for one application only, and develop for other reasons than technical features. One of the present technical ecosystem characteristics is the multiplicity of communication applications (Messenger, WhatsApp, etc.), and also the fact they create specific channels, unlike emails which may be read from different email clients. Analyses by Nouwens et al. (2017) show that "*[users] appropriate the features and technical constraints of their apps to construct personal communication environments*". These communication places have specific membership rules, perceived purpose and emotional connotations; are personal to the user and not related to the technical features only; and evolve, i.e., user's actions and experiences shape the communication place, which in turn shapes user's actions. For example, users manage emotional issues such as not wanting to see a contact at the top of the contact list; not wanting to see a contact in the app at all while reluctant to delete it from the list (even if the correspondent does not know this, i.e. moral etiquette for one's own sake); skipping from one app to another as app identities and/or contact statuses evolve; etc. This leads users to develop personal strategies based on more or less explicit rules (e.g. criteria for contacts to

be on one app and not another), develop workarounds (e.g. managing to push the contact down in the list or saving past messages to another media), and face breakdowns (e.g. turning-off indication of when last on-line applies to all contacts, when family, friends or colleagues should be differentiated). Let us frame this using our perspective: given that emotional states, attribution of identities to apps, idiosyncratic usage rules and strategies for maintaining these rules may be subject to intense intra- and inter-variability, design should include explicit consideration of adaptation features.

7.2. Importance of considering social level and work practice

Social level and work practice are also mandatory aspects of any usage analysis. Typically, these aspects impact and may even structure appropriation processes of email clients.

Let us take a simple but intuitive example: summarizing the list of tasks a working group has to handle in an email, with a few lines for each task. Two strategies may be considered: sending one email with the list of tasks, or sending one email per task. Individual usage is likely to develop in relation to how the group implicitly or explicitly decides to communicate, e.g. having one "reference" email or developing task-related threads. The usages that crystallize at the social level (or are imposed by the hierarchy) may be consistent with, contribute to, or come into conflict with individual usages and the associated use of the email client. This will impact the fact that plugins designed to avoid breakdowns related to the granularity issue and the n-n relations between emails and tasks appear as useless or on the contrary, very useful and even necessary.

Another issue, which again stresses the importance of adaptation means, is that many users use applications they did not choose. As email clients are often free or included in packages, one may infer that users are active in selecting the applications that best fit their needs. In direct contrast, surveys indicate that many users do not use email clients that are chosen individually or rationally. They use or choose a given email client because it is imposed on them, they are used to it (possibly because it was imposed on them), it is the currently or locally most-used email client, or just because it is pre-installed on their computer (Ducheneaut et al., 2001; Haraty et al., 2016).

7.3. Value for the "email for PIM" analysis for other systems/contexts

The value of this study for other systems/contexts is dual

First, email is far from being a unique case of unexpected usages, and the type of analysis we presented in this paper may be conducted for other systems. Another good example is how spreadsheet usage influences or even structures activities (presentation, coordination, decision-making or archiving) for which they have not been designed (Dourish, 2017).

Second, understanding a given technology usage and its evolution is useful because it impacts how new technologies will be used. Users often project onto new artifacts working procedures that are derived from the use of old ones. Yates and Orlikowski (2002) mention that "*in moving their communication to a new medium, members of a team or community typically import existing genres and genre systems, improvise around them, and gradually learn to take advantage of new opportunities afforded by the medium*". Based on an impressive survey of the transference of PIM research prototype concepts to the mainstream (67 systems including 37 email clients, ranging from 1987 to 2012), Kljun et al. (2013) show that most of the successfully transferred technologies "*had a slow and smooth integration into present application environments, without interfering with the current trustworthy user practices*", adding to existing tools rather than attempting to replace them.

As an example, while email was an outcome of the technological/infrastructural breakthrough "computer-to-computer asynchronous communication", more recent breakthroughs allow for new types of applications such as drives, cloud-based project management suites and communication apps. Although early users are naturally aware of, and excited by, new advances, developing an appropriation perspective may not be pointless. As an example, when considering collaborative edition, an on-line editor is arguably efficient. However, these applications also raise issues related to individual/collective perceptions of a "work-in-progress", and if individuals feel at ease or not with sharing thoughts and drafts rather than fully-worked-out texts (Tchounikine, 2011).

7.4. Relations with other works

Designing for appropriation may be seen as a specific case of end-user development. An important specificity of the approach studied in this article is, however, to consider adaptations which do not require any programming. Although high-level visual languages may ease technical difficulty (see for example (Mørch et al., 2017)), programming implies shifting to another specific activity.

Well in line with many aspects of our analysis, Kaptelinin and Bannon (2012) suggest allowing "intrinsic practice transformations" (transformation initiated and accomplished by users when contradictions within existing practices appear) by designing assemblies of artifacts "from the outside", via integrative technologies such as meta-tools or connectors. The design and implementation of generic meta-tools or connectors is an open question. Although much more focused, our consideration of meta-plugins for a given application rejoins this idea.

Finally, another way to consider appropriation issues is to consider the affordance notion as the entry point. Given our theoretical perspective, the literature proposing/discussing an AT-perspective for affordances is of particular interest (Bærentsen & Trettvik, 2002; Kaptelinin & Nardi, 2012). For instance, following Bærentsen and Trettvik, we may say that adaptation features are not affordances as such: they become affordances if users relate them to their activity. However, the Structuration Theory approach to affordances (Vyas et al., 2017), which suggests addressing affordances as a dynamic emerging relationship between people and their environment, intersects the perspective presented in this paper in many ways.

8. Conclusions

In this paper, we studied why users should be provided with adaptation means, and how the design of such adaptation means may be framed: usage analyses in terms of functional values and instrumental directions; review of current adaptation means in the light of these analyses; consideration of the notions underlying users' present practices; enhancement of the current offer with plugins and meta-plugins, taking into consideration conceptual difficulties and scaffolding; and indexation and documentation of adaptation means in light of instrumental directions, as a way to draw attention and create the conditions for peer support.

The way email clients are used for PIM well illustrates appropriation processes. The facts that email is a communication/activity hub ("an habitat") and may be adapted or cheated relatively easily both play a key role. For other types of applications, appropriation phenomena may be less obvious, but also more problematic. If the task reminder instrument a user has built becomes incompatible with the email client's technical features, because of changes within the psychological part of the instrument and/or the application features, the email client may lose this functional value of remembering tasks. As this is related to the unexpected functional value of email clients, and many other means exist, it may be considered as not catastrophic. However, for other applications, if such an evolution occurs for the application's expected functional value, and/or for unexpected functional values that cannot be attributed to another artifact, more serious breakdowns may occur. Consideration of appropriation phenomena, and designing for appropriation, is thus essential.

9. References

- Bærentsen, K. B., & Trettvik, J. (2002). An activity theory approach to affordance. *Proceedings of the NordiCHI 2002 Conference on Human-Computer Interaction*. New York, NY: ACM
- Bardram, J.E., Bertelsen, O.W., 1995. Supporting the development of transparent interaction. In *International Conference on Human-Computer Interaction*, pp. 79-90.
- Bellotti, V., Ducheneaut, N., Howard, M., Smith, I., Grinter, R.E., 2005. Quality versus quantity: E-mail-centric task management and its relation with overload. *Human-Computer Interaction* 20 (1-2), 89-138.
- Bødker, S., Klokmose, N.C., 2011. The Human-Artifact Model: An Activity Theoretical Approach to Artifact Ecologies. *Human-Computer Interaction*, 26(4), 315-371.

Bødker, S., Klokmoose, N.C., 2012. Dynamics in artifact ecologies. In: Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design, pp. 448–457.

Bota, H., Bennett, P.N., Awadallah, A.H., Dumais, S.T., 2017. Self-Es: the role of emails-to-self in personal information management. In: Proceedings of the CHIIR 2017, pp. 205–214.

Bryant, S. L., Forte, A., Bruckman, A., 2005. Becoming Wikipedian: transformation of participation in a collaborative online encyclopedia. In Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work, pp. 1-10.

Cecchinato, M. E., Sellen, A., Shokouhi, M. Smyth, G., 2016. Finding email in a multi-account, multi-device world. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 1200–1210, New York, USA, ACM.

Cecchinato, M.E., Cox, A.L. Bird, J., 2015. Working 9-5? Professional differences in email and boundary management practices. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 3989-3998.

Civan, A., Jones, W., Klasnja, P., Bruce, H., 2008. Better to organize personal information by folders or by tags?: The devil is in the details. In: Proceedings of the Association for Information Science and Technology, 45(1), 1-13.

Copic Pucihar, K., Kljun, M., Mariani, J., & Dix, A. J., 2016. An empirical study of long-term personal project information management. *Aslib Journal of Inf. Management*, 68(4), 495-522.

Dabbish L.A., Kraut, R.E., 2006. Email overload at work: an analysis of factors associated with email strain. In: Proceedings of the 2006 ACM Conference on Computer Supported Cooperative Work, Banff, Alberta, Canada, pp. 431–440.

Dix, A., 2007. Designing for Appropriation. In: Proceedings of the 21st British HCI Group Conference, Volume 2, Lancaster, UK, pp. 28-30.

Dourish, P. , 2017. *The stuff of bits: an essay on the materialities of information*. MIT Press.

Dourish, P., 2003. The Appropriation of Interactive Technologies: Some Lessons from Placeless Documents. *Computer Supported Cooperative Work* 12: 465-490.

Ducheneaut, N., Bellotti, V., 2001. Email as Habitat: An Exploration of Embedded Personal Information Management. *Interactions* 8(5), 30-38.

Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A. G., Mehandjiev, N., 2004. Meta-Design: A Manifesto for End-User Development. *Communications of the ACM* 47(9), 33-37.

Gantt, M., Nardi, B. A., 1992. Gardeners and gurus: patterns of cooperation among CAD users. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 107-117.

Grevet, C., Choi, D., Kumar, D., & Gilbert, E., 2014. Overload is Overloaded!: Email in the Age of Gmail. In: Proceedings of the SIGCHI conference on human factors in computing systems, ACM, pp. 793-802.

Hanrahan, B. V., Pérez-Quiñones, M. A., Martin, D. (2014). Attending to email. *Interacting with Computers* 28(3), 253-272.

Haraty, M., McGrenere, J., Tang, C., 2016. How personal task management differs across individuals. *International Journal of Human-Computer Studies* 88, 13-37.

Jones, W., Capra, R., Diekema, A., Teevan, J., Pérez-Quiñones, M., Dinneen, J.D., Hemminger, B., 2015. “For Telling” the Present: Using the Delphi Method to Understand Personal Information Management Practices’. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, ACM Press, New York, USA, pp. 3513–3522.

Kaptelinin, V., Bannon, L., 2012. Interaction design beyond the product: Creating technology-enhanced activity spaces. *Human-Computer Interaction* 27(3), 277-309.

Kaptelinin, V., & Nardi, B. (2012). Affordances in HCI: Toward a mediated action perspective. *Proceedings of the CHI 2012 Conference on Human Factors in Computer Systems*. NY: ACM

Kaptelinin, V. Nardi, B., 2006. *Acting with Technology: Activity Theory and Interaction Design*. Cambridge: MIT Press.

- Kaptelinin, V., 2003. UMEA: translating interaction histories into project contexts. In: Proceedings of the Conference on Human Factors in Computing Systems, New York, ACM Press, pp. 353-360.
- Kljun, M., Mariani, J., Dix, A., 2013. Transference of PIM research prototype concepts to the mainstream: successes or failures. *Interacting with Computers*, 27(2), 73-98.
- Lieberman, H., Paterno, F., Kalnn, M., Wulf, V., 2006. End User Development: an emerging paradigm. In: *End User Development* (Lieberman, H., Paterno, F., & Wulf, V. eds.), Kluwer Publishers, pp 1-8.
- Mackay, W.E., 1990. Users and customizable software: A co-adaptive phenomenon. Doctoral dissertation, Massachusetts Institute of Technology.
- Mackay, W.E., 1988. More than just a communication system: diversity in the use of electronic mail. In: Proceedings of the ACM conference on Computer-supported cooperative work, pp. 344–353.
- Mørch, A.I., Caruso, V., Hartley, M.D., 2017. End-User Development and Learning in Second Life: The Evolving Artifacts Framework with Application. *New Perspectives in End-User Development*, pp. 333-358.
- Mørch, A.I., 2011. Evolutionary Application Development: Tools to Make Tools and Boundary Crossing. In: *Reframing Humans in Information Systems Development* (Isomäki H. & Pekkola S. eds), Springer, pp. 151-171.
- Müller, H., Gove, J.L., Webb, J.S., Cheang, A., 2015. Understanding and comparing smartphone and tablet use: Insights from a large-scale diary study. In Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction, pp. 427-436. ACM.
- Nouwens, M., Griggio, C. F., Mackay, W. E., 2017. WhatsApp is for family; Messenger is for friends: Communication Places in App Ecosystems. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 727-735.
- Norros, L., Liinasuo M., & Hutton, R. (2011): Evaluating the potential of new technology tools for safety critical work. *Interacting with Computers* 23, 299-307.
- Rabardel, P., Waern, Y., 2003. Special issue “From Computer Artefact to Instrument for Mediated Activity”. *Interacting with Computers* 15 (5&6).
- Reyes P., 2003. Supporting Emergence of Threaded Learning Conversations Through Augmenting Interactional and Sequential Coherence. In: Wasson B., Ludvigsen S., Hoppe U. (eds) *Designing for Change in Networked Learning Environments*. Springer, Dordrecht, 83-92.
- Sappelli, M., Pasi, G., Verberne, S., de Boer, M., Kraaij, W., 2016. Assessing e-mail intent and tasks in e-mail messages. *Information Sciences*, 358, 1-17.
- Singh, N., Tomitsch, M., Maher, M. L., 2013. Understanding the management and need for awareness of temporal information in email. In: Proceedings of the Fourteenth Australasian User Interface Conference, Volume 139, pp. 43-51.
- Stevens, G., Quaisser, G., Klann, M., 2006. Breaking It Up: An Industrial Case Study of Component-Based Tailorable Software Design. In: *End User Development* (Liebermann, H., Paterno, F. & Wulf, V. eds), Springer, pp. 269-294.
- Tchounikine P., 2017. Designing for Appropriation: A Theoretical Account. *Human-Computer Interaction* 32(4) 155-195.
- Tchounikine, P. (2011). *Computer Science and Educational Software Design - A Resource for Multidisciplinary Work in Technology Enhanced Learning*. Springer.
- Trigg, R. H., Bødker, S., 1994. From implementation to design: tailoring and the emergence of systematization in CSCW. In Proceedings of the ACM conference on Computer supported cooperative work, pp. 45–54.
- Vyas, D., Chisalita, C.M., Dix, A., 2017. Organizational affordances: A structuration theory approach to affordances. *Interacting with Computers* 29(2), 117-131.
- Vygotsky, L.S., 1978. *Mind and society: The development of higher psychological processes*. Cambridge: Harvard University Press.

- Whittaker, S., Bellotti, V., Gwizdka, J., 2006. Email in personal information management. *Communications of the ACM* 49(1), 68-73.
- Whittaker, S., 2005. Collaborative task management in email. *Human-Computer Interaction* 20 (1&2), 49-88.
- Yates, J., Orlikowski, W., 2002. Genre systems: Structuring interaction through communicative norms. *Journal of Business Communication* 39(1), 13-35.