Proactive Behavior of a Personal Assistive Agent

Karen Myers and Neil Yorke-Smith*

Artificial Intelligence Center, SRI International, Menlo Park, CA 94025, USA
{myers,nysmith}@ai.sri.com

Abstract. The increased scope and complexity of tasks that people perform as part of their routine work has led to growing interest in the development of intelligent personal assistive agents that can aid a human in managing and performing tasks. As part of their operation, such agents should be able to anticipate user needs, opportunities, and problems, and then act on their own initiative to address them. We characterize the properties desired for *proactive* behavior of this type, and present a BDI-based agent cognition model designed to support proactive assistance. Our model for proactive assistance employs a meta-level layer to identify potentially helpful actions and determine when it is appropriate to perform them. We conclude by identifying technical challenges in developing systems that embody proactive behaviors.

1 Introduction

We are interested in developing intelligent personal assistive agents that can aid a human in managing and performing complex tasks. Our overall goal is to reduce the amount of effort required by the human to complete the tasks she intends. Effort here encompasses both the activities necessary to perform the tasks, and the cognitive load in managing and monitoring them. Thus, a personalized assistive agent may aid its user directly by performing tasks on her behalf or in conjunction with her [1], and indirectly through actions such as providing context for her work, minimizing interruptions, and offering suggestions and reminders [2].

We are exploring these ideas within a system for intelligent personalized assistance called CALO [3]. The focus for CALO is to support a busy knowledge worker in dealing with the twin problems of information and task overload. CALO's current task-related capabilities are grounded in a *delegative BDI model* [4], in which the system adopts intentions only in response to being explicitly assigned goals by the user. CALO can perform a variety of routine office tasks delegated by the user, such as arranging meetings and completing online forms, as well as more open-ended processes such as purchasing equipment or office supplies and arranging conference travel.

One limitation within the current CALO framework is the lack of a *proactive* capability that would enable a CALO agent to anticipate needs, opportunities, and problems, and then act on its own initiative to address them. We are interested in developing proactive behaviors along these lines within CALO, to increase the overall effectiveness of the system as a personal assistant.

* Authors listed alphabetically

Our research objectives in this area are as follows. First, we want to understand the types of proactive behavior that would be helpful to incorporate into an assistive agent. Second, we want to characterize how an agent can best reflect over possible actions and current commitments (both user and system), as a guide to which intentions to adopt and how to pursue them. Finally, we would like to develop a theory of proactivity that characterizes when an agent should take initiative to assist the user, and the nature of the assistance that should be given.

Our approach to proactivity employs a meta-cognition layer that reasons about *what tasks* may be appropriate for the agent to perform and *when* initiative should be taken to perform them. Such reasoning is inherently meta level: in addition to requiring deliberation over a model of the user's state and intentions, it further requires that the agent reason about its capacity to perform potentially helpful tasks given its current commitments and capabilities. Such meta-reasoning is essential to ensure that proactive behavior will not interfere with current and planned actions of the agent and user, unless the benefits of doing so outweigh the associated costs. This type of reflective reasoning can be viewed as complementary to more common forms of agent meta-reasoning related to control of deliberation, conflict resolution, explanation, and goal selection.

After describing related work in Sect. 2, we characterize the properties desired of proactive behavior in Sect. 3 and present examples within an office assistance scenario. Sect. 4 presents an extended BDI framework with a meta-level layer designed to support proactive assistance, along with requirements on a theory of proactivity to drive this behavior. We outline additional technical challenges in developing systems that embody proactive behaviors in Sect. 5, and conclude in Sect. 6.

2 Related Work

Isbell and Pierce [5] describe an *Interface-Proactivity continuum* that ranges from zero to full automation: 'Do It Yourself', 'Tells You to Pay Attention', 'Tells You What to Pay Attention To', 'Makes Suggestions', and 'Makes Decisions'. While systems developed in artificial intelligence often lean toward the autonomy side of this continuum — in contrast to work in human-computer interaction, which has emphasized the interface side — CALO, like some other previous efforts to design assistive agents, spans several points. We next survey a selected subset of these efforts.

The Office Assistant in Microsoft Office was based "in spirit" on the Lumière project [6]. Although more limited in scope than CALO, Lumière shares an ambition to help a user. The domain of the project was user tasks in an office software package; Lumière deliberately considered proactive behavior. In expressing the rationale, Horvitz et al. [7] describe the meta-level cognition intrinsic in the desired behavior:

> [W]e have also been interested in the question of deliberating about when to step forward to assist a user. We believe strongly that such intrusions should be done in a careful and conservative manner, with the express approval of users. We built versions of the Lumière prototype that employ Bayesian models to control such "speculative assistance" actions, coupled with user-interface designs that showed promise for minimizing disruption and for putting the user in control of interruptions. On putting the user in control, we designed a speculative assistance display that included a prominently featured "volume control" for interruptions; the volume control would always be available

whenever such advice would appear. Such a volume control allowed users to move a slider in a "bother me less" or "bother me more" direction, changing the threshold at which active assistance would be provided.

The Electric Elves project [8] developed personal assistants with a range of functions related to supporting a busy office worker — an application domain similar to that of CALO — including a set of proactive capabilities designed to further general user objectives. These included, for example, delaying or rescheduling meetings when the user was likely to be late.

There has been much recent work on assistive technologies to aid people with cognitive disabilities in managing their daily activities in which proactivity plays an important role (e.g., [9, 10]). These systems monitor a person's actions to understand what she is doing, and interact when appropriate to provide reminders, situationally relevant information, and suggestions to aid in problem solving.

Theories of collaborative problem solving clearly relate to the notion of proactive assistance. For the most part, theories of collaborative activity extend Belief-Desire-Intention (BDI) models of agency [26] to incorporate notions of joint beliefs and commitments. For example, Joint Intention theory [11] formalizes the communication acts between agents to establish and maintain joint belief and intention: the obligations on what 'message' to communicate and under what circumstances to do so. A form of meta-level reasoning over agent state drives the collaborative behavior.

Planned Team Activity theory [12] also captures collective intentionality by introducing plural-subject constructs; applications of this formalism focus mostly on team formation. SharedPlans theory [1], in contrast, contains only single-subject intentions, but augments them with an 'intend that' construct. It specifies collaborative refinement of a partial plan by multiple agents, handling hierarchical action decomposition and partial knowledge of belief and intention. The theory expressly includes meta-cognitive tasks, such as cultivating collaborative goals by deliberating over commitments.

COLLAGEN [13] is an example of a system that instantiates these ideas, drawing on the SharedPlans theory. It provides a framework for building assistive agents that collaborate with a human to achieve tasks together. Its assistance is based around precoded models of tasks; the current task state is described as a collaborative dialogue consisting of a plan tree (tracking the state within the task model) and a focus stack (tracking the current focus of attention). By reasoning over these structures, COLLAGEN responds to user actions and utterances by acting or communicating appropriately.

Applications of Joint Intentions have focused on dialogue management in agent collaboration (e.g., [14, 15]). Applications of SharedPlans have included also task allocation and analysis of helpful behavior [16], but are limited in terms of proactive scope.

Collaborative problem solving and proactive assistance are both rooted in the notion of an agent taking action to assist another. Indeed, a collaborative agent will often need to act proactively to fulfill its commitments to its partner. Proactive assistance goes beyond collaborative problem solving, however, in that an agent may take actions unilaterally on behalf of its user without any joint agreement as to the desirability or suitability of the actions.[1] In fact, it is precisely this degree of autonomy that makes

[1] One can construe an implicit agreement between user and agent that the agent act as the user's assistive agent; compare the discussion of the general desire $\hat{\mathbf{d}}$ in Sect. 4.

proactive assistance potentially valuable, as it enables the agent to support its user without interfering with her normal activities.

Although a *theory of proactivity* will have much in common with theories of collaboration such as Joint Intentions and SharedPlans, whereas those frameworks focus on high-level characterizations of how and when to provide assistance, an operational theory of proactivity requires further elaboration of these concepts. In particular, a proactive assistive agent must have an explicit model of user desires, in addition to current user goals and plans, as well as a theory that defines how those can be furthered by actions that the agent is capable of performing. In addition, a proactive assistive agent must weigh the cost and benefits of potential goals and the plans to achieve them [17] to assess which are appropriate to perform. The requirements for and organization of these meta reasoning functions are our focus for the remainder of this paper.

3 Characterizing Helpful Assistance

Ethnographic studies of human work habits and task management (e.g., [18, 2]) reveal that people usually achieve all their important tasks. We become adept at multi-tasking and remembering the things that really matter; however, we fail to perfectly achieve tasks with soft deadlines, or to remember less critical details. It is these areas where assistance technology can thus provide greatest utility.

It is not our goal to address the general problem of inferring user intent [6, 17]. Although research on CALO encompasses recognizing from her actions what task a user is working on [3], our starting point (for the moment) is that the user has entered a description of her tasks and tasks assigned to her agent into an electronic todo list [18, 19]. In addition, we assume that the agent is told or can infer a mapping from a subset of these entries to formal models within a *task ontology* (i.e., the tasks have associated semantic descriptions), potentially drawing on techniques such as those described in [20]. Similarly, we assume that the user employs electronic artifacts to keep track of her commitments and calendar, and works within an instrumented desktop environment.

Within this setting, there are four challenges we must address in order to develop effective personal assistive agents. Each challenge involves the agent reasoning about its actions, and also meta-reasoning about the reasoning itself. In the sequel, we distinguish tasks performed solely by the user (*user tasks*) from those performed solely by the agent (*agent tasks*), and those performed jointly in partnership (*shared tasks*).

When to act? A personal assistive agent acts when delegated tasks by its user, and when it is obliged to act by commitments it has made (e.g., an agreement with a merchant to purchase an item on its user's behalf). Our hypothesis is that the agent can act proactively at other times, under its own initiative, in order to assist its user. A pivotal meta-control issue is answering the question of under what circumstances the agent should consider some proactive action.

What actions to take? We envision a personal assistive agent aiding its user in many ways. On some occasions its assistance will be initiated proactively; on other occasions by the user or as a result of actions by another agent.

The agent should consider actions it thinks will be helpful to the user, after a cost-benefit analysis.[2] However, since there will be a measure of uncertainty about the current situation, the user's goals and focus of attention, and the effects of its actions, the agent's proactivity should be careful: safe and deferent to the user. The further we move to the proactive end of the Interface-Proactivity continuum, the greater the care required of a helpful assistive agent in its deliberation.

How to perform the actions? Once the agent has decided to act on its own initiative, or is compelled to act, it must deliberate further about the modality and timing of its action [21]. Is it better to do nothing, to suggest, to confirm then act, or to act without consulting the user? To act now or later? To interrupt or not?

How to learn to do it better? A helpful assistant should seek to broaden and refine its problem solving skills through learning. Such a learning assistant should seek feedback, implicit and explicit, in order to better know when to act, what to do, and how to do it. As Isbell and Pierce point out [5], the agent must maintain effective operation and yet obtain feedback for learning; an example is the online active learning employed in the time management component of the CALO system [22].

3.1 Examples of Proactive Behavior

Figure 1 provides a list of possible proactive activities that an assistive agent might perform on behalf of its user in an office setting (such as that of CALO). We divide the list into four categories: *Act directly*, *Act indirectly*, *Collect information*, and *Remind, notify, ask*. The following scenario illustrates how a proactive behavior on the part of an intelligent assistant can provide value in the office assistance domain.

CALO observes the items currently in your electronic todo list, what you are currently working on, what you have delegated to your CALO and to people, and your commitments for the week ahead. CALO assesses that your workload is likely to be uncomfortably high at the end of the week. Via a chat message, CALO offers you a reminder of an important meeting early next week, with the suggestion that a paper review (on your todo list) could be transferred to a colleague (whom CALO identifies as having appropriate expertise and time in his schedule), to leave you time to focus on the meeting. In addition, CALO begins to prepare background material for the meeting without being explicitly asked. It attaches the relevant documents to the item in your todo list and the event in your calendar.

This scenario illustrates two distinct types of proactive behavior for an agent. The first type, which we call *task-focused proactivity*, involves providing assistance for a task that the user either is already performing or is committed to performing; assistance takes the form of adopting or enabling some associated subtasks. Task-focused proactivity is exemplified in the above scenario by CALO collecting background information

[2] The agent might even refrain from acting on a user-delegated task, if it can provide sufficient justification to the user. Of course, the agent's role as an assistant requires that the user be able to override the agent's decisions, as well as more broadly to advise its operation.

- **Act directly**
  - perform the next step of a shared task
  - perform or prepare for future steps of a shared task now
  - initiate the first step of a shared or agent task
  - suggest (shared) tasks the agent can take over and perform
  - establish a learning goal (i.e., for the agent to learn new capabilities)
- **Act indirectly**
  - suggest a user task be delegated to a teammate
  - suggest a meeting be rescheduled
  - suggest a lower priority task be postponed to free resources
  - suggest (better) ways to achieve a (shared) task
  - anticipate failures of (shared) tasks and look for ways to reduce the failure likelihood or the impact of failure
- **Collect information**
  - gather and summarize information relevant to a user or shared task
  - monitor the status of tasks delegated to a teammate
  - monitor and summarize resource levels and commitments
  - analyze the possible consequences/requirements of a (shared) task
- **Remind, notify, ask**
  - remind of upcoming deadlines and events
  - remind of the user's next step in a shared task
  - monitor and filter incoming messages
  - ask for feedback/advice from user
  - ask for clarification or elaboration of a (shared) task

Fig. 1. Possible Proactive Activities

in support of a scheduled meeting. We note that systems such as COLLAGEN are designed for task-focused operation, although not task-focused proactivity.

The second type of proactive behavior, which we call *utility-focused proactivity*, involves assistance related to helping the user generally, rather than contributing directly to a specific current task. An example of this type occurs in the scenario when CALO takes the initiative to recommend transferring a paper review task in response to the detection of high workload levels. This action is triggered not by a motivation to assist with any individual task on the user's todo list, but rather in response to a higher-level motivation (namely, workload balancing).

3.2 Principles for Proactive Behavior

We define nine principles to guide desired proactive agent behavior, akin to the principles for intelligent mixed-initiative user interfaces of [23, 24]:

- **Valuable**: advances the user's interests and tasks, in the user's opinion
- **Pertinent**: attentive to the current situation
- **Competent**: within the scope of the agent's abilities and knowledge[3]

[3] Iba [24] points out that competence by itself is not sufficient for helpfulness and, further, argues that limited incompetence need not preclude helpfulness.

- **Unobtrusive**: not interfering with the user's own activities or attention, without warrant
- **Transparent**: understandable to the user
- **Controllable**: exposed to the scrutiny and according to the mandate of the user
- **Deferent**: gracefully unimposing
- **Anticipatory**: aware of current and future needs and opportunities
- **Safe**: minimizes negative consequences, in the user's opinion

The principles reflect the centrality of the user and her experience. The agent's actions are valuable only if they ultimately add value for the user. They assist only if they are performed in a manner that takes account of the user's focus and immediate as well as longer-term needs. Horvitz et al. for instance capture the behavior sought in the Lumière project: "The sensibility of an intuitive, courteous butler ... potentially valuable suggestions from time to time ... genuine value ... minimal disturbance." [6]

Prior research on assistive agents emphasizes the importance of ease of understanding by the user of the agent's operation, together with ease of directing, ignoring and correcting the agent, as well as working entirely without it [8, 10, 23]. Transparency and controllability are essential to build trust, which is especially important in an agent with an extended life cycle, such as a user's assistant [25], and even more so if the agent acts on its own initiative.

Returning to the earlier example, CALO's actions are pertinent to the important upcoming meeting. CALO itself is not capable of reviewing the paper; identifying a colleague who potentially is able, CALO does not delegate the task from your todo list automatically, but leaves you in control to take the suggestion or not. This suggestion and the preparation of background materials are both safe, defined in this case by an absence of changes of state other than a gain in information. Throughout, CALO's actions are unobtrusive: the communication is via a chat message with context, and the completed information gathering is again in context, attached to the relevant artifacts in your working environment.

4 An BDI Framework for Proactivity

Having characterized helpful proactive assistive behavior, we now introduce — more concretely — an extended BDI model of agency designed to support such behavior. More specifically, we define a meta-level layer that augments a BDI framework to enable (a) identification of potentially helpful actions, and (b) determination of when those actions should be performed. This proactive functionality critically relies on agent deliberation over both its capabilities and commitments, as well as the capabilities, commitments and state of the user,

We begin in Section 4.1 with background material on BDI frameworks, including brief overviews of two specific frameworks that have influenced our architectural design for proactivity. Sect. 4.2 then describes our meta-level framework for proactivity. Sect. 4.3 sketches requirements for a theory of proactivity that would be embedded within the architecture, while Sect. 4.4 outlines an approach to operationalizing this theory in a working system.

4.1 Background

BDI frameworks have become a popular architectural choice for the design of cognitive software agents. The BDI model provides an explicit, declarative representation of three key mental structures of an agent: informational attitudes about the world (Beliefs), motivational attitudes on what to do (Desires), and deliberative commitments to act (Intentions). This explicit representation enables ready inspection of an agent's operation by an observer or by the agent itself, thus enabling a range of reflective, metacognitive capabilities.

The primary deliberative processes of a BDI agent can be broadly characterized as focusing on *goal selection* (i.e., identifying what intentions to pursue), and *action selection* (i.e., how to pursue them). This reasoning necessarily takes into account the current BDI cognitive state of the agent to determine what is feasible and desirable given current beliefs and commitments. BDI agent frameworks such as PRS [27] and SPARK [28] implement these decision-making processes as a combination of base- and meta-level reasoning: simple strategies are hard-coded in the base level of the agent but more sophisticated agent-specific meta-level strategies can be invoked as needed. In particular, meta-level reasoning supports control of deliberation, conflict resolution, identification of learning goals, and as described in Sect. 4.2, proactive behavior by an intelligent assistant.

Dignum et al. [29] adopt three motivational attitudes for an agent. In their *B-DOING* framework, *desires* correspond to purely internal motivations (what the agent wishes), *obligations* correspond to specific external commitments (responsibilities to another agent), and *norms* correspond to general societal motivations and conventions (from the agent's role in society). For example, one might have a desire to pay the least for a book, an obligation to honor one's bids on eBay, and a norm to fulfill public commitments before private. Roughly, the three motivations balance the agent's own interests (desires), those of other agents (obligations), and those of society (norms). All three may conflict within themselves and with each other. From the sum of these motivations, such an agent generates goals, which it must believe to be consistent, achievable, and so forth [30, 4]. From its goals, the agent adopts intentions for execution, in standard BDI fashion.

Whereas the B-DOING framework distinguishes different types of motivational attitudes, the *delegative BDI model* [4] distinguishes different types of goals: Candidate Goals — a set that may be internally inconsistent, and so forth — and Adopted Goals. The latter set has key properties (consistency, coherence) that are simply assumed of goals in most BDI frameworks.[4] Designed specifically to support an assistive agent, the delegative BDI model also incorporates forms of user-specified guidance and preferences on the execution of these tasks, and on the agent's cognition, called *Advice* (motivated by [31]).

The distinction between desires, goals, and intentions is important for an agent to meta-reason about the consistency, coherence, and so forth, of its goals [30]. The distinction between desires and the two classes of goals is also important for an agent that

[4] Many implemented BDI agents assume that the desires of the agent are consistent and feasible, and effectively treat goals and desires as equivalent; thus, these systems might better be called B(G)I implementations.

does not — at least, not immediately — pursue all of its consistent goals. However, the delegative BDI framework, like most frameworks in the BDI heritage but in contrast to the B-DOING framework, considers all motivations of the agent as desires, not distinguishing them by their possibly differing origins, natures, and implications. In fact, a delegative BDI agent can be seen to have a single desire, namely, to perform the goals delegated to it by its user.

A user-assistive agent will typically have motivational attitudes beyond desires. Such an agent will have obligations arising out of its actions: obligations that follow from the decisions it makes. Some of the agent's obligations will be process oriented, such as obligation to act or communicate according to commitments or agreements arising from a collaboration framework. In addition, because of its desire to assist its user (and because of the norm of the role of an assistant), the agent will take on obligations to perform what its user asks, i.e., delegated tasks in the delegative BDI model. Besides obligations, such an agent will be subject to the norms of expected behavior within the societal team context in which it operates. It will also have norms arising out of its role, such as to work for its user's perceived good.

Thus, while the B-DOING framework lacks the distinctions between types of goals for meta-reasoning about helpful assistance, the delegative BDI framework lacks the distinctions between types of motivational attitudes. Moreover, neither framework in the various aspects of its meta-reasoning encompasses deliberation in order to manifest proactive assistance by generating Candidate Goals.

As described in the next section, we chose to build our framework for proactivity on the delegative BDI model, with the recognition that, ideally, it should be extended to support the distinctions among motivational attitudes articulatd by the B-DOING model.

4.2 Architecture for Proactive Assistance

Figure 2 depicts proactive goal generation in an extension of the delegative BDI agent architecture. As usual in a BDI formulation, the agent's base-level cognition reasons about how to realize Adopted Goals as intentions. Multiple forms of meta-cognition are depicted to the right. In addition to the usual BDI meta-cognition over aspects such as agent control — for example, over goal selection — we show proactive goal generation, an extension to the prior delegative BDI model. Other forms of meta-cognition include the development of learning goals, for example, as shown.

For a personal assistive agent, the agent can be thought of as holding an overarching meta-desire of being a helpful assistant to its user, which we denote $\hat{\mathbf{d}}$. We can envision a limited number of additional desires at both the base and meta-levels. One desire might be to learn (although one could construe this as the agent bettering itself in order to become a better assistant). Indeed, in principle, the majority of an assistive agent's desires — or at least goals that might arise from them — can be considered as consequences of the overarching high-level desire $\hat{\mathbf{d}}$. In practice, we will choose an explicit representation of motivational attitudes, to avoid the complexity of excessive first-principles meta-reasoning during execution.

Candidate Goals (CGs) are created through two mechanisms. At the base level, they arise from the agent's motivations to achieve tasks delegated by the user. At the meta-

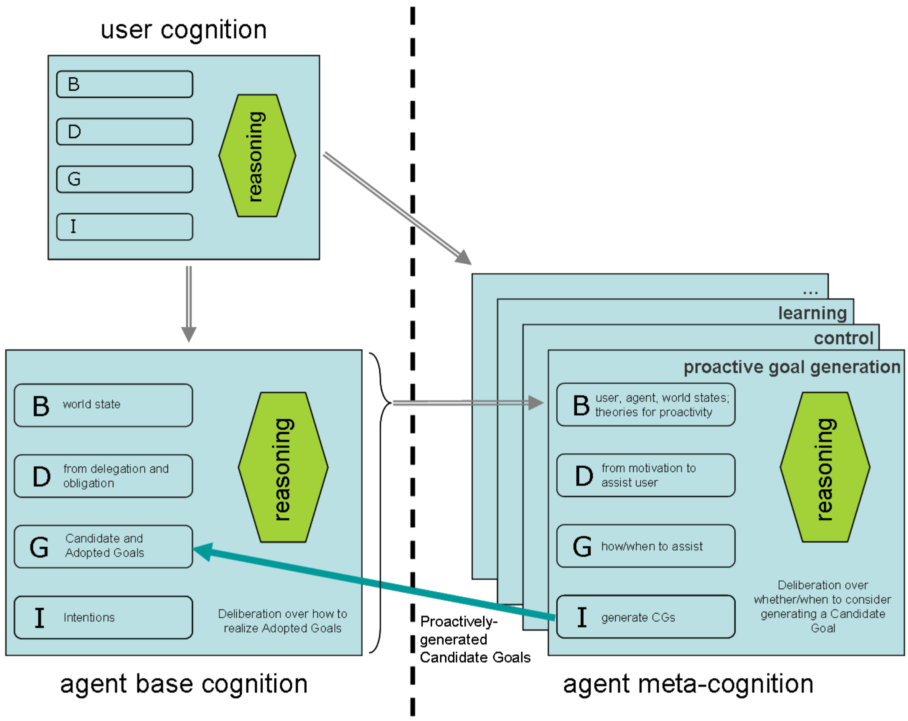

Fig. 2. Extended BDI Agent Architecture for Proactive Assistance

level, CGs are generated proactively as depicted, as a result of deliberation over a *theory of proactivity*, described further in Sect. 4.3. This aspect of meta-cognition, motivated by the high-level desire $\hat{\mathrm{d}}$, reasons over agent beliefs about user, agent, and world states, user and agent capabilities, as well as theories of helpful activity that we will describe.[5]

As we have noted, the agent's generation of a CG does not imply that it will necessarily adopt the CG. The control aspect of meta-level cognition chooses how to execute any adopted goal; in particular, CALO might wait before acting, suggest it acts, ask whether it should act, just act, and so forth. The architecture does not specify the characteristics of the agent's behavior, which will vary from agent to agent.

Similarly, the architecture does not specify the mechanism for reasoning to determine which CGs to create for transferral to the base-level portion of the agent. This strategy can be freely specified according to the character of the agent by appropriate instantiation of the theory of proactivity.

[5] Not shown in the figure is the user-stated advice of different kinds that guides the agent's goal deliberation. Desire and Goal Advice inform meta-level cognitive decisions: *Desire Advice* governs the goal generation process and *Goal Advice* governs goal selection. *Execution Advice*, which by contrast applies to the base-level, governs goal adoption as intentions.

4.3 Theory of Proactivity

In Sect. 3 we identified four challenges for an agent to support proactive assistance: when to act, what actions to take, how to perform the actions, and how to learn to improve its behavior. We now propose a meta-level *theory of proactivity* designed to support a proactive agent in meeting these challenges. This theory is composed of subtheories for *safe actions*, *user desires*, and *helpful activity*.

Theory of Safe Actions A theory of *safe actions* is necessary to define bounds on what an agent is allowed to do when performing tasks proactively. In particular, given that the agent will be acting on its own initiative without user awareness of its actions, it is important that only actions with benign or positive effects be performed, so as not to interfere with user activities or change the world in unexpected ways.

We postulate that the definition of a safe action varies from context to context, as some conjunction of: maintains world state; maintains world state except to increase knowledge; immediately reversible without cost; immediately reversible with negligible cost; reversible without cost; reversible with negligible cost; reversible; without negative consequence on user tasks; without negative consequence on tasks of others in the team; and with limited use of shared (team) resources.

Theory of User Desires A theory of user desires is necessary to describe the long- and short-term objectives of the user. Such a theory provides the means to assess the situated value of each potential agent action in terms of the user's objectives. The question for the agent is then: when are actions of varying degrees of safety, utility, and timeliness to be considered? If a task has many safe actions and high perceived benefit, should it be barred because one action is potentially unsafe, such as accepting a meeting request on the user's behalf?

While task-focused proactivity seeks to provide assistance to the user with immediate, tangible goals, in contrast utility-focused proactivity addresses more general objectives of the user. Utility-focused proactivity requires a representation of the user's unstated *interest goals* (in the terminology of the OCC cognitive model [32]) as well as explicitly stated *achieve* and *replenishment* goals. Since interest goals differ between individuals, a helpful assistive agent requires such a model (perhaps learned) of its user, in order to assess the value of agent actions.

Further, meta-reasoning over the sufficiency of the model (i.e., how complete, current, and accurate the agent believes is its knowledge of the user's desires) will, on one hand, guide the agent's goal deliberation and, on the other hand, guide its consideration of learning goals.

Theory of Helpful Activity A theory of helpful activity defines the principles to direct the agent's reasoning to determine what actions would most help the user now and in the future. In particular, this theory encodes the logic for selecting among possible intentions and the means to pursue them, given a characterization of current user desires.

4.4 Operationalizing Proactivity: Assistance Patterns

As Grosz and Kraus [33] note, BDI and collaboration theories are less often directly implemented in practical agent designs as much as used to provide a 'system specification', or even simply as a source of insight informing the design. For example, COLLAGEN's discourse reasoning algorithms originate from reasoning with SharedPlans' 'intend that' construct, but do not implement reasoning over such constructs.

Similarly, our planned implementation of the meta-level component for "proactive goal generation" in Figure 2 need not necessarily reason over explicit theories of user desires and helpful activities. Rather, those theories will be compiled into a form of meta-knowledge that we term *Assistance Patterns* (APs). Assistance Patterns provide a form of knowledge representation that bridges the gap from the high-level desire $\hat{\mathbf{d}}$ to help the user to more concrete motivations for the agent. APs are defined in terms of a set of *trigger conditions* that, when satisfied, identify one or more Candidate Goals to be passed to the base-level component of the agent. AP triggers are defined in terms of beliefs about the user's mental state (e.g., goals, commitments, focus of attention), the agent's own state, and the world state.

As an example, one AP cues from the trigger that there exists a shared task for which the agent can perform the next step; the CG that results is that the agent perform this step. As a second example, consider an AP that cues from a belief that the user has overlooked possible synergy between two tasks; the CG that results is that the agent propose notifying the user accordingly.

Many APs are *generic*: they are general capabilities that apply in any circumstance within the domain of the agent. For example, the AP to perform the next step of a shared task may be expected to be relevant as a source of CG generation for all users in all circumstances. Such generic APs could be built explicitly into the agent architecture. While there are likely a set of universally applicable patterns, in general it would make sense to also have certain patterns that are specialized to a given context.

5 Additional Technical Challenges

In this section, we note some additional technical challenges to be addressed in developing an operational assistant with helpful proactive behavior.

Understanding the User's Work Environment The example scenario presented above hinges on the assistive agent's capability to infer associations among and reason over information from the user's work environment, such as todo items, calendar entries, projects, resources, plans, and current task and location. How can this information be made accessible to an intelligent agent in a semantically grounded way?

For example, semantic information about user and shared tasks is critical. Three possible sources of information are (1) inference from user actions (intent and plan recognition) [6, 3]; (2) user semantic statement (as we assumed above); (3) inference from user nonsemantic statement, possibly confirmed with an explicit disambiguation request. In the third case, for example, the agent might identify that "book conference" on the user's todo list corresponds with a known task in the task ontology by leveraging techniques such as those described in [20]. Since studies show that people decompose their work into projects and todos at differing abstraction levels [2]; a related challenge is to identify the levels of abstraction at which to define a task ontology, and how to relate user-specified tasks into it.

Acquiring Task Parameters To act, if it is ever to adopt an intention from a task, the agent must have instantiations for the task's input parameters. For example, it is not enough to identify a `conference-travel` shared task: the agent must know to which conference the user is referring. One approach is to acquire the parameters to instantiate the task fully, by asking the user to specify them. Since this risks disturbing the user with a perceived irrelevant request (unless the user has just asked her agent to perform the task, the request comes out of the blue), another approach is to guess the parameters values from learned history, or to perform information-gathering actions (e.g., look at the user's calendar).

A second approach is to act on a set of possibly matching, partially instantiated tasks by performing safe, conditional actions, one for each possible task instantiation. For example, gather flight and hotel quotes for each conference. A strong notion of safety is needed here, since the agent must not reserve a flight to each venue. In conjunction or instead, the agent can perform conformant actions that support any of the possible tasks. For example, whatever the destination, the user will need to submit a travel authorization form, and many of the fields can be prepopulated.

Personalization and User Interaction By adapting to its user's preferred working and communication styles, and her capabilities and experience (e.g., [34]), an agent becomes a more helpful and thus trustworthy assistant over time. A part of personalization that is central to user experience is that of interaction [23]. As we have argued, the question of how to perform the action can be as important as what action to perform. Thus, a helpful assistive agent will deliberate about both communication and action.

6 Conclusion

Proactive behavior by an assistive agent — which encompasses more than acting directly to achieve an assigned task — promises to make such an agent more helpful to its user. We have characterized the properties desired of such behavior, and presented an extended agent cognition model that features a meta-cognition layer charged with identifying potentially helpful actions and determining when it is appropriate to perform them. The meta-reasoning that answers these questions draws on a theory of proactivity that describes user desires, conditions under which it is safe to perform actions, and a model of helpfulness. Assistance Patterns represent a compiled form of this knowledge that instantiates this meta-cognition over the agent's beliefs about the user's mental state as well as over world state.

The significant technical challenges include goal representation and reasoning in line with the extended delegative BDI model and guided by a theory of proactivity. At the same time, critical for the operation of proactive behavior are knowledge of user desires and the world state (including the user's current task and focus), acquisition of task parameters, mixed initiative operation and agent-human interaction, and learning.

Our immediate next steps are to establish the theory of proactivity to formalize the desired behavior, and to implement the extended delegative BDI model and associated Assistance Patterns within the SPARK agent framework [28] that forms the heart of CALO's task execution ability.

Acknowledgments. The authors thank other members of the PExA project, and the anonymous reviewers for their suggestions. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency, or the Department of Interior-National Business Center (DOI-NBC).

References

1. Grosz, B., Kraus, S.: Collaborative plans for complex group action. Artificial Intelligence **86** (1996) 269–357
2. Czerwinski, M., Horvitz, E., Wilhite, S.: A diary study of task switching and interruptions. In: Proc. of CHI'04, Vienna, Austria (2004) 175–182
3. Myers, K., Berry, P., Blythe, J., Conley, K., Gervasio, M., McGuinness, D., Morley, D., Pfeffer, A., Pollack, M., Tambe, M.: An intelligent personal assistant for task and time management. AI Magazine **28** (2007) To appear.
4. Myers, K.L., Yorke-Smith, N.: A cognitive framework for delegation to an assistive user agent. In: Proc. of AAAI 2005 Fall Symposium on Mixed-Initiative Problem-Solving Assistants, Arlington, VA (2005) 94–99
5. Isbell, C.L., Pierce, J.S.: An IP continuum for adaptive interface design. In: Proc. of HCI International 2005, Las Vegas, NV (2005)
6. Horvitz, E., Breese, J., Heckerman, D., Hovel, D., Rommelse, K.: The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In: Proc. of UAI'98, Madison, WI (1998) 256–266
7. Horvitz, E.: Lumière project: Bayesian reasoning for automated assistance. `http://research.microsoft.com/~horvitz/lum.htm` (2005)
8. Chalupsky, H., Gil, Y., Knoblock, C., Lerman, K., Oh, J., Pynadath, D., Russ, T., Tambe, M.: Electric elves. AI Magazine **23** (2002)
9. Pollack, M.: Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment. AI Magazine **26** (2005) 9–24
10. Haigh, K., Kiff, L., Myers, J., Guralnik, V., Geib, C., Phelops, J., Wagner, T.: The independent LifeStyle assistant: AI lessons learned. In: Proc. of IAAI-04. (2004)
11. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. Artificial Intelligence **42** (1990) 213–261
12. Kinny, D., Ljungberg, M., Rao, A.S., Sonenberg, L., Tidhar, G., Werner, E.: Planned team activity. In: Artificial Social Systems. LNAI 850, London, UK, Springer–Verlag (1994)
13. Rich, C., Sidner, C.: COLLAGEN: A collaboration manager for software interface agents. User Modeling and User-Adapted Interaction **8** (1998) 315–350
14. Subramanian, R.A., Kumar, S., Cohen, P.R.: Integrating joint intention theory, belief reasoning, and communicative action for generating team-oriented dialogue. In: Proc. of AAAI-06, Boston, MA (2006)

15. Jennings, N.R.: Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. Artificial Intelligence **75** (1995) 195–240
16. Grosz, B.J., Kraus, S., Talman, S., Stossel, B., Havlin, M.: The influence of social dependencies on decision-making. In: Proc. of AAMAS'04. (2004)
17. Fern, A., Natarajan, S., Judah, K., Tadepalli, P.: A decision-theoretic model of assistance. In: Proc. of IJCAI'07, Hyderabad, India (2007)
18. Bellotti, V., Dalal, B., Good, N., Flynn, P., Bobrow, D.G., Ducheneaut, N.: What a To-Do: studies of task management towards the design of a personal task list manager. In: Proc. of CHI'04, Vienna, Austria (2004) 735–742
19. Conley, K., Carpenter, J.: Towel: Towards an intelligent to-do list. In: Proc. of AAAI 2007 Spring Symposium on Interaction Challenges for Intelligent Assistants, Stanford, CA (2007)
20. Gil, Y., Chklovski, T.: Enhancing interaction with to-do lists: Using artificial assistants. In: Proc. of AAAI 2007 Spring Symposium on Interaction Challenges for Intelligent Assistants, Stanford, CA (2007)
21. Horvitz, E., Kadie, C.M., Paek, T., Hovel, D.: Models of attention in computing and communications: From principles to applications. Communications of the ACM **46** (2003) 52–59
22. Weber, J.S., Pollack, M.E.: Entropy-driven online active learning for interactive calendar management. In: Proc. of IUI'07, Honolulu, HI (2007)
23. Horvitz, E.: Principles of mixed-initiative user interfaces. In: Proc. of CHI-99. (1999)
24. Iba, W.: When is assistance really helpful? In: Proc. of AAAI 2007 Spring Symposium on Interaction Challenges for Intelligent Assistants, Stanford, CA (2007)
25. McGuinness, D., Glass, A., Wolverton, M., da Silva, P.P.: Explaining task processing in cognitive assistants that learn. In: Proc. of FLAIRS'07, Key West, FL (2007) To appear.
26. Rao, A.S., Georgeff, M.P.: Modeling agents within a BDI-architecture. In: Proc. of KR'91, Cambridge, MA (1991) 473–484
27. Georgeff, M.P., Ingrand, F.F.: Decision-making in an embedded reasoning system. In: Proc. of IJCAI'89. (1989) 972–978
28. Morley, D., Myers, K.: The SPARK agent framework. In: Proc. of AAMAS'04. (2004)
29. Dignum, F., Kinny, D., Sonenberg, E.: From desires, obligations and norms to goals. Cognitive Science Quarterly **2** (2002) 407–430
30. Dastani, M., van der Torre, L.: Programming BOID-plan agents: Deliberating about conflicts among defeasible mental attitudes and plans. In: Proc. of AAMAS'04. (2004) 706–713
31. Myers, K.L., Morley, D.N.: Policy-based agent directability. In Hexmoor, H., Castelfranchi, L., Falcone, R., eds.: Agent Autonomy. Kluwer Academic Publishers (2003)
32. Ortony, A., Clore, G.L., Collins, A.: The Cognitive Structure of Emotions. Cambridge University Press, Cambridge (1988)
33. Grosz, B.J., Kraus, S.: Collaborative plans for complex group action. Artificial Intelligence **86** (1996) 269–357
34. Edwards, J.L., Scott, G.: LOCATE intelligent systems demonstration: Adapting help to the cognitive styles of users. In: Proc. of AAAI-06, Boston, MA (2006) 1937–1938