Activity Context Aware Digital Workspaces and Consumer Playspaces: Manifesto and Architecture

*Vikas Agrawal, Genoveva Galarza Heredero, Harsha Penmetsa, Arijit Laha, Lokendra Shastri

Center for Knowledge Driven Information Systems, Infosys Labs
Infosys Limited, Bangalore, India
*vikas_agrawal01@infosys.com (Corresponding Author), genoveva_g@infosys.com, harsha_penmetsa@infosys.com, arijit_laha@infosys.com, lokendra_shastri@infosys.com

Abstract
We define and propose a manifesto and an architecture for smart digital workspaces and consumer playspaces, that “know” what the user is doing (activity structure, context, goals), how are they doing it (methods), what resources are they using (allocation and discovery), when (time) and where (location, application, device) are they doing it, who are they (profile, history), what is their role (responsibility, security, privacy) and who are their collaborators (social network), all the while observing, recording this context of work and play (institutional and social tribal knowledge). These smart workspaces and playspaces to be developed in the next five years, will let the users seamlessly move between applications and devices without having to remember or copy what they did earlier (activity context transfer and exchange), proactively show them steps others took in meaningfully similar situations before (semantic task reasoning), quickly find and show them directly related information and present answers to questions based on what they mean (proactive semantic extraction and search), in the context they need it, with access to provenance, quality and derivation of information, connect them to insights of experts within the organization and beyond, helping them reason and decide faster, with greater confidence, within a framework for managing, semantically dividing, tracking and enabling distributed work. We report two examples of the application of this architecture: a patient care system in a hospital and an assistive living system.

Motivation
Pervasive context-aware computing technologies are essential enablers for next generation applications for the digital workplace, consumer electronics, research, education, government and health-care. These enhanced technologies are expected to be in the mainstream in the next five years. Context-aware cognitive support requires activity and context information to be captured and, ever more often, moved across devices - securely, efficiently and with multi-device interoperability.

As computers have become more powerful, the tools from AI and applications in HCI have come much closer. The academia and industry need to jointly work on creating systems that capture, transfer, and recall activity context across multiple devices and platforms used by people individually and collectively. There is a need to explore techniques to represent activity context using a synthesis of these approaches to reduce demands on people, such as the cognitive load inherent in activity/context/device switching and enhance human performance within activities.

The scope of this paper includes techniques for creating activity context driven systems providing end-user value through monitoring, exchange and support on activities which can be performed better with help of computational devices than otherwise. The research focuses on using AI techniques to improve the human-computer interface for better performance of knowledge work by humans. Therefore, applications in machine-to-machine systems (manufacturing, smart grid, load balancing), standard data mining and web-based behavioral analytics are kept out of scope. To help human users, the tools should enable proactive finding of unknown knowns and unknown unknowns, as shown in Figure 1, based on activity context.

Figure 1. Finding Knowns and Unknowns Based on Activity Context
Typical knowledge work involves acting on the known knowns to actively search for known unknowns. This problem has been well-addressed by databases, keyword-based search engines and improving quality of semantic search engines. The quality of the results of such work can be significantly enhanced by a computing system that proactively identifies and finds relevant data, best known methods, and steps others have taken before, from within and outside the user’s institutional or social setting. While such relevant data and best known methods exist in multiple repositories and forms, this diversity makes it very hard for a human user to find these unknown knowns. Further, fast and scalable numeric and symbolic inference may lead the knowledge workers to find insights in internal and external datasets that they might have never sought of their own accord, but that might have an immediate and major impact on their work output and level of engagement. We propose that by capturing, tracking, replaying and reasoning on activity context, these problems can be addressed to the desired depth, with corresponding investment of effort in activity modeling, domain modeling, inference systems, semantic search, activity recognition and user intent prediction, to create an integrated system. In the next section, we identify specific characteristics of contemporary and near-future workspaces, playspaces, workers and individuals. This will help us determine features needed to be supported by activity context-aware digital workspaces and consumer playspaces. In what follows we will use the term “workers” to represent knowledge workers and people who use consumer playspaces. We will use “workplace” to refer to both a digital workspace in industry and a consumer playspace.

Workers’ Characteristics and Trends

We describe below multiple implications of the key worker characteristics and trends summarized in Figure 2 above.

Aging Workforce:
The aging workforce results in loss of knowledge (Abowd, Bobick et al. 2002; Quintas 2002; Rappaport, Bancroft et al. 2003; DeLong 2004; Stevens 2010), loss of best known methods and tacit institutional tribal knowledge. This can only be captured properly using activity context capture tools, and retrieved using an efficient episodic memory semantic search system.

Movement to New Roles and New Employers:
As employees move to new roles and responsibilities (Droege and Hoobler 2003; McQuade, Sjoer et al. 2007), their previous work methods, expertise and knowledge needs to be captured and available for recall in context of on-the-job training of new workers. Similarly, where the employees are themselves new workers, if their new employers provide systems that enable them to learn from the best known methods developed by experts in the organization, this can significantly speed up the learning process.

Limited Memory, Multiple Disturbances, Application Switches, Context Switches, Short Time Frame of Focus, Short Attention Spans:
Most human actors have limited short term and long term memory and so it become very difficult to maintain context of one activity while switching to another activity and switching back (Kaiser 2008). The effort needed to create a context switch leads to major loss of time and effort.

Limited Access to “Valuable” Information:
As information in enterprise remain fragmented with multiple discordant schemas, and a mix of numerical data, structured text and unstructured text, workers find it very challenging to integrate data from multiple sources and make meaningful choices, decisions and conclusions in the context of their work. Semantic integration of disparate data sources and presentation after suitable inference relative to the domain business and technical processes, can go a long way in helping the users find relevant information. In addition, linked access to provenance of the information is critical to enable effective decision-making and follow-up.

Increasing Mobility, Distributed Locations, Limited Face Time:
More and more enterprises find their workforce distributed across time-zones, with multiple travel commitments, and limited face time with colleagues (Oshri, Van Fenema et
al. 2008; Vlaar, van Fenema et al. 2008; Basov and Shirokanova 2010; Srikanth and Puranam 2011). A system that can capture the context of work, and replay it for colleagues in a structured, algorithmically reproducible form, will form the foundation of seamless handover of tasks and enabling effective collaboration.

Widespread Access to Computer Networks and Availability of Small Computing Devices:
Ubiquitous access to computer networks provides an unprecedented convenience and simultaneously creates expectations for workers to attend to work wherever they are located. Multi-device delivery channels (Kousalya, Narayanasamy et al. 2011) and context transfer across devices are required to fulfill this need of today’s workforce.

Workplace Characteristics and Trends

We discuss the implications of some key workplaces characteristics and trends summarized in Figure 3 below.

Multiple Applications and Fragmented Data Sources:
As knowledge workers navigate multiple applications they need to manually transfer the context from one application to the next, and need to keep bridging the fragmentation between data sources. Context Capture Plugins (Antunes, Correia et al. 2011) that extract and transfer activity context from one application to another, one data source to another and one device to another.

Poor Search, Organize and Share Functions, Tribal Knowledge Lost, Decision Context Lost:
Multiple enterprises struggle with knowledge workers wasting precious time and effort with searching for valuable information that already exists in the organization and methods that have been well-documented in the organization (Von Krogh, Ichijō et al. 2000; Smith 2001; Seidler-de Alwis and Hartmann 2008; Goffin and Koners 2011). As decisions are made, the provenance and ownership gets lost, as statistical or business analysis is done the line of analysis is rarely captured in a form that is algorithmically reproducible in a structured manner. This implies direct loss of institutional tribal knowledge. If we can organize and share information and methods using the activity context in which they were created and where they are most relevant to be used, the search problem will be significantly alleviated.

These worker and workplace characteristics create hard problems, some of which can be meaningfully addressed using AI and HCI driven solutions.

Key Business Problems

Based on the above characteristics and trends, we have extracted the following list of the key problems that a worker is very likely to experience.

1. Work gets Done, Workers Move On, Knowledge Gets Lost:
   - Task forces run, decisions made, context lost.
   - Loss of decision-making parameters and best known methods.

   A loss of decision-making parameters and best known methods forces enterprises to rerun experiments, revisit decisions multiple times with multiple stakeholders, re-discover best known methods through trial and error even though it has been done before.

2. Multiple Applications, Devices and Data Sources:
   Work on one device or application, worker moves to another device or application, context and information lost.

   As workers move between devices and applications, the application context and device context of activity is lost, which forces the worker to re-do work, and recreate context as they move or as they resume work between interruptions or in the course of multi-tasking.

Moving between applications and data sources puts additional cognitive load on the user, causing loss of time, effort, and efficiency due to switching between multiple applications and data sources where there is lack of semantic interoperability

3. Search Done, Meaning Missed, Information Missed:
   Due to lack of semantic search tools in the context of the activity, relevant information is missed information and insights in structured and unstructured data are never discovered.
Manifesto for Activity Context Aware Digital Workspaces

We propose that the solution of the above problems can begin with a system that has the following as its key value propositions:

- Activity Context Aware Guidance
- Semantic Relevance in Information Retrieval
- Seamless Transfer of Activity Context Across Devices and Applications.

Therefore, we propose the following manifesto for activity context aware digital workspaces and consumer playspaces as shown in Figure 4.

We need...

...a smart digital workspace / playspace that knows

- what am I doing,
- how am I doing it,
- what resources am I using,
- where and when am I doing it,
- who am I, what is my role,
- who are my collaborators,

all the while observing, recording this context of work / play, (Institutional Tribal Knowledge)

These smart workspaces and playspaces...

- lets us seamlessly move between applications and devices without having to remember, or copy what we did earlier.
- prevents a framework for managing, meaningfully dividing, tracking and enabling distributed team work / play.
- quickly finds and shows us directly related information and answers to questions based on what we mean, in the context we need it, with access to the source, quality and how the information was derived, (includes offers/suggestions/guidance)
- connects us to insights of experts within the organization and beyond.
- progressively shows us steps others have taken in meaningfully similar situations before.
- helps us reason and decide faster, with greater confidence.
- Such smart workspaces and playspaces need to be created for specific domains.

Figure 4. Manifesto for Activity Context Aware Digital Workspaces and Consumer Playspaces

To achieve this manifesto we need research in multiple areas, including but not limited to the following:

Activity Modeling, Representation, Recognition, Detection, and Acquisition:
Which (low-level) human activities can be reliably learned and detected? How indicative are those for human tasks and intent? Which granularities of activities could be chosen for creating an extensible hierarchy of human activity? What types of, and to what extent, context information can be captured and incorporated in activity models? What are the most effective and efficient methods for incorporating context information in activity models?

Context Representation within Activities:
What machine languages are most suitable for activity representation to enable activity and context switching and context recall across devices, platforms and technologies? Do we need user-device specific activity and context dialogue sub-languages?

Semantic Activity Reasoning:
How to model and represent activities, objects, resources, actions and their semantics in their context during task performance? How do we design activity/context models to enable the searching of repositories of previous activities that have behaviorally and semantically similar components to current activity requirements?

Security and Privacy:
What features must be designed into activity /context models for information exchange across enterprise or private domain boundaries to enable masking, security and privacy measures without compromising user experience?

Information integration and Exchange:
How can we integrate and exploit the growing amount of information available from devices, services, the environment and general background knowledge to support activity context recognition tasks? What common ontologies or data vocabularies will be useful? What exchange techniques and formalisms will be most effective in specific domains? How can the externalized cognitive state transfer be properly affected?

What are the relevant use-case scenarios and collaboration environments? What are suitable software architectures, user interfaces, developer tools, benchmarking tools for activity-based computing? What kind of text, context and behavioral analytics is needed?

Context Capture:
How far can the context capture be automatic and to what extent will it require collaborative meta-dialogue between people and devices? What might be ways of determining the most relevant elements of context for a given task and for an activity/context switch?

Context Elements in an Activity

Below is a suggestive list of context elements in an activity based on fundamental interrogatives (See Figure 5):

User: Users work within a role, permissions, preferences, bringing past and immediate history, memory, skills, goals and perceptions.
**Type of Activity and Domain:** People create diverse activities in multiple domains, including but not limited to office work, healthcare, education, and entertainment.

**Social:** Users have the support of collaborators, connected devices and adjacent networks.

**Spatial and Temporal:** People may be at a certain geolocation, experiencing local conditions (weather, traffic, network connectivity). Tasks may be synchronous or asynchronous.

**Resources Available:** Users may have access to other people, databases, multiple applications, networks, related datasets, transportation methods, non-electronic resources (tools, paper etc.).

**Devices and Interfaces:** People may work on a variety of devices such as laptops, desktops, netbooks, tablets, cell phones, using multiple applications, operating systems and interfaces.

**Important Outstanding Research Issues**

Immediate focus and progress is required in the following research issues over the next 2-3 years:

- User/Intent Modeling (Teevan, Dumais et al. 2008), Activity Recognition (Donald 2005), Detection, Acquisition, Observation, Recording Tacit Knowledge (Eraut 2000) and Cognitive State Transfer.
- Activity Context Analysis (Nardi 1995), Modeling (Diaper 2011; Laha 2011), Representation (Dourish 2004), and Ontologies (Chen, Finin et al. 2003)
- Text, Context and Behavioral Analytics
- Security and Privacy (Shih, Narayanan et al. 2011)

**Business Value Delivered**

Over the next five years, technological trends will continue to increase the computational power of small mobile devices and their market penetration will continue to increase significantly. Demographic trends, e.g. an aging workforce in some countries, higher staff turnovers, and the still further increasing demands in areas such as education, healthcare, government, and industry and commerce, will all considerably benefit from the additional capabilities of context sensitive digital devices. Higher staff turnover, for example, has already stimulated organizations to capture and reuse the institutional knowledge of their experienced staff which, in the future, could be effectively deployed in context sensitive digital devices to provide cognitive support to employees.

Currently, corporations like Google (Android), Apple, Nokia and Microsoft are coming up with proprietary solutions which are likely to become increasingly divergent, although useful within their devices. To enable interoperability of task context across devices, research is needed towards common representation frameworks, techniques and languages.

There are solid commercial reasons to embark on creating a consortium for task context representation solutions and working towards finding common exchange protocols and languages. As a leading information technology research and advisory company, Gartner Inc. (Pettey 2010) has suggested that:

Context-Aware Computing - By 2016, one-third of worldwide mobile consumer marketing will be context-awareness-based. Context-aware computing will foster people to be more digital with the assets they have available. Context-aware computing is taking advantage of location and time and is a new era of augmented reality.

Context-aware computing centers on the concept of using information about an end user or object’s environment, activities, connections and preferences to improve the
quality of interaction with that end user. The end user may be a customer, business partner or employee. A contextually aware system anticipates the user’s needs and proactively serves up the most appropriate and customized content, product or service. Gartner predicts that by 2013, more than half of Fortune 500 companies will have context-aware computing initiatives and by 2016, one-third of worldwide mobile consumer marketing will be context-awareness-based (Pettey and Tudor 2010).

More than $150 billion of global telecom spending will shift from services to applications by 2012, and the global market for context-aware services will amount to $215 billion. “Unlocking this potential will be one of the next major challenges for IT,” said Mr McGee. “For example, we expect 75 per cent of new search installations to include a social search element. The world is digital and business leaders can’t ignore it.”

Application in Multiple Domains

Figures 6-9 summarize applications of the activity context aware digital workspaces and consumer playspaces in multiple domains of industry.

Proposed Architecture

We propose the following architecture for activity context-aware digital workspaces (Figure 10, Next Page) to provide assistance to users of existing computing systems, leveraging technology such as Inference Engines, Semantic Analysis Engine, Guidance Engine, Semantic Search, Semantic Extraction Engine, Action Recognition and Simulation and Context Capture Plugins.

Key Components

Workspace: Native Systems, Sidebar and Context Capture Plugins

The Workspace consists of the environment in which the user performs his knowledge work, interacting with multiple applications (which we call Native Systems) and a UI that provides activity context aware assistance.

The Native Systems are software systems, including operating systems, productivity suites, statistics software, software development tools, etc., used by the knowledge workers in their regular course of work. These systems can
be divided into two types depending on their domain specificity: *Generic Productivity Tools*, which are applications intended for generic activities in different domains (e.g. text editing tools, spreadsheet tools, e-mail clients, browsers, etc.), and *Domain Specific Work Tools*, which are applications created for one purpose and one particular domain (e.g. patient admission tool in a hospital system, retail management software for a supermarket, etc.).

In order to allow the Context-Aware System to interact with the work environment, the System communicates with a set of *Context Capture Plugins* built for existing tools to capture and transfer structured information about the actions the user is taking (e.g. the user is typing, clicking a button or touching the screen) and the application context around it (e.g. the user is typing in a field meant for possible diseases), to the *Workspace Manager*.

For providing assistance, e.g., offering suggestions to the knowledge worker, showing information regarding the context or the process of the task, etc., the System incorporates maintains a *Sidebar UI* for a Native System.

This whole workspace is controlled by a set of application specific *Presentation Managers*, which communicate with the Context Capture Plugins as well as with the Sidebar UI, to receive information and guidance from the *Workspace Manager*, calculate an optimal time, opportunity and granularity for the information to be presented and then passes it to the Sidebar or Context Capture Plugins in the Workspace.

**Context Working Memory and Workspace Manager**

During knowledge work, the System captures, stores, and replays the context of the activity being performed. During this performance, all this context information is stored in a structured form inside the *Context Working Memory* and made available to other modules to access and modify via a publish and subscribe protocol.

The *Context Working Memory* contains the following information:

- Initially, it is loaded with the important context information from the task at hand and previous activities (e.g. which activities were performed, when, what are the output artifacts of previous activities, etc.)
- During the execution, it contains procedural and contextual information from the Native Systems captured by the Context Capture Plugins and sent to the *Workspace Manager*
- It also contains procedural and contextual information from the Sidebar UI, sent and handled by the *Workspace Manager*.

![Figure 10. Proposed Architecture of Activity Context Aware Workspace](image-url)
• Contains structured information about what task is being done, what stage is it at and all the information related to the task that has been generated and used so far.
• Procedural information of the internal actions and decisions of the Context-Aware System.
• Contextual information about the user, usage, previous performed activities, community information, which can be useful for the performance of the current activity, is pulled in from the Episodic Memory.

The Workspace Manager, which is also called the Context Manager for its direct role in capturing and saving the activity context in the Working Memory, performs a crucial role in the Context-Aware System:

- Receives the context captured from the Context Capture Plugins and the Sidebar and saves it in the Context Working Memory.
- Receives the guidance from the Guidance Manager and transfers it to the Presentation Managers for it to be presented in the Native Systems or the Sidebar.

Activity Manager: Guidance Engine, Action Recognizer and Action Simulator

The Activity Manager comprises three different modules: the Action Recognizer, the Action Simulator and the Guidance Engine. It keeps track of the activity being performed, recognizes the actions being detected by the Workspace Manager and the activity to which they belong. It simulates this activity and generates the appropriate guidance for the user.

The Action Recognizer subscribes to the Working Memory for any new raw information about the actions performed and their context. When a new context capture is available (e.g. a button with name ‘Order Tests’ has been clicked in the work environment), the Activity Recognizer fetches the Activity Model and relevant section of the Domain Ontology, as well as the context stored in the Working Memory regarding previous activities performed and recognized. With this information and by using a combination of pattern matching and entity class matching, it recognizes the activity and the action within the activities, which are being performed. After the raw information is identified with an activity, it is structured according to the action structure and stored back into the Working Memory, after deleting the raw information.

Guidance Engine subscribes to the Working Memory for discovering new actions and activities being recognized. If an action identified and saved back by the Action Recognizer is found, this module initiates the guidance through the activity, trying to determine what will be the future actions, the information needs of those actions, where and how to fulfill those needs, and how to process this information.

For this purpose, the Guidance Engine sends the information about recognized action to the Action Simulator. It also sends the past actions stored in the Working Memory, information from relevant tasks from episodic memory, and requests the simulation of the activity to determine the next actions to be performed by the knowledge worker.

At this point, the Action Simulator pulls the information about the activity structure and the important entities and relationships from the Activity Knowledge Base and the Domain Ontology, and tries to infer the next action to perform given the current action passed by the Guidance Engine. Using the existing entities and relationships, it analyzes the information that is already present in the context (from Working Memory) and finds out what are the information needs for the activity.

Once these information needs are identified, the Action Simulator queries system resources including Activity Knowledge Base to determine information sources that might satisfy the information needs. After collecting this information, the Action Simulator returns it back to the Guidance Engine.

As an example, let us imagine a patient care scenario, where the knowledge worker, a doctor, accesses the personal data of his patient, his history, the signs and symptoms which were detected during the initial examination, etc., and tries to identify the set of possible diseases that the patient might have. When the user clicks on a button for ordering tests, this unstructured action is saved in the Context Working Memory and given structure by the Action Recognizer. The Guidance Engine asks the Action Simulator to simulate the next action to be done (ordering tests) and the information needs for this (types of tests to perform). The Action Simulator queries the Activity Knowledge Base to find out where to find this information (domain ontology) and how, using the relationship disease-tests existent in the domain ontology, where disease is the set of possible diseases listed by the doctor in the previous activity).

The Guidance Engine receives the information which has been fetched by the Action Simulator and does additional processing needed to complete the guidance (calculation, evaluation, inference, semantic search, decision-making, etc.). Also, if there is any information from past instances
needed for this additional computation, the Guidance Engine fetches it from the Working Memory. The information resulting from the simulation and guidance modules is stored in the Working Memory.

The Workspace Manager, subscribes to the Working Memory for updates from the Guidance Engine, identifies the new guidance elements waiting to be provided to the user, and sends them back to the Presentation Manager.

**Activity Knowledge Base**

The Activity Knowledge Base contains the description of the task that is to be done, its structure, its division into different activities, the actions within each activity, the entities and relationships and the system resources that can be used to accomplish the outcomes of the activities/task.

The main part of this KB is the **Activity Structure**, which describes a four-level organization of the system. These levels are, from more general to more specific, top level to the lowest level, *task, activity, action and operation*.

The highest level of this organization is the *task* which this workspace is design to fulfill. We define this task as a sequence of activities structured like a *directed acyclic graph*.

An *activity* can be defined as a procedure necessary to reach a goal within a specific context and produce a certain outcome using a determinate incoming information set. An activity is uniquely described by its context (consisting of the types of incoming information, newly produced information and outcome information elements) and by its workspace model, which specifies all the existing UI elements in the Workspace, which are a combination of the Native Systems and the Sidebar, along with the context of these elements (e.g. a certain text field within the activity of collecting a “Set of Possible Diseases”, is specifically dedicated to accept a string element from the “Disease” class of information). Finally, an activity is also associated with a set of actions that are executed by the user to reach the activity outcome.

An *action* is a decomposition of activities in a sequence of 1) atomic operations executed on the UI elements by the user, and 2) points at which the system can offer assistance or guidance to the user. An action is context dependent and has semantic content e.g. an action is not merely to type a string in a text field, but to type a string representing the name of an instance of the class “Disease” in a text field during the activity “Set of Possible Diseases.”

Finally, *operations* are the lowest level atomic events that can be performed in each of the UI elements and modify their state. For example, a click on an unfocused text field is an operation which will change the text field state from unfocused to a state of focus. Operations are context independent and devoid of semantic content, only related to the element on which they happen.

By capturing the operations and the context in which they occur (which is the function of the Context Capture Plugins), the Action Recognizer is able to perform a pattern -match and identify the action being performed. Afterwards, using previously recognized actions already stored in the Working Memory, the Action Recognizer can also identify the activity that the user is doing and what is expected to be the next sequence of actions in order to accomplish the activity’s goal.

An important part of the Activity Knowledge Base is the **System Resources and Tools** knowledge base, which relates each activity and its important information elements to the tools that can be used to fulfill those needs, in order to create the guidance needed at the assistance points identified in each action. These tools are directly related to the information needs of each activity, which are detailed in an **Important Entities and Relationships** knowledge base, which encodes the strong dependency between the relevance of the relationships and the context and activity to which they belong. For example, the relationship between an element from the class “Symptom” and an element from the class “Medicine” can be a relationship of the type “Medicine treats Symptom” or from the type “Medicine can-cause-secondary-effect Symptom”. The first relationship is the one particularly useful for the patient care activity of “Decide Treatment Plan”, while the secondary relationship can be useful during both the “Diagnosis” and the “Follow-up” activities. These two knowledge bases use elements described in the **Domain Ontology**.

**Knowledge-Base and Domain Ontology**

The architecture includes a knowledge-base which includes a domain ontology that specifies the domain concepts. The elements of activity model are described using the terms available in the domain knowledge base and a common sense knowledge base describing relevant general concepts. The knowledge base gets updated as the generalizations and abstractions from episodic memory are included in it.
Episodic Memory and Semantic Memory

The Episodic Memory serves as the archive of information created during episodes of activity. The information is archived along with a detailed trace of the activity context surrounding its creation. In addition, the information is semantically indexed using the concepts in domain ontology and activity models.

The Generalization and Abstraction Engine identifies candidates for transfer episodic memory (Shastri In Revision) to semantic memory (common sense knowledge base and domain knowledge base).

Inference Engine

The inference engine forms the core reasoning system for the entire activity context-aware workspace. We use a semantic extraction and inference method that extracts and encodes a large number of entities, categories, taxonomies, facts, first-order rules, and utilities, answers queries rapidly by backward reasoning, responds to new inputs (facts) by finding explanations (abductive reasoning), making predictions, instantiating goals and identifying actions that are likely to enhance expected future utility. This approach (Shastri and Ajjanagadde 1993) leverages understanding of how the brain might support scalable, real-time reasoning to design a scalable, high-performance inference system.

Semantic Extraction Engine and Classification Engine

The semantic extraction engine (Shastri, Parvathy et al. 2010) extracts the following from natural language input, after lexical pre-processing and syntactic parsing: Entities, Entity Types, Facts, Events and role-fillers in the event (who, what, whom, where, when), and relationships such as cause, effect, purpose etc. The method combines statistical techniques (term frequency analysis, statistical parsing), classical natural language understanding techniques and probabilistic inference with specific forms of semantic and discourse analysis, using a knowledge repository that includes a domain-specific lexicon, domain ontology, common sense rules, idioms, slang and SMSese, while enabling bridging inferences and embodied semantics. This engine produces semantic frames, thus identifying relationships, facts and events. This creation of semantic frames is aided by linking rules that map the syntactic structure of a sentence to a semantic frame that constitutes meaning of a sentence. The antecedents of a linking rule are syntactic structures and its consequents are semantic roles in semantic frames. The system uses common sense reasoning to understand productive metaphors within limited domains and to resolve some forms of ambiguities. The components of this system are largely domain-independent and the system can be extended to new domains by simply adding domain-specific lexicons and ontologies.

External and Internal Knowledge Source Wrappers, Multi-Device Delivery Servers

Multiple application wrappers are provided for connection to internal enterprise databases, external sources of structured and unstructured text and media, search engines, domain specific knowledge sources, public information sources and social-media data. To provide context-aware services across devices we provide multi-device servers that deliver device context sensitive granular UI and data.

Demonstration of Architecture Concepts and Principles

In this section, we describe the two use cases developed to demonstrate the previously described architecture. In both implementations we show the high value of a Context-Aware system to keep track of the knowledge worker activity, captures its context, and generates guidance and assistance that helps the user in the performance of knowledge work.

Patient Care

The healthcare domain has a very rich set of needs demanding context aware systems, due to the large amount and multiple types of information that the knowledge worker has to have access to, in context, including general medical knowledge, tacit knowledge based on experience and prior episodes, information regarding the patient(s), past information of patient(s), information about how the current activity is being performed, and also related information such as similar cases from the past, knowledge from other doctors from the community, demographics, ongoing outbreak information, news, etc.

The Demonstration of Concept (DoC) that we present here implements the complete task of patient care, from the detection of signs and symptoms and the research about the history of the patient, all the way to creation of a follow up plan, passing through identification of possible diseases, diagnosis, tests, analysis of test results and treatment plan. This DoC implements a User Interface which communicates with the Presentation Manager, with behavior similar to the Context Capture Plugins and Sidebar that may be used together with Native Systems used by the hospital.
The task starts when the patient reaches the hospital. The patient’s data is included in the Native Administrative System of the hospital by the administration staff, as well as the initial symptoms detected in the initial examination.

When the doctor logs in, he can see his Inbox (Figure 11), which displays all the patients that are assigned to him. Here shows up the first assistance point: The system estimates which patients require more immediate attention and highlights those patients’ details to the doctor. The doctor selects this patient and proceeds to identify a “Possible Set of Diseases”.

The Workspace is divided into three sections, the Initial Information Panel, containing the information regarding past activities (e.g. the recollection of signs and symptoms and the history of the patient), the Action Panel, where the knowledge worker can perform the current activity (e.g. identify a list of possible diseases), and the Assistance Panel, were assistance generated by the system is displayed (e.g. suggestions of possible diseases that might match the set of signs and symptoms).

When the “Possible Set of Diseases” activity is initiated, the Guidance Engine with the help of the Action Recognizer and Simulator identifies the user’s intention to identify possible diseases that are consistent with the signs and symptoms, so the system searches if there are any possible resources to fulfill that information need and generates a list of recommendations that display in the Assistance Panel.

At this stage, the information available to the Guidance Engine from the Activity Knowledge Base in its System Resources and Tools database establishes that there are two possibilities for finding possible diseases in this simple model: Consistency and coincidence of the signs and symptoms with the diseases present in the Domain Ontology, or using external sources, articles and news to relate the travel history of the patient to possible outbreaks of disease (Figure 12).

Internally, this particular activity (“Possible Set of Diseases”) is described as a state machine, where the global state of the Workspace is the combination of the states of all the elements in the Workspace Model. Transitions between states occur when an operation is performed and the state of one of the elements changes. The goal is also described as a combination of states (Figure 13 – please see details in Supp. Figs. 1-3).

Once the environment and the initial assistance (Semantic Suggestions Assistance) is generated after triggering by activity recognition, the knowledge worker can perform
multiple actions to accomplish the goal of filling the list of possible diseases. Each text field can be filled by typing the name of a disease, by typing some characters and selecting one of the Lexical Suggestions that are offered in a dropdown element, or drag and drop any of the Semantic Suggestions offered in the Assistance Panel (Figure 14, Previous Page).

When inserting the string by typing, after a few characters are entered in the text field, an assistant creates a list of Lexical Suggestions from the Domain Ontology, offering to the user in a dropdown a set of disease names that are similar to the substring entered in the field. If the knowledge worker keeps on typing without selecting one of the suggestions, another assistant does Lexical Matching and displays an error message in case that the string doesn’t belong to the class of a disease name. If it does, a third assistant checks the consistency of the disease in the text field with the signs and symptoms recorded during the previous activities (Figure 15).

The Semantic Matching assistant calculates a relative score using the signs, symptoms and history data available and shows the resulting measure in green (accepted) if this measure is above a minimum threshold or in yellow if it is below, giving the user the option of accepting this disease or rejecting it (Figure 16).

After at least one of the text fields is filled and accepted, the final goal of the “Possible Set of Diseases” activity has been reached and it is possible to step forward to the next activity. The button “Order Tests” allows us to go to the next screen, where the next activity will be recognized and initiated.

“Order Tests” activity is also described as a transition of states and points of assistance. One of these assistance points is initiated at the very starting point of the activity, at which the possible diseases identified by the doctor are used for querying the Domain Ontology for a set of recommended tests that the doctor can select to test his hypotheses. Another assistance point is a simple visualization functionality that helps the doctor to identifying which tests are related to each disease by highlighting these tests when the user hovers on any of the diseases (Figure 17).

We have developed similar assistant modules in the Guidance Engine for the next action of the patient care task, “Examination of Test Results.” After the tests are selected and ordered, and test results come back, the index page highlights the patient informing the doctor that the test results are back. Clicking on each performed test, the test results are shown in green if they are inside the normal ranges and in red when the results are out of normal measures. Each disease in the Assistance Panel is shown associated with its particular distinguishing tests and another relative score is calculated by using the data obtained by testing (Figure 18, Next Page).

Another interesting function is provided by the exploration and analysis of the Episodic Memory by considering...
previous episodes of the same or other users to improve the relative scores and guidance provided by the system. In the example shown, the patient named Charles Dent presenting a certain set of signs and symptoms had been diagnosed with Swine Flu using his travel history information and pulling related data from external sources (news, disease outbreaks, etc.).

The next patient, Clark Edwards, present signs and symptoms similar to Charles Dent, with similar travel history. Considering these factors, the relative score computation and guidance is modified based on the disease diagnosed for the previous patient (Swine Flu) and the choices made by the doctor for that diagnosis (Figure 19).

This application demonstrates the value added by the architecture described in the previous section with the capture of activity context, its analysis by the Action Recognizer, the assistance offered by the Guidance Engine and the Action Simulator as assistance points, and the use of the Episodic Memory to retrieve and compute the information obtained during previous cases.

**Assisted Living**

While the architecture described above can be used to build systems for supporting very complex knowledge-works, like the Patient Care use case just explained, such systems pose highly complex design challenges. We also present another use case of moderate complexity, an Assisted Living (AL) service. In this scenario (Laha, Shastri et al. 2012), the Assisted Person (AP) is an aged person who resides at her own house and is willing to take help in form of reminders to take medication, exercise, etc., and suitable interventions in case of health-related emergencies and contingencies.

The AP subscribes to the AL service and uses a personal device (mobile phone, smartphone, or a specialized device that can be a wristwatch with GPS and some sort of display screen) in order to be tracked, sent reminders or contacted by the “Service Managers.” In this particular case, we implement a solution with a Blackberry phone as AP’s personal device, but the solution is applicable to a large majority of feature-phones and smartphones.
emergency situation e.g., the subscriber had a fall (detected based on the analysis of accelerometer data) or some contingency, say, the subscriber is away from home - has traveled to another part of the city to pay a visit to a friend - the system shifts into a contingency mode where the service managers’ judgment and direction is needed to respond properly.

In this Demonstration of Concept, we implemented the latter case, in which the AP is detected to be far away from home and close to her medicine time, and this triggers the contingency management activity, in which the Service Manager contacts the AP to ask whether she has her medicines with her.
If the AP replies that she doesn’t, the Supervisor contacts her again to sort out the estimated location (Figure 22, Previous Page) for the time in which the medicine has to be taken. When the future location is sorted out, the system assists the Service Supervisor to find out best combination of medical stores and delivery services nearby which can provide the AP with her medicine on time (Figure 23, Previous Page).

Conclusions and Future Work

Thus, we have described the manifesto for an activity context aware digital workspace, and a working architecture demonstrated in implementations of a Patient Care System and Assisted Living System.

We invite the research community to provide feedback and critical reviews of this proposal and research work, to help define further work in this important area.

Acknowledgement. This work was supported at Technology Research Labs at Infosys Labs, by Infosys Limited.

References


McQuade, E., E. Sjoer, et al. (2007). "Will you miss me when I'm gone?: A study of the potential loss of company knowledge and expertise as employees retire." Journal of European Industrial Training 31(9): 758-768.


