Supporting Interaction with the Internet of Things across Objects, Time and Space

Fahim Kawsar, Gerd Kortuem, Bashar Altakrouri
School of Computing and Communications, Lancaster University, UK
{kawsar, kortuem ,b.altakrouri}@comp.lancs.ac.uk

***Abstract*— Effectively assisting people in complex and highly dynamic work environment requires advances in high-level declarative activity models that can describe the flow of human work activities and their intended outcomes, as well as novel interaction approach for distributing and coordinating information across physical objects, time and space. This paper describes a novel modeling technique for pervasive systems based on high-level models of human activities, so-called *situated flows* and presents a corresponding flow driven distributed software framework. This framework provides the foundation for discovery, adaptation and execution of flows in real time matching the dynamics of real world activity and consequently drives the pervasive interaction using push interfaces to facilitate a synergic coordinated interaction experience in dynamic work environments.**

Keywords – Workflow, Pervasive Interaction, Cooperative Physcial Objects, and Instrumented Interactive Space.

I. INTRODUCTION

Over the past years, research in pervasive computing has demonstrated the potential of context-aware and proactive technologies for improving human work performance. Examples include systems for tracking and automatically recording task performance in industrial maintenance scenarios [1,2], measuring and informing workers about their exposure to equipment vibrations [3] and context-aware information capture and presentation at hospitals [4,5]. Pervasive work support systems make use of sensing infrastructure and self describing physical objects augmented with digital resources (so called *Internet of Things*) along with handheld and wearable terminals to analyze work activities in real-time and to provide users with relevant and timely information pertaining to their work. Projecting in the future we can imagine that work environments will be densely instrumented, be able to understand minute details of work activities and will actively assist users in attaining their objectives.

Although there have been significant advances in sensing, as well as rich mobile terminals that are capable of providing seamless interaction experience with Internet of Things, great challenges for the development of pervasive work support systems still remain: the first challenge relates to the lack of technology-independent and transferable models of human work activities. Activity recognition approaches are driven from the bottom up and use models that are highly dependent on algorithms and technologies; they are not suitable as *declarative* modeling tool for specifying organizational work processes. Workflow technologies based on BPEL [6] and other languages provide an interesting starting point for the development of declarative activity models, yet existing approaches lack features to express physical context (location etc) and are not suited for integration with activity recognition technologies.

The second challenge relates to the design of distributed and embedded interaction techniques and user interfaces to effectively support people in demanding work environments such as hospitals and industrial plants ensuring spatial and temporal consistency. There has been much progress on mobile and wearable device interfaces however, the question of how to distribute information in a physical environment across time and space considering user context and work processes with the goal to maximize human performance has not yet been tackled. A complementary aspect of this distribution is the synergic coordination of physical work objects distributed spatially that can lead to consistent and seamless user experience. Thus, pervasive interaction mechanism across physical objects, time and space in a situated fashion is at the core of future pervasive work support systems.

In this paper we address these two challenges using a high-level models of human activities, so-called *situated flows* and a corresponding distributed software framework. Situated flows model human work processes as a set of physical *actions* glued together by a plan (a set of transitions), which defines how activities should or could be performed to achieve a specific outcome. In contrast to traditional workflows, situated flows are *situated* and *context-aware*: they are linked to physical entities like equipment and people, moving with them through different environments, thereby reacting to and being influenced by their context. We use situated flows to distribute work process and guidance information across connected physical work objects. A corresponding software framework provides the foundation for the flow driven interaction and the physical object coordination in the dynamic workplace by facilitating discovery, association, adaptation, and execution of the flows in real time. In addition the framework drives the presentation and distribution of work process information: mobile terminals allow people to uncover flows and task information embedded in the surrounding physical environment. This provides them with a seamless user experience ensuring spatial and temporal coordination of the physical objects and interaction consistency. Consequently, the contributions of the paper are two-fold, i) situated flow, a novel approach to model high-level human activities that can drive seamless interaction in pervasive work environment, and ii) a distributed software framework that provides the foundation for flow driven interaction, and synergic co-ordination across physical connected objects.

In the next section, we present situated flow and illustrate how it can be used in modeling human activities. Then we discuss the design issues for a flow-driven software framework for supporting coordinated interaction experience in a pervasive work environment followed by the technical details of the framework. Next, we proceed to the feasibility of our approach by demonstrating a scenario-based evaluation. Then we discuss a range of issues uncovered during our experience with the system. Finally, we position our research with respect to the related work and conclude the paper.

II. SITUATED FLOWS: A MODEL FOR PHSYICAL OBJECTS IN PERVASIVE WORK SPACES

People are often involved in complex real world activities that require coordinated use of multiple physical objects distributed across different locations. To support coordination among physical objects ensuring a seamless user experience and maintaining interaction and interface consistency across multiple physical objects, a suitable modeling technique is required that can express the inter-object interaction relationship with respect to the human activity in context. Situated flow is one modeling technique that can address this aspect of coordination across multiple physical objects.

A situated flow (*flow*, for short) is a high-level programming language for modeling real-life processes and human activities. It consists of a set of *actions* glued together by a plan (a set of *transitions*), which defines how actions should be performed to achieve some goal under a set of constraints [7,8,9]. Situated flows are closely related to classical workflows [6], yet they differ in that they are situated and context-aware: they are (logically or physically) linked to real-world entities like people and objects, being carried by them or moving with them through physical environments. Flows are executed in parallel to the real-world actions they describe: when an action that is described in a flow is performed in the real world (by a person), the flow progresses one step. From a human interaction point of view, situated flows define *opportunities for action*. A flow embedded in an object defines tasks and actions that can or should be performed *with the object* or *in the room*.

By attaching flows to physical objects that describe the actions and opportunistic assistances to guide those actions, we can build an interactive space where human interaction with multiple physical objects can be supported in a cooperative fashion. Situated flows distributed across multiple objects have an implicit understanding of each other and thus can contribute in shaping the behavior of the physical object which in turn influence the overall human activity that span across time, space and objects. In Figure 1, we present a nursing home scenario where situated flows that model the daily rounds of a nurse in a hospital are shown, the daily care schedule of a patient and the operation procedures or handling instructions of various pieces of medical equipment are modeled in a number of flows and are attached to real world entities. The nurse's flow defines the sequence of activities she need to perform during her morning duties that includes supporting morning hygiene routine of five patients and documenting patients physical condition. Each patient also has a flow, depending on his/her medical condition. The nurse's flow does not define a strict patient order, so the nurse can select the patients in an arbitrary fashion. Some of the medical equipments used during this routine also have flows attached that specify the handling procedure or other situational information that has to be addressed during the use.

Flows can be combined dynamically to form new flows in a context-dependent manner (see Section IV.B.2.c). In the next section, we discuss the design issues addressed in our flow driven framework.

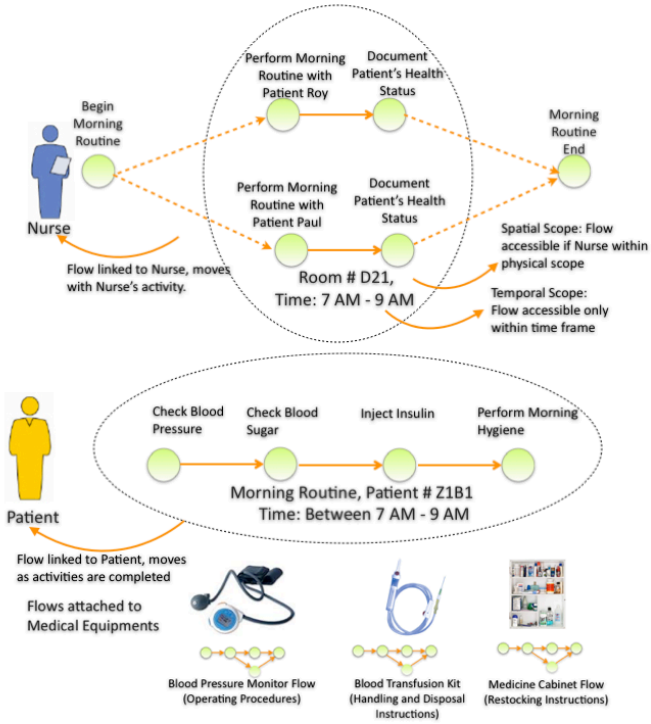

Figure 1. Situated flows in Nursing Home Scenario. Patient flow, nurse flow (Personal Flows) and three equipment flows (Object Flows).

III. DESIGN ISSUES

In this section, we first walk through a nursing home scenario to illustrate the flow driven coordinated interaction across multiple physical objects, time and space to support complex real world activity. This scenario is extracted from a feasibility study with nurses supporting Dementia and Alzheimer's patients in the Mainkofen Hospital, Germany. Based on this scenario, we elicit the design challenges and explain the design decisions to meet those challenges.

"Alice is a nurse and is responsible for taking care of five dementia patients in her designated ward. Every morning, Alice starts her day by picking her personal flow by touching her mobile phone on the surface of the computer (augmented with an RFID reader) located in the ward's main office. The computer identifies Alice and the corresponding flow is pushed to her mobile phone. The flow explicitly specifies her daily tasks for the ward involving five patients. Once she enters the ward, she decides to support patient Bob first as there is no strict patient order imposed in her flow today. She moves forward towards Bob's bed, and her mobile phone picks up the Bob's flow defined by Bob's doctor. Alice's flow is now automatically refined through composition of Bob's flow into her flow to reflect the list of tasks that Alice should

perform with Bob. The phone interface is also changed to support the care service to Bob. She starts the care service by checking his blood pressure and blood sugar level using a blood pressure monitor and a glucose meter respectively and documents them properly using the phone interface. Then Alice needs to give an insulin injection to Bob as defined by his flow. The injection needle has a flow attached to it specifying the insulin unit and instructions for the injection. This flow is automatically picked up and merged within Bob's flow and consequently within Alice's flow. Thus Alice's flow and associated medical equipments assist her to perform assigned duties precisely. After that Alice helps Bob with his morning hygiene routine (e.g., shower, toilet, etc) before moving to her next patient."

The above scenario exposes two design issues for a flow driven interactive space supporting coordination across multiple distributed physical objects, which we discuss next.

A. *Design Issues for the Coordination across Physical Objects with Situated Flows*

As discussed in the earlier section, a situated flow provides the foundation for the dynamic interaction and coordination among the physical objects in an instrumented environment. Situated flows are attached to connected physical objects, and create a synergy among these objects through their runtime execution. As exposed in the above scenario, there are static and runtime aspects of the flows.

1) Static Issues with Flow: There are two static aspects:

a) Flow Generation: The first design concern is the generation of the situated flows. As mentioend in the earlier section a situated flow defines a collection of actions and states glued together with a plan. Usually these flows are predefined and generated ahead of the execution. However, these actions and states are not rigid in a sense that, with appropriate refinement (explained later) they can adapt to address the situation in context. To support such dynamics we have adopted a loosley structured flow representation mechanism, where a template based model is used to generate the actual flow in execution (see Section IV).

b) Flow Distribution: The second design concern is the distribtuion of the flows, specially across multiple entities. As described in the scenario, flows are dispersed across multiple real world entities, e.g., a nurse, a patient, a medical equipment, etc. Flows attached to physical objects and persons are distributed ahead of the execution. In our design this distribution is currently managed logically, i.e., linking the real world entity to a flow repository that provides the appropriate flow for the entity in context.

2) Runtime Issues with Flow: There are four runtime aspects as shown in Figure 2:

a) Flow Discovery: The first runtime issue is how a flow is discovered spatially. Flows are logically linked to real world entities, e.g., a medical equipment, a nurse, etc., When these entitities come in proximity to each other spatially, their respective flows are discovered by each other. Currently, our design decision is to use a mobile terminal that acts as a flow explorer. When the terminal comes in a designated location equipped with multiple flows (both personal and object flows), the terminal discovers and accesses the flows. As a triggering mechanism, two approaches can be used: i) 2D Marker: a flow is logically linked to a 2D marker and a camera equipped mobile terminal can pick the flow by scanning the marker, ii) An RFID Tag: a flow is logically linked to an RFID Tag and an RFID reader equipped mobile terminal can pick the flow by touching the tag.

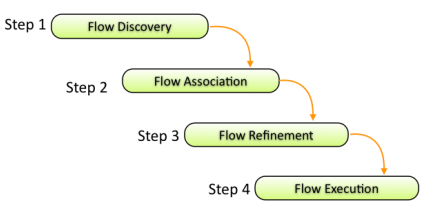

Figure 2. Lifecycle of Situated Flows.

b) Flow Association: In a designated environment, there could be many physical connected objects and/or people. Accordingly, there could be numerous combinations of activities and sequences of activities that can be performed. It is thus important to differentiate the flows according to the relevancy of the tasks. For example, in our scenario the ward might have more than five patients assigned to Alice, in this case Alice's flow should only incorporate the activities relevant to her patients and should associate only those patients' flows in her flow. Similarly, the medication tray for patient Bob might be different from other patients. Thus the flow attached to the patient's medication tray has to be associated properly. Currently, we have addressed this issue in our flow representation through predefined flow identity. Each flow before and during its execution, clearly knows which of the flows it needs to be associated with.

c) Flow Refinement: Another important aspect closely related to the association is the flow refinement. By refinement, we mean that a certain activity in the flow can be expanded to compose another flow to adapt to the situation at hand. For example, in our scenario Alice's flow is refined when she approaches Bob by composing Bob's flow in her flow, similarly the injection flow was merged to Bob's flow (and Alice's flow) when Alice starts using the injection. This refinement can be performed dynamically or statically. For static refinement once a flow is discovered, associated flows are discovered logically and refined. For dynamic refinement, when a certain activity is reached in the flow, possible refinements are performed considering the activity and associated flows. Currently, both refinement options are supported in our framework. The refinement techniques will be discussed in detail in the next section.

d) Flow Execution : The final issue with the flow is the actual execution of the flow. As mentioned above our flow representation contains descriptions of a sequence of activities, and corresponding assistances. Once a certain activity is performed, the flow moves to the next activity carrying the results of the activity enabling the entities involed in the next action to utilize them. Consequently, this execution plays a crucial role in providing the coordination across multiple physical objects, as it is the flow that carries the semantic messages across objects distributed spatially ensuring a cooperative behaviour. Ideally, the advancement of the flow should be done implicitly through rigorous context

recognition (e.g., activity etc.). However in our current framework we have taken a relaxed approach of using location context only (recognized during flow discovery) and statically mapping the flow state in a mobile terminal and allowing a human user to interact with the terminal to move the flow forward.

B. *Design Issues with Dynamic Interfaces of the Connected Physical Objects*

In the previous subsection, we have discussed how a situated flow models the activity and associated physical objects coordination. Strong design elements in our software framework are the interfaces of these connected objects and interaction techniques that enable a person to interact with these objects and their associated flows. In essence there is no specific interaction technique involved in our flow driven interactive space. Persons carrying flows in a mobile terminal are supported to naturally carry out real world activities with or without another real world entity (e.g., another person, a physical object, etc.). However, during the activity and the execution of the corresponding state of the flow, the mobile terminal is updated with a relevant interface in response to the activity and the physical objects in context. This enables a peripheral awareness as well as an ambient guidance for the person performing the activity to have assistance when needed. The dynamic update of the interface is managed applying a remote interface mechanism. A flow state contains a link to the remote interface and the corresponding interface is pushed to the mobile terminal executing the flow. In the next section, we will discuss the technical detail of our flow driven software framework addressing these design issues.

IV. ARCHITECTURE

In this section, we first discuss the representation of the situated flow, and then we move to the technical detail of the software architecture that implements the flow driven co-ordination framework for interactive spaces.

A. *Representation of the Situated Flow*

A situated flow is represented in a document that defines the sequences of activities, their dependencies, and minute details of the activities including the interface and guidance information. In our current implementation this document is represented in an XML file with a number of blocks (Figure 3). These are:

- *Identity Block:* This part of the document specifies the ID and type of the flow. This ID plays a crucial role in the flow refinement process.
- *Meta Information Block:* This block describes the purpose and owner of the flow. This block can be extended to address the security aspects.
- *Context Block:* This block specifies the static context associated with the flow. A flow can only be executed if it is in the defined context. Currently only location is used in this block.
- *Activity Block:* This block specifies the collection of activities to be performed to complete the flow.

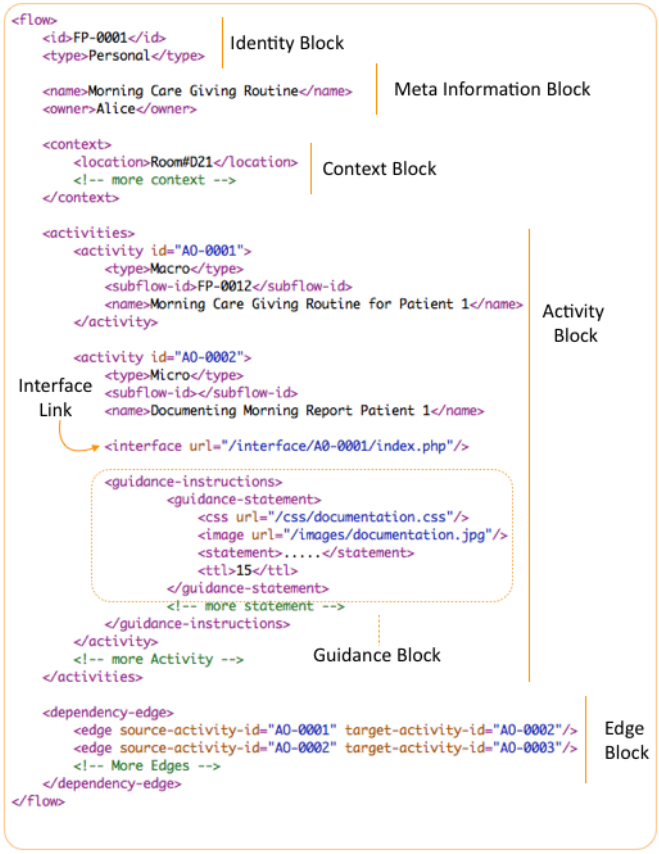

Figure 3. A sample representation of a Situated Flow.

Activities can be of two types:

- *Micro Activity:* This type of activity is not composable, so a flow cannot be refined on this activity.
- *Macro Activity:* This type of activity is composable and contains a link to another flow. During flow association (static refinement) or execution (dynamic refinement), this activity is replaced with the linked flow's activity or sequences of activities

Each activity contains a link to a remote interface that is pushed to the terminal executing the activity. In addition each activity contains a guidance block that states the assistive instructions offered to the person performing the activity.

- *Edge Block:* This final block specifies the dependency across the activities contained in the flow, i.e., the order of the activities to be performed.

In the next subsection, we present the architecture and technical details of the implementation.

B. *Description of the Architecture*

Figure 4 depicts the basic architecture of the flow-driven software framework. The architecture follows a typical client-server approach, where a thin-client is used as a triggering device that initiates the interaction and performs the execution of the flow through communication with a flow server. The communication across the system is implemented using HTTP-XML in a RESTful manner. This ensures the lightness,

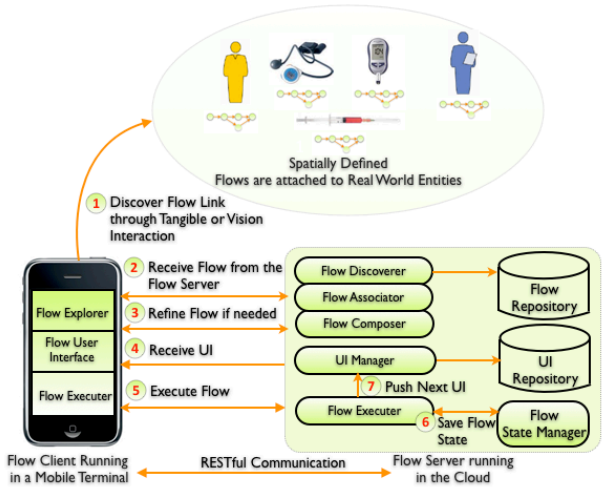

Figure 4. Architecture of the Flow Driven Framework.

scalability and interoperability of the approach. In the following we first discuss the technical details of the thin-client before moving to the technical details of the flow server.

1) Thin Mobile Client*:* The basic functionality of the mobile client is to discover the spatially defined flows and then presenting users with an appropriate user interface to move the flow forward.

a) Discovering a Flow: As discussed in our design, flows are spatially defined and have a physical scope. They can be attached to real world entities, e.g., a patient, a ward door, a medical equipment etc. Once a user carrying the mobile terminal comes in the vicinity of a flow, the terminal can pick up the flow. Currently in our system, the vision based 2D marker tracking approach is implemented. As a thin client, we have used an iPhone 3GS with OS 3.1.2. To enable real time 2D ID-Marker tracking with a camera, a simplified C++ version of ARToolkit was ported to the iPhone and is used with private Camera Controller APIs of the iPhone SDK with a 12fps refresh rate. Real world entities are tagged with 2D markers, which are tracked by the iPhone by hovering the phone over the tagged entities. A tactile feedback is provided to the user to confirm successful tracking, and consequently the flow (encoded in the 2D marker) is retrieved from the flow server.

b) Executing a Flow: Once a flow is tracked, the flow client communicates with the flow server. The flow server performs the association and composition operations (explained next) and pushes the corresponding user interface to the terminal. Once the actual physical activity is performed, users can provide the results of the activity and mark the completion of the activity by explicitly interacting with the mobile client, thus executing the flow and moving it forward. The flow server remotely pushes this interface. As shown in Figure 4, every flow activity contains a dedicated remote interface link. Currently, this interface is implemented as simple HTML page, and the thin client uses a simplified web browser to present the interface. Figure 5 shows sample user interfaces for the scenario presented earlier.

2) The Flow Server: The flow server is the central component of the framework and provides the building blocks to drive the entire interaction. The flow server is implemented in Java and MySQL is used as the backend persistent storage. The server contains the following modules:

a) Flow Discoverer: This component is a simple web server that accepts requests from a mobile client to retrieve the flow from the flow repository. In addition, it performs a reasoning operation on the flow to decide if any association and composition is needed. If so, it passes the flow to the next component, other wise it passes the control to the User Interface (UI) manager to push the corresponding UI.

b) Flow Associator: This component is responsible for associating the correct flow out of the candidate flows depending on the context and identity of the mobile client. The primary operation involves identification of the macro activity and discovery of the other flows available in the same physical scope that are relevant to the current flow in context. Accordingly, this component passes the control to the next Flow Composer component.

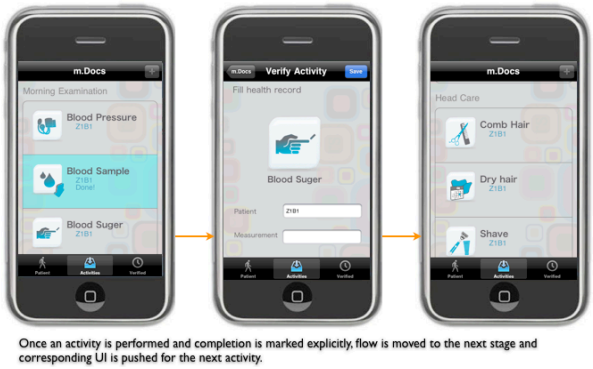

Figure 5. Sample Push Interfaces Running on the Mobile Client.

c) Flow Composer: The flow composer works closely with the Flow Associator. Once a collection of flows is received from the Flow Associator, this component performs the composition accordingly. This composition can be done either statically or dynamically, depending on the mode of operation. This mode is predefined and set globally in the system. For static composition this component performs a Depth First Search (DFS) algorithm on each macro activity of a flow and expands the flow accordingly considering the context (e.g., location, current flow, dependency, and current activity) as shown in Figure 6, before returning the flow to the UI manager component. For dynamic composition, when a flow activity is executed and the next activity is of macro type, this component performs a basic search and traversal operation considering the context and expands the flow accordingly.

d) User Interface (UI) Manager: The functionality of this component is straight forward, depending on the state of the flow, (i.e., current activity), UI manager pushes the corresponding UI to the mobile client after retrieving it from the UI repository. Currently, the UIs are managed as HTML pages with dynamic forms to capture the results of the flow execution.

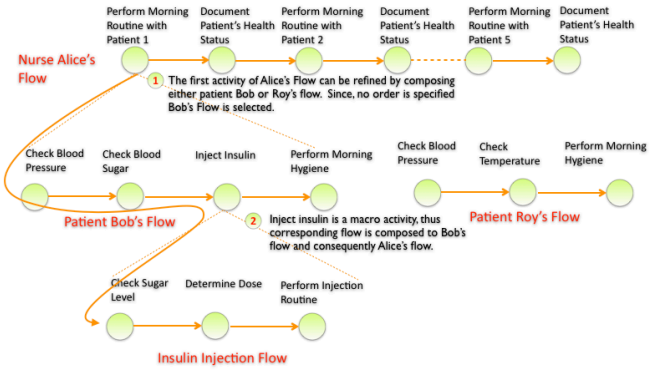

Figure 6. Static Flow Refinement Process.

e) Flow Executor: This component is a crucial component in the framework, as it provides the foundation for the cooperation across physical objects and other real world entities. When a real world activity is performed involving physical objects, the state of the objects and results of the activity is passed to this component by the mobile client, using explicit interaction with the UI pushed by the UI manager. Once received, the Flow Executer extracts these states' results and associates them with corresponding activities and physical objects involved. Then it passes them to the Flow State Manager that saves them in a persistent storage. These results and states are later used when an explicit dependency is tracked in the flow by the Flow Executor and accordingly the new UI state is pushed to assist the activity and interaction providing an implicit coordination across physically dispersed objects.

V. SCENARIO BASED EVALUATION

In this section, we provide a scenario-based evaluation [10] of our flow driven software framework by walking through the scenarios presented in the design section, demonstrating how the different system components work together.

- **Scene # 1**: Alice picks her flow by hovering her iPhone on the ward door tagged with a 2D marker. The iPhone client application sends a flow request to the Flow Discover by sending the Tag ID. Since dynamic refinement option is set globally the Flow Discover returns Alice with her flow as it is and the corresponding UI is pushed to her iPhone (Figure 7).

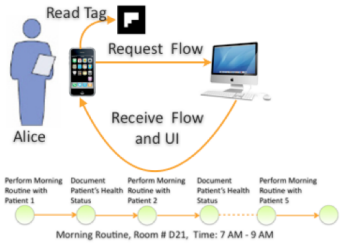

Figure 7. Scene 1 – Flow Discovery.

- **Scene # 2:** There is no strict patient order defined in the flow, so Alice decides to support patient Bob out of her five patients. Alice moves to Bob's bed and reads the 2D marker attached to his bed using her iPhone. The iPhone client application sends the Tag ID to the flow discoverer, and the Flow Discoverer then passes Bob's flow to the Flow Associator which determines that Alice and Bob's flows need to be composed and accordingly passes them to Flow Composer. This component expands Alice's flow to merge Bob's flow and then return the flow to Alice with corresponding UI via the UI Manager (Figure 8).

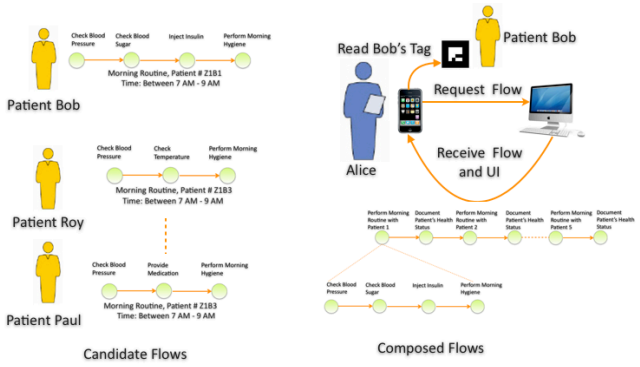

Figure 8. Scene 2 – Flow Refinement (Association and Composition).

- **Scene # 3 and Scene # 4:** Now Alice's iPhone application has an updated UI that shows the first task Alice has to perform with Bob, i.e., measuring the blood pressure with a blood pressure monitor, so Alice checks the blood pressure and records it in her iPhone and marks that she has completed this task by explicitly sending this interaction result to the Flow Server. Flow Executor receives this request, and saves the results via the Flow State Manager and then instructs the UI Manager to push the next UI for the next activity. The next activity is identical except this time Alice needs to check the blood sugar level with a glucose meter. These scenes are shown in Figure 9.

Figure 9. Scene 3 & 4 – Flow Execution.

- **Scene # 5**: The next activity defined is Bob's Flow is to inject insulin using a specified injection. This injection has a flow attached to it. Alice reads the tag of the injection and sends the ID to the Flow Server instructs the Flow Associator and Flow Composer component to merge it firstly with Bob's and consequently with Alice's flows. Accordingly, the UI Manager pushes an updated UI to Alice to execute the flow. During the second activity of this flow, Alice needs to determine the dose of the insulin. When this activity is reached the Flow Executor extracts the previous activity state from the Flow State Manager, and by analyzing the dependency, i.e., interaction with the glucose meter it updates the UI through the UI manager to assist Alice with the appropriate measurement, thus ensuring a smooth coordination across multiple objects (e.g., glucose meter and insulin injection). This is depicted

in Figure 10. When Alice completes the injection operation she marks the end of the activity triggering a request to the Flow Executor, which subsequently removes the injection needle flow and updates Alice's UI through the UI Manager.

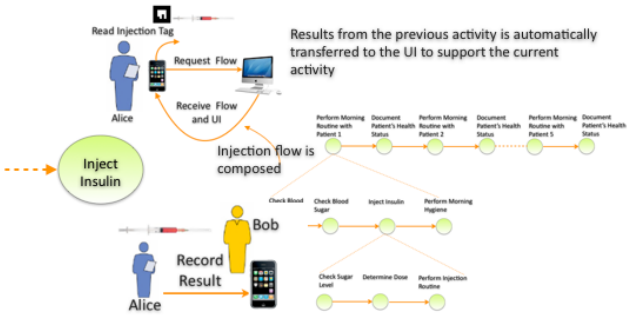

Figure 10. Scene 5 – Flow Refinement and Object Co-ordination.

Essentially, in this section we have shown how our system supports real world activities through discovery, refinement and execution of flows. We have also shown how the system provides the co-ordination across multiple physical objects through Flow Executor and Flow State Manager.

VI. DISCUSSION

In the previous sections, we have presented the primary building blocks of the software framework for a situated flow driven dynamic workplace environment. Our early insights with the system open up a range of interesting research issues for understanding the impact of flow driven pervasive systems in critical workplaces. Some of these issues inline with our future work are discussed below.

Support for Rich Context: Currently, in our system no dedicated sensing infrastructure is used. Flows are attached to real world entities with spatial and temporal scope. These scopes are statically defined during the distribution of flows and are used during the flow refinement phase (e.g., flow association and composition). The flow executor running in the mobile terminal is dedicatedly used to move the flow forward. However, through the integration of rich context information (e.g., activity, dynamic location, etc.) it is possible to execute the flows implicitly without any explicit interaction. Furthermore, rich context like precise activity recognition can determine the quality of the user activity. With such support, if any deviations or errors are identified during an activity execution, suitable notification can be provided to the user for improving the quality of the work. This is particularly important for critical work places like a hospital where strict guidelines and policies are imposed to ensure successful and error free operations. Currently, we are working on integrating a context recognizer in our system that can identify users' actions with physical objects and corresponding objects' states that can be used to move the flow forward implicitly.

Dynamic Refinement Mechanism: Related to the rich context, another import issue is the flow refinement. Currently, depending on the type of activities (macro or micro), a flow is refined by composing additional flows related to the activity. This refinement algorithm uses the static identity and spatial scope of the flow. However, this refinement process should incorporate rich contexts using which system should be able to correctly identify the flows and activity states that need to be refined, i.e., a same flow can be refined in multiple ways depending on the number of contexts available. In such case, the search space for the candidate flows can be minimized and the refinement process can be performed more efficiently. Currently, we are working on defining a dynamic context cost function on top of the context recognizer during the refinement process to improve the filtering and consequent association and composition phases.

Spatial Interface: One fundamental aspect of designing assistive technology for a complex dynamic workspace is a compelling user interface that ensures quality, continuity and usability of information presentation and a seamless user experience. Addressing these interface level issues for a flow driven work place (e.g., in a hospital, a construction site, etc.) introduces considerable design challenges due to i) peoples' primary engagement in perspective physical activities, ii) mobility support, and iii) temporal and spatial values of information. In our system, a mobile terminal is used as a client device that facilitates the discovery and execution of flows through explicit UI. Although, this is a compelling solution and in fact well adopted, many complex real world activities need hands-free operations. For supporting these activities, one solution is to embed the user interface in the physical environment (i.e., physical objects, work surfaces, etc.) and to provide multimodal interaction techniques, like speech, audio, etc. so that people's primary activity is not disrupted. The recent progression of mobile projectors and projector-augmented mobile phones provide an effective solution towards addressing this issue. Fundamentally, mobile projection technology can transfer any physical object and surface into a dynamic information display. In addition, considering the form-factor, these projectors can be wearable thus meet the mobility and hands-free requirement. Consequently, we are working on an alternative mobile client with a spatial interface incorporating wearable mobile projected display with speech and audio support.

State Management: Considering the scalability, lightness and interoperability of the interfaces our solution has adopted a RESTful approach to provide the communication foundation and UI presentation across the system. However, REST is a stateless protocol, which required us to transfer the state of a flow (i.e., activity results and physical objects' states) into the flow server. These results and states play a vital role in providing the coordination across physical objects. However, an alternative solution is to store these states in the mobile client and to run the coordination algorithm locally. We have discarded this alternative primarily for two reasons: i) ensuring a thin client, whose operations are limited to tracking and presenting HTML pages, ii) deploying, extending and modifying new and existing coordination algorithms running on the flow server independently.

VII. RELATED WORK

Situated flow creates direct links between actions in the real world and entities in the pervasive computing system. The high degree of flexibility resulting from the utilization of the flow paradigm enables the design of seamless adaptation and

coordination mechanisms spanning spatially distributed physical objects. We have mentioned earlier that situated flow is adopted from the classical workflow [6] and is normalized by multiplexing two distinguishing properties: context-aware and situated. The corresponding software framework has taken a classical client server approach yet with careful design decisions, e.g., thin client, REST as an underlying communication mechanism, etc. The fundamental difference of this work from the state-of-the art is the use of situated flow as the primary architectural model to ensure coordinated behavior across spatially distributed physical objects.

Existing software infrastructures that facilitate the development of pervasive applications [11,12,13,14,15] tackle some of the individual challenges addressed in this work. However, they do not provide a thorough solution for building human-oriented pervasive applications that are highly adaptive on different time-scales to ensure a consistent and coordinated interaction experience. A key reason for this is that the application model realized by existing infrastructures usually follows the traditional patterns of desktop applications [16]. In this model, the application is started either directly due to a user request or indirectly, e.g. because some predefined condition has been detected. Thereafter, the user interacts with the application until it is no longer needed. While such a mental model of a pervasive application can to some extent support simple and well-defined user tasks, it cannot support more complex tasks consisting of a set of sub-tasks spanning across multiple physical objects that must be performed in different environments. In our proposed solution, we have taken a completely different approach. There is no application model in our system rather situated flows radically dispersed across time and space formulate the application and interaction dynamically. Through the execution of the physical activity and the corresponding flow application states move forward.

Apart from these, existing software infrastructures for pervasive applications are also narrowly scoped towards isolated scenarios such as intelligent class and meeting rooms [11,12,13] or home automation [14]. To efficiently support these scenarios, they rely on statically centralized control mechanisms. Thus, they cannot support dynamic environments. Other infrastructures that apply decentralized control mechanisms are usually targeted at small groups of spontaneously networked computer systems such as personal area networks [17]. By providing control mechanisms of the environment in context through situated flows attached to physical entities, our approach provides a complete distributed solution. In addition by maintaining the states of the interaction, the framework provides a real time coordination of physical objects.

VIII. CONCLUSION

We anticipate that future work environments will be densely instrumented to understand details of work activities and to support users proactively to accomplish their goal at hand. To this end, we have introduced situated flow as a novel technique to model high-level activities in complex work environments. In addition, we described a distributed software framework that provides real time flow discovery, adaptation and execution creating a seamless interaction experience in pervasive workspaces. We have also discussed the utility of our approach through a scenario based case study. We consider that the notion of situated flow and the corresponding software framework will contribute effectively to further research exploration in the pervasive computing domain, particularly one that involves connected physical objects in critical workspaces.

Acknowledgment: This research has been funded by the European Commission (EC 213339 ALLOW).

REFERENCES

[1] Kunze K., Wagner F., Kartal E., Kluge E. M., and Lukowicz P., Does Context Matter? - A Quantitative Evaluation in a RealWorld Maintenance Scenario. Conference on Pervasive Computing, 2009, May 11-14, Nara, Japan.

[2] Stiefmeier, T., Roggen, D., Ogris, G., Lukowicz, P., Tröster, G. Wearable Activity Tracking in Car Manufacturing. IEEE Pervasive Computing. Vol. 7:2, 2008, 42-50.

[3] Kortuem G., Ball, L., Busby, J., Davies, N., Efstratiou, C., Iszatt-White, M., Finney, J., Kinder, K., Sensor Networks or Smart Artifacts? An Exploration of Organizational Issues of An Industrial Health and Safety Monitoring System. Proceedings of Ubicomp 2007, Innsbruck, Austria.

[4] Kjeldskov, J. and Skov, M. B. 2007. Exploring context-awareness for ubiquitous computing in the healthcare domain. Personal Ubiquitous Comput. 11, 7 (Oct. 2007), 549-562.

[5] Bardram, J., Hansen, T., Mogensen, M., and Soegaard, M. Experiences from Real-World Deployment of Context-Aware Technologies in a Hospital Environment. Proceedings of Ubicomp 2006. pp 369-386.

[6] Leymann, Frank; Roller, Dieter: Modeling business processes with BPEL4WS. In: Information Systems and e-Business Management (ISeB). Vol. 4(3), Springer, 2006.

[7] Bucchiarone, A., Lafuente, A., Marconi A., and Pistor, M. A formalisation of Adaptable Pervasive Flows. 6th International Workshop on Web Services and Formal Methods (WS-FM'09), 2009.

[8] Marconi, A., Pistore, A., Sirbu, A., Eberle, H., Leymann, F. and Unger, T. Enabling Adaptation of Pervasive Flows: Built-in Contextual Adaptation. Joint ICSOC & ServiceWave 2009 Conference.

[9] Kortuem, G., Kawsar, F., and Altakrouri, B., "Flow Driven Ambient Guidance."; Eighth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2010), Mannheim, Germany March 29 - April 2, 2010

[10] Kazman, R., Abowd, G., Bass, L., and Clements, P., Scenario- based analysis of software architecture. IEEE Software, 13(6):47–55, November,1996.

[11] Román, M, Hess, C. K., Cerqueira, R., Ranganathan, A., Campbell, R. H., Nahrstedt, K.: Gaia: A Middleware Infrastructure to Enable Active Spaces. IEEE Pervasive Computing, pp. 74-83, Oct-Dec 2002

[12] Garlan, D., Siewiorek, D., Smailagic, A., Steenkiste, P.: Project Aura: Toward Distraction-Free Pervasive Computing. IEEE Pervasive Computing, April- June 2002.

[13] Ponnekanti, S. R., Johanson, B., Kiciman. E.,, Fox, A.: Portability, extensibility and robustness in iROS. IEEE 1st International Conference on Pervasive Computing and Communications (PerCom), 11-19, 2003

[14] Amigo (Ambient Intelligence for the Networked Home Environment) IP, www.amigo-project.org, 2004.

[15] Becker, C,. Schiele, G., Gubbels, H., Rothermel, K.,: BASE - A Micro-broker- based Middleware For Pervasive Computing. 1st IEEE International Conference on Pervasive Computing and Communications (PerCom 03), pp. 443- 451, March 23-26, Fort Worth, USA, 2003.

[16] Hess, C. K., Román, M., Campbell, R. H.: Building Applications for Ubiquitous Computing Environments. International Conference on Pervasive Computing (Pervasive 2002), pp. 16-29, Zurich, Switzerland, August 26-28, 2002

[17] Becker, C., Handte, M., Schiele, G., Rothermel, K.: PCOM - A Component System for Adaptive Pervasive Computing Applications. 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom), Orlando, USA.