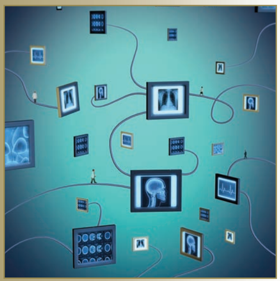

Beyond Health Tracking

A Personal Health and Lifestyle Platform

A personal health and lifestyle platform helps individuals maintain a personal health and lifestyle record and obtain personalized, lifestyle-related advice to improve their health by changing their daily habits. The platform leverages the data stored in a typical personal health record, augmenting it with environmental and sensor data and enabling the monitoring and analysis of an individual's habits. Sharing habits and advice with doctors and friends empowers individuals to become wellness coproducers and leads to a personal health and lifestyle record that is much more useful to the individual maintaining it.

Florian Daniel, Fabio Casati, and Patrícia Silveira
University of Trento

Monica Verga and Marco Nalin
San Raffaele Hospital of Milan

Medical practitioners have long recognized the importance of prevention as a way to maintain the health of an aging modern society and cope with the growing burden on public healthcare systems. For those with noncommunicable diseases (NCDs), such as cancer, depression, diabetes, or cardiovascular diseases, prevention focuses on diet, physical activity, smoking, alcohol consumption, sleep patterns, and stress.[1] More generally, prevention aims at improving people's lifestyle.

In Preve, an EU FP7 project, we are studying how to empower individuals with personal IT solutions and services that motivate them to manage and modify their lifestyles to preserve their health and overall wellbeing. Results so far clearly show that changing and maintaining new habits requires guidance and support. Such help is most effective when it comes from *coproducers* — not only the individual but also healthcare professionals, friends, family, employers, schools, restaurants, or food markets. Help can be explicit, such as when a doctor recommends a specific physical exercise, or implicit, such as when the school cafeteria serves healthy food. As such, coproducers represent the part of the individual's social network that can influence his or her health and wellbeing.

An important part of lifestyle management and modification is having a personal health record (PHR), a health software tool that lets an individual integrate, store, manage, and share personal health information from different sources. WebMD and Google Health are examples of common PHRs. Unfortunately, this solution tends to benefit health professionals, such as doctors or therapists, more than the individual who must maintain the record. Given that the information producers — the health institutions, family doctors, pharmacies, and so on — rarely actively

 1089-7801/11/$26.00 © 2011 IEEE

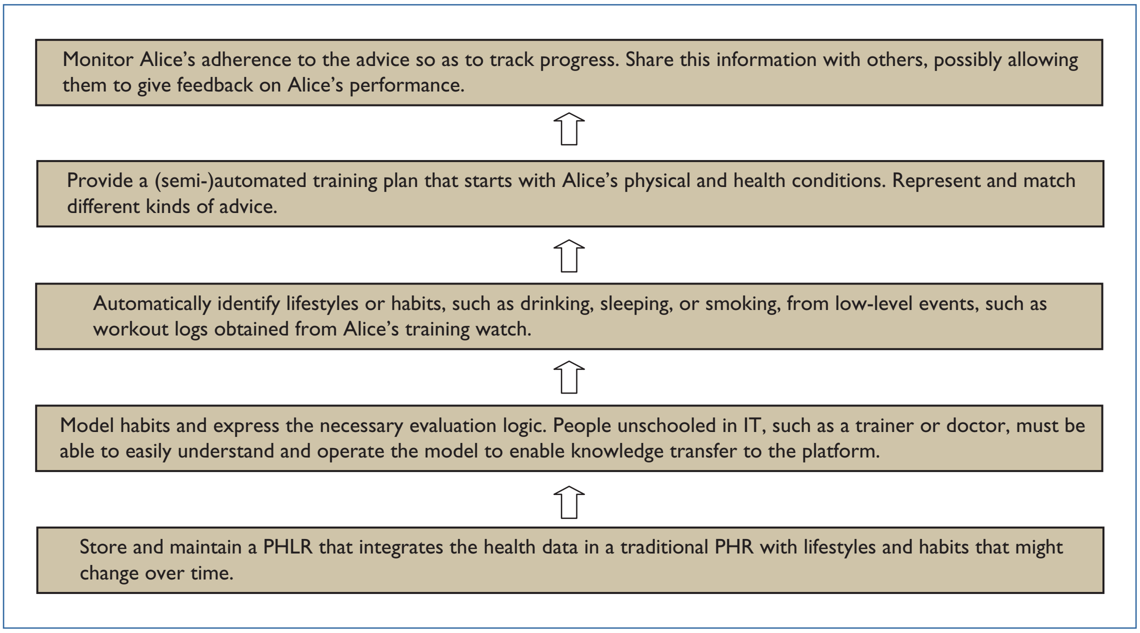

Figure 1. Supporting Alice as she trains for her first half marathon. This degree of support, which becomes increasingly sophisticated, suggests the need to go beyond the traditional personal health record (PHR) in offering advice, monitoring progress, and allowing shared information.

share their information, maintaining a PHR is a labor-intensive burden for the individual, who must resort to manual recordkeeping methods.

To motivate individuals to improve their lifestyle, we've developed a personal health and lifestyle record (PHLR) and a platform that monitors and assesses an individual's lifestyle and provides personalized advice on how to improve that lifestyle (or makes it easier for health professionals to provide such advice). Although, the platform currently supports only physical activity, our ultimate goal is to support NCD prevention.

Lessons in Activity Monitoring

Our work started as a joint effort with the Italian Cycling Federation to develop Pinkr, a GPS-based monitoring application that tracks the cyclist's position in real time, computes his or her power output at that instant as well as overall energy consumption, shares that data with others in real-time, and provides nutrition advice.

From this work, we learned that sports people — even amateurs — are incredibly competitive and like to monitor their performance. We also learned that competitive training, sharing performances, and obtaining others' feedback boost motivation. Finally, we learned that competitors are quite willing to follow advice to improve their performance.

These lessons pushed us to extend Pinkr to support training scenarios for people like Alice. Alice is a young woman who likes to run occasionally, but she doesn't train continuously and so has no particular running goal. Overall she's healthy except for having high blood pressure. Her recent weight gain has prompted her to run more. To improve her motivation, she has bet a friend that she'll be able to run a half marathon within four months, and she's bought a GPS-equipped training watch to track her workouts.

Alice has never trained for an event like this, so she needs a personal training plan that tells her which kind of workout she should do on which day and for how many weeks, accounting for her current health state. Typically, an expert would create such a plan, considering Alice's training objective (run a half marathon), current physical preparation (beginner), and health. Alice will also need a log or diary to record her workouts, because she might not always be able to train on the planned day. Finally, Alice would like to give her friend access to the plan so that the friend can see how close Alice is to winning their bet.

Supporting this scenario means taking over the role of the expert planner and helping Alice track her progress, which in turn means meeting the research challenges in Figure 1. By extending the PHR idea, our personal health and lifestyle (PHL) platform aims to provide this degree of support and lay the foundation for lifestyle-oriented wellness and prevention.

Integrating Health Data

The popularity of PHRs[2] has grown in the past few years. Generic PHRs, such as Google Health, let users view commonsense information about diseases, participate in chats and forums, receive exam results, and perform quick illness tests. Microsoft's HealthVault has a variety of more specific tools: Lose Weight captures weight directly from the scale, processes it, and produces charts; Get Fit captures pulse rate and running distance from the individual's training watch and graphically shows performance; Heart360 reads metrics from a mobile blood pressure monitor and provides historical views and indicators; and LifeScan collects measures of glucose levels to monitor diabetes levels.

In contrast, solutions like Vivago (www.istsec.fi), which monitors body signals, or Polar (www.polar.fi) and Nike Plus (www.nikerunning.nike.com), which monitor workouts, center on a single health concern. Telemedicine[3] focuses on cardiovascular diseases, offering ECG monitoring, for example.

Although they're useful for their particular function, none of these approaches integrates collected data in a wider vision, models or tracks a person's lifestyle, or provides automated, remote health suggestions. Integrating health data is a hard but critical task. In addition to traditional questionnaires — still the most prominent way to collect data — sensors increasingly automate and ease the collection of health data.[4–6] Sensor data is typically streaming, raising new issues about data processing and integration. Generic middleware solutions exist for data stream processing, such as Hermes, Armada, Echo, and IBM's Gryphon, but healthcare has its own processing solutions, such as ReMoteCare,[7] Harmoni,[8] and Health Care Monitoring of Mobile Patients.[9]

Again, however, these processors are restricted to a single function, data integration. Our platform and PHLR aim to move beyond that, translating abstract sensor data into life events, which will simplify data integration and ensure that domain experts manipulate data at the appropriate abstraction level.

Platform Model

As Figure 1 shows, providing lifestyle-driven advice demands features that transcend conventional PHRs by supporting multiple user roles, lifestyle data, habits, events, and real-time advice provision. Figure 2 illustrates the conceptual model (an extended UML class diagram) that underlies our PHL platform and addresses these issues. As the diagram shows, the PHL platform introduces several novel components.

It has three core concepts:

- A PHR contains a profile with information such as weight, height, gender, and age; medical facts, such as exam results and surgeries; and a set of diagnoses, such as allergies or diseases.
- A lifestyle record (LR) contains a set of habits that characterize the individual's lifestyle at a given time, such as being a beginning runner or a light smoker. In this context, "lifestyle" is a set of habits, which appear as text labels in the LR.
- A PHLR is the integration of a PHR and LR.

In addition to these data artifacts, the PHL platform leverages three PHL models that enable monitoring customization: the *habit model*, *advice trigger model*, and *advice model*. All three rely on data in the PHLR as well as the individual's life events being tracked. Through these models, the PHL platform becomes extensible to accommodate myriad lifestyle-related concerns, from sports training to NCD prevention.

Habit Model

The habit model expresses how to match a habit to a person, essentially how to aggregate life events to produce actionable knowledge in the form of a habit. The model accommodates both manual and automatic association. In manual association, the individual fills in an online form that the platform provides. Automatic association is based on a set of life events, such as Alice's workouts, taken from sensor measurements. Sensors can range from simple GPSs that monitor physical activities to complex sensors that evaluate sleep states.

Advice Trigger and Advice Models

The advice trigger model decides when to give advice, and the advice model monitors the advice given using the PHLR as input for advice

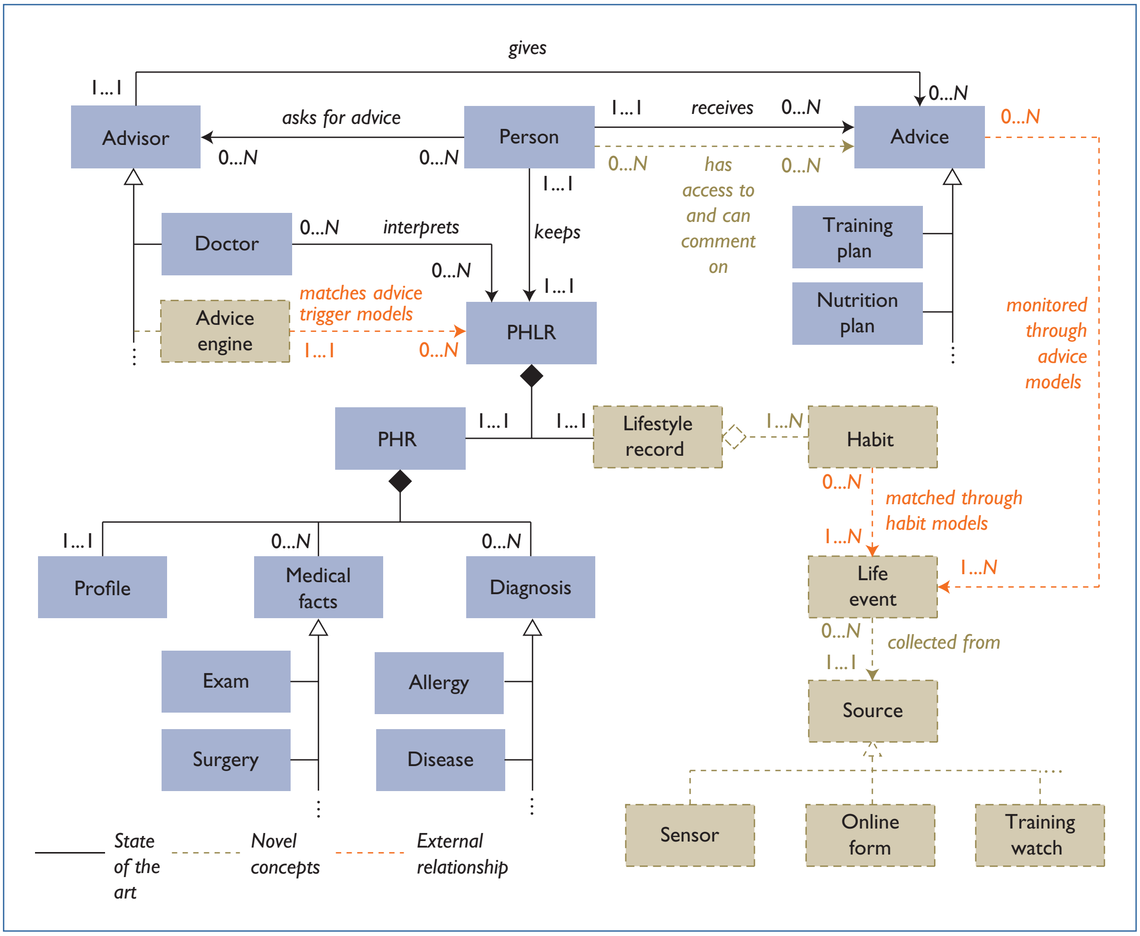

Figure 2. Conceptual model of the personal health and lifestyle (PHL) platform. Black lines and blue boxes denote common practice. The rest of the model consists of novel concepts and relationships, such as an advice engine and the matching of habits and life events.

provision. The goal of advice can be to prevent NCDs that strongly correlate to particular lifestyles, such as cancer and smoking. It could also be to support a physical goal, such as running a half marathon, in a way that accounts for the individual's health conditions, such as Alice being a beginning runner and slightly overweight with high blood pressure. In this case, the advice is the training or nutrition plan.

Advice provision can be manual, in which doctors or trainers enter advice, or automatic, in which the advice engine dispenses advice according to advice trigger models. These models are conceptually similar to sophisticated condition-action rules that monitor when a PHLR verifies a certain condition — either a lifestyle condition in isolation, such as being overweight, or a lifestyle condition plus a goal, such as having high blood pressure and wanting to run a half marathon. The met condition triggers advice.

The advice model automatically monitors how well the individual is following the advice or medical protocol or progressing toward a goal. In advice monitoring, the platform matches raw life events with advice models to obtain progress information, similar to how the habit model aggregates life events to produce habits.

Finally, the platform accommodates the need for others to access advice and possibly lifestyle events, turning the PHL platform into a social instrument that friends, trainers, and doctors can use to comment on an individual's progress or lifestyle choices. This characteristic is what transforms the platform into a vehicle that enables people to join the individual in managing his or her health and become health coproducers.

Moving from Events to Advice

Figure 3 illustrates a habit model and an advice trigger model for the Alice scenario.

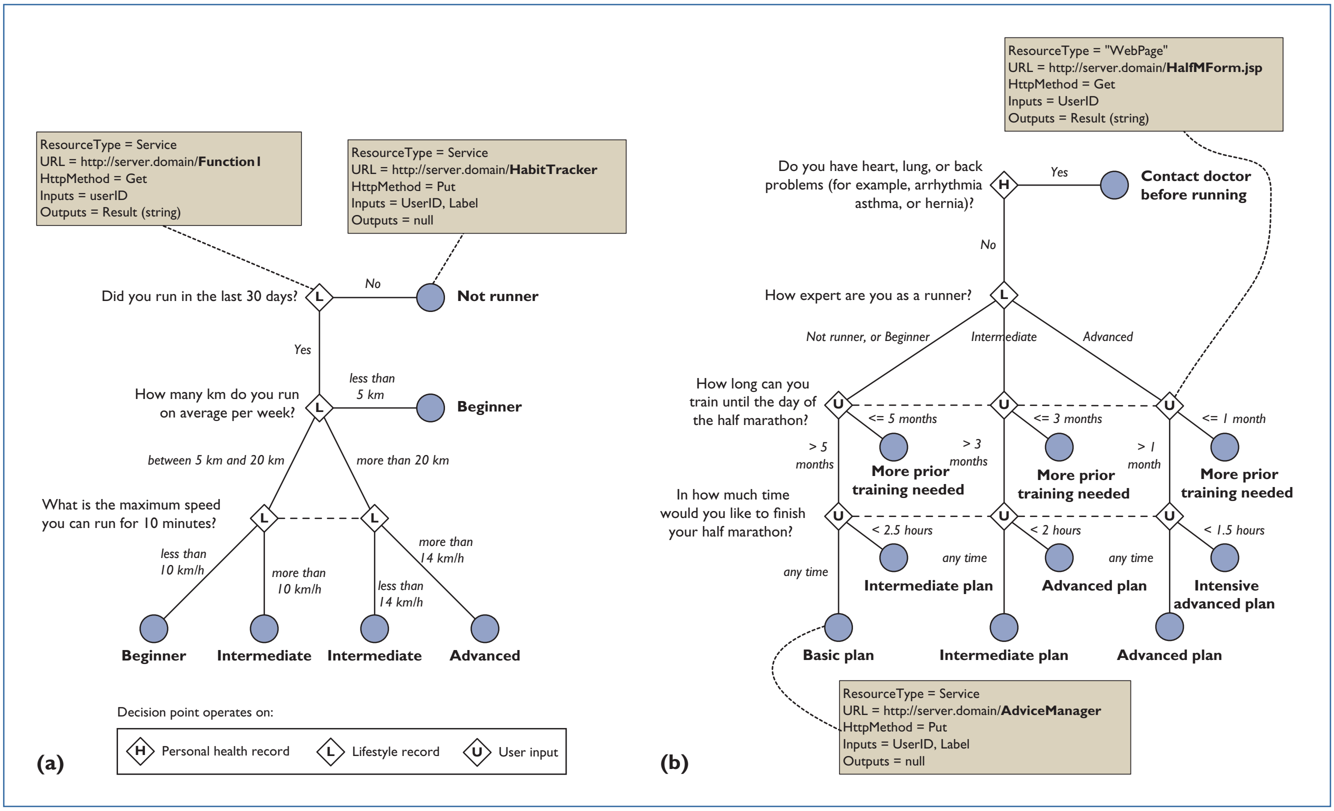

Figure 3. Two personal health and lifestyle (PHL) platform models to support Alice as she trains for a half marathon. (a) A habit model and (b) an advice trigger model. Dashed lines relate nodes with the same decision to be taken or same service to be invoked. The text boxes exemplify how to annotate nodes in the tree.

The habit model expresses how to rank Alice's running performance; the advice trigger model decides which training plan is suitable. We don't show the advice model (the training plan), because its modeling is similar to the two other PHL models. The advice model monitors Alice's progress toward running a half marathon and how well Alice is following advice.

The PHL models' graphical formalism is oriented to domain experts, such as doctors, trainers, or dieticians, who have the domain knowledge necessary to characterize habits and advice from a set of health conditions, habits, or life events. However, these domain experts lack the knowledge to implement the software that can automatically evaluate such descriptions. Thus, IT experts must work with domain experts to implement the necessary evaluation functions as Web services or webpages and bind the PHL model to these functions through annotations.

Formally, PHL models are similar to decision trees with a structure that we describe using the tuple `Model` = <`Name`, `ModelType`, {$\texttt{Node}_i$}, {$\texttt{Arc}_j$}, {$\texttt{Event}_k$}>, where

- `Name` is the unique name associated with the model;
- `ModelType` specifies if the model is a habit, advice trigger, or advice model;
- {$\texttt{Node}_i$} is the set of tree nodes;
- {$\texttt{Arc}_j$} is the set of arcs connecting the tree nodes; and
- {$\texttt{Event}_k$} is the set of lifestyle events that may trigger model evaluation.

We define a node as `Node` = <`Label`, `NodeType`, `ResourceType`, `URL`, `HttpMethod`, {$\texttt{Input}_m$}, {$\texttt{Output}_n$}>, where

- `Label` contains the text label associated with the node (such as, "Did you run in the last 30 days?");
- `NodeType` is either `InternalNode` or `LeafNode`;
- `ResourceType` is either `Service` (to invoke an evaluation function) or `WebPage` (to ask

the user for explicit input via the PHL portal);

- `URL` points to the respective Web service implementing the decision logic or the webpage that the user accesses to input the necessary data;
- `HttpMethod` specifies whether a `Get`, `Post`, or `Put` operation must invoke the Web service or webpage (we use models only to create or update facts in the PHLR, so we don't support delete operations);
- {$\texttt{Input}_m$} represents the set of input parameters of the Web service or webpage; and
- {$\texttt{Output}_n$} represents the set of output parameters.

Intermediate nodes always have at least one input parameter, which is the user's unique `UserID`, and one output parameter, `Result` (the evaluation's result).

We explicitly label each node to clarify what data each internal node operates on and to show how the evaluation of a PHL model can exploit the integration of PHR and lifestyle data and user input. The labels in Figure 3 (legend in the lower left corner) are for presentation only, given that in practice each service knows what data to look at, and webpages always represent user input.

The model's arcs are of the form `Arc = <Parent, Child, Condition>`, where `Parent` is the parent node in the tree, `Child` is the child node, and `Condition` (tree arc label) allows the definition of a condition over the evaluation output of the arc's parent. Conditions can consist of basic comparison operators (`<`, `<=`, `=`, `>=`, `>`) for numbers and operators (`=` or `<>`) for strings. For example, the interpretation of the "yes" label after the first node in Figure 3a is `"Parent.Result = 'yes'?"`

The PHL engine begins model evaluation after receiving a user request or in response to a triggering event. For each internal node, starting with the root node, the engine invokes the respective Web service or webpage and evaluates the node's arcs against the result. Evaluation follows only arcs whose conditions are true. A correct tree definition, therefore, requires the specification of mutually exclusive conditions for each arc. If evaluation reaches a leaf node, the engine invokes that node's Web service and terminates tree processing.

The process of modeling habits and advice is innovative and subjective and involves domain knowledge that most IT experts don't have, as well as programming knowledge that most domain experts lack. A habit definition strongly depends on the doctor's interpretation and specialization, but an IT expert must be involved to translate that knowledge into a machine-readable model. Tailoring might also be necessary to ensure that the PHL model accurately accounts for an individual's lifestyle needs and medical conditions.

To our knowledge, there is no well-defined literature on modeling habits and advice. Once we implement our PHL platform, we'll be able to validate our models' suitability and fine-tune the level of granularity to adequately model habits, advice triggers, and advice. Experience will also tell us if we must account for model dependencies, such as those between one piece of advice and the advice trigger model of another piece, or if we must add more complex evaluation logic, such as nonexclusive conditions.

Architecture and Implementation

The PHLR and PHL models drive our platform's functionality, letting individuals maintain a PHR, store lifestyle events, have an automatically updated LR, obtain and monitor health improvement advice, and share their advice and monitoring pages with their health coproducers.

Figure 4a depicts our platform's functional architecture. The PHL engine automatically evaluates PHL models, each of which must be deployed in the engine and bound to the model's set of triggering life events. The engine constantly monitors the life events flowing through the lifestyle event bus, which acts as event mediator between the sensors and the PHL engine, and evaluates a model when it receives a user request or a triggering event.

Life events can come from a variety of sources, such as online forms, training watches, mobile applications, electronic scales, or glucometers — each of which requires a suitable event adapter. Our current platform supports online forms, training watches, and a dedicated mobile application. For example, a workout event consists of a date, time, activity type, and the running track's GPS coordinates. The deployment or termination of a new model in the engine causes the generation of a life event that can trigger other models. For example, the deployment of Alice's training plan can trigger

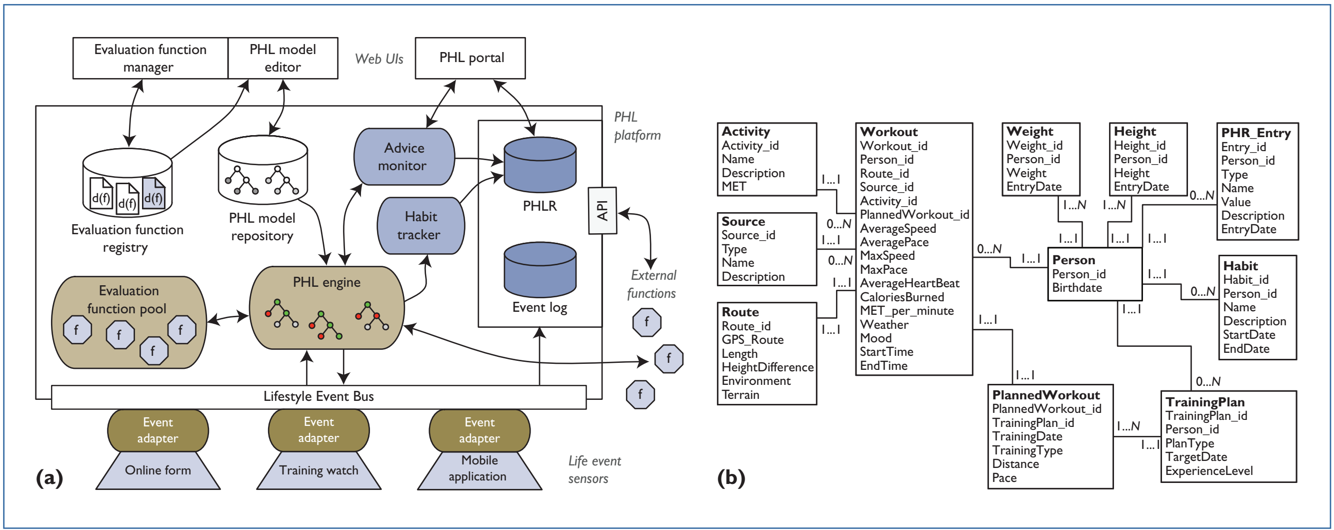

Figure 4. Architecture of the personal health and lifestyle (PHL) platform and integration with evaluation functions. (a) Functional architecture and (b) simplified, anonymized view over the platform's internal database as exposed to evaluation functions for physical exercise assessment and monitoring.

a nutrition plan that considers her high blood pressure.

Once the engine instantiates a model, it evaluates that model's internal nodes until it reaches a leaf node. Each node causes the engine to interact with either a Web service or a webpage, depending on the node's annotation. Leaf nodes are associated only with services, which in this case are the habit tracker and advice monitor. The habit tracker adds and updates identified habits, for example, if Alice advances from beginner to intermediate status. The advice monitor gives and monitors advice.

The IT expert can associate internal nodes with a webpage to collect user input or with an evaluation function that is either built in or an external RESTful Web service. The platform's evaluation function pool stores built-in services, such as those to assist training; external functions can run remotely. Services can securely access both the PHLR and the event log, providing a view of past events through a dedicated API, which gives the functions a restricted and anonymized view of the stored data.

Figure 4b shows the read-only view that we use in our prototype implementation. The PHR comprises the entities Person, Weight, Height, and PHR_Entry, containing diagnoses, for example. Person, the core entity, provides only anonymous identifiers plus basic data that an IT expert might need to implement the functions. It reveals no private information. The PHR's internal implementation is based on our former work on digital sociosanitary records in the province of Trento.[10]

The LR contains only the Habit entity. The event log comprises Workout, Activity, Source, and Route entities, which could be a run, running or cycling, human input or a sensor, and the workout path that the GPS tracks, respectively. Finally, advice representations include the TrainingPlan and PlannedWorkout entities. In Figure 3a, to answer, "Did you run in the last 30 days?" the evaluation function would, for example, issue a query to view workouts that have been registered in the last 30 days.

Model Management

IT experts manage the implemented evaluation functions and PHL models through the evaluation function manager, a Web application that stores descriptors of both built-in and external evaluation functions. IT and domain experts can then use the descriptors in the PHL model editor to design models. An internal PHL model repository stores the completed models and supports their deployment in the PHL engine.

Users interact with the platform through the PHL portal, a Web application that gives them access to all the platform's features. Figure 5 shows the screens that Alice might use to manage her PHR, upload her workouts, inspect her progress, and share her training plan with friends.

Unlike habits, which the PHL engine tracks in the LR, advice and advice triggers can require suitable webpages to gather user input, communicate advice, and support monitoring and sharing. At present, our user interfaces are hardcoded, but our goal is to enable the users to visualize advice progress and conformance data through interfaces that they can compose using simple graphical widgets.[11] In this way, we can fully decouple advice monitoring from its visualization.

The PHL models allow a great deal of flexibility in platform implementation, but there are other opportunities for extension. Implementing custom evaluation functions opens the option of plugging in custom decision logic. Abstracting raw sensor data into life events enables the addition of new sensors and data sources, given a respective event adapter. Finally, designers can extend the platform's internal database structure, for example to accommodate new PHR data or new life event types. Although this last extension is relatively intrusive, the use of data views minimizes its impact on the evaluation functions operating on the shared PHLR.

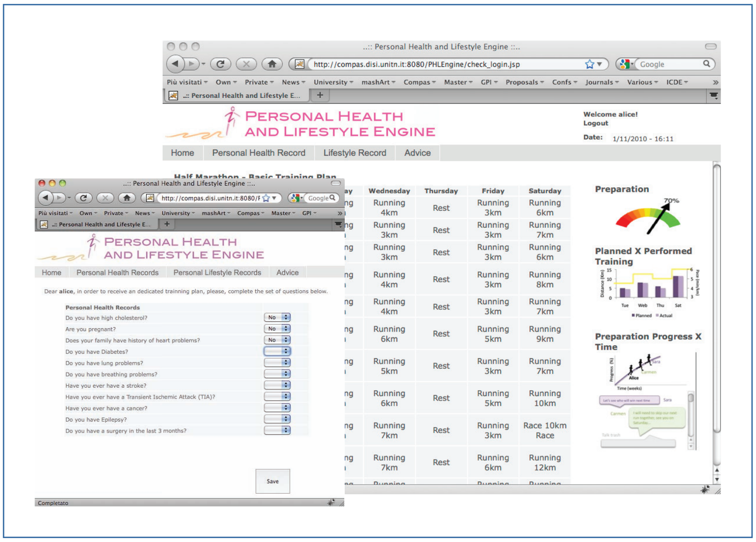

Figure 5. Personal health and lifestyle (PHL) portal. Alice might use these screens to manage her personal health record (PHR), upload her workouts, inspect her progress, and share her training plan with friends.

Our work shows great potential for PHLRs and lifestyle engines to improve individual health. The Pinkr deployment to aid training in amateur cycling revealed that both social interaction and continuous monitoring are promising positive motivators in lifestyle management and improvement. Our experience with Pinkr and our project work also point to habit and advice models as viable techniques for defining how to derive lifestyles and analyze progress in following advice.

Our work has also taught us that collecting lifestyle events and integration health data is far from being easy for many reasons — including IT, medical, and legal concerns. Scalability might become an issue if the platform's user base becomes large enough, say thousands of users. However, because our training scenarios to date have not needed scalability, we could not test our platform for this.

We plan to continue this work by validating our PHL platform and models in the NCD domain. We also plan to make a continuously updated PHL platform prototype accessible at http://compas.disi.unitn.it:8080/PHLEngine/lifestyle.jsp.

The most exciting aspect of our future work will be model validations, the discovery of new models, and the unveiling of correlations between lifestyles and NCDs. Such revelations will drive our work as we add users and sensors and increase the scale of data collection and mining. Our hope is that such activities will enhance medical knowledge and help individuals in all walks of life improve their health and wellbeing.

Acknowledgment

This work has been partially supported by the EU Projects PREVE (FP7-ICT-248197) and CILMI (FP7-IRSES-247620).

References

1. *World Health Statistics 2010*, World Health Organization, 2010; www.who.int/entity/whosis/whostat/EN_WHS10_Full.pdf.
2. M. Ball, C. Smith, and R. Bakalar, "Personal Health Records: Empowering Consumers," *J. Healthcare Information Management*, vol. 2, no. 1, 2007, pp. 76–86.
3. T.L. Huston and J.L. Huston, "Is Telemedicine a Practical Reality?" *Comm. ACM*, vol. 43, no. 6, 2000, pp. 91–95.
4. A. Yamaguchi et al., "Monitoring Behavior in the Home Using Positioning Sensors," *Proc. IEEE Eng. in Medicine and Biology Soc.* (EMBS 98), IEEE Press, 1998, pp. 1977–1979.
5. M. van Gils et al., "Feasibility and User Acceptance of a Personal Weight Management Based on Ubiquitous Computing," *Proc. IEEE Engineering in Medicine and Biology Society* (EMBS 01), IEEE Press, 2001, pp. 3650–3653.

6. J. Pärkkä et al., "A Wireless Wellness Monitor for Personal Weight Management," *Proc. IEEE Eng. in Medicine and Biology Soc.* (EMBS 00), IEEE Press, 2000, pp. 83–88.
7. M. Fischer et al., "ReMoteCare: Health Monitoring with Streaming Video," *Proc. Int'l Conf. Mobile Business* (ICMB 08), IEEE CS Press, pp. 280–286.
8. I. Mohomed et al., "HARMONI: Motivation for a Health-Oriented Adaptive Remote Monitoring Middleware," *Proc. Int'l Workshop Ubiquitous Computing for Pervasive Healthcare Applications* (UbiHealth 06); www.cs.toronto.edu/~iq/harmoni_ubihealth2006.pdf.
9. G. Amato et al., "Health Care Monitoring of Mobile Patients," *ERCIM News*, vol. 60, 2005, pp. 69–70.
10. J. Stevovic et al., "Event-Driven Privacy-Aware Infrastructure for Social and Health Systems Interoperability: CSS Platform," *Proc. Int'l Conf. Service-Oriented Computing* (ICSOC 10), Springer, pp. 708–710.
11. F. Daniel et al., "Hosted Universal Composition: Models, Languages, and Infrastructure in MashArt," *Proc. 28th Int'l Conf. Conceptual Modeling* (ER 09), Springer, 2009, pp. 428–443.

Florian Daniel is a postdoctoral researcher at the University of Trento. His research interests include mashups and user interface composition approaches for the Web, Web engineering, and quality and privacy in business intelligence applications. Daniel has a PhD in information technology from Politecnico di Milano. Contact him at daniel@disi.unitn.it.

Fabio Casati is a professor of computer science at the University of Trento. His research interests include the prevention of noncommunicable diseases (NCDs), remote and real-time healthcare, collaborative programming, and models for scientific disseminations that can help scientists work more efficiently. Casati has a PhD in information technology from Politecnico di Milano. Contact him at casati@disi.unitn.it.

Patrícia Silveira is a doctoral student in the Doctoral School of Information and Communication Technology at the University of Trento. Her research interests include patterns and models to represent and monitor lifestyles and their correlation with NCDs and business-process compliance in service-based environments. Silveira has an MSc from the Pontifical Catholic University of Rio Grande do Sul, Brazil. Contact her at silveira@disi.unitn.it.

Monica Verga is a research scientist at San Raffaele Hospital of Milan. Her research interests include motivational strategies for personal wellbeing and disease prevention, personal health systems, edutainment, surgical robotics, and privacy and security. Verga has an MSc in biomedical engineering from Politecnico di Milano. Contact her at verga.monica@hsr.it.

Marco Nalin is a research scientist at San Raffaele Hospital of Milan. His research interests include personal health systems, mobile health management applications for personal wellbeing and disease prevention, cognitive robotics and edutainment, surgical robotics, energy management systems, cloud computing, and privacy and security. Nalin has an MSc in electronic engineering from the University of Padova. Contact him at nalin.marco@hsr.it.

cn *Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.*