

# Orquestração de Serviços de um Plano de Cuidados Ubíquo

Eliseu Germano<sup>1</sup>, Bruno Silvestre<sup>1</sup>, Sergio T. Carvalho<sup>1</sup>

<sup>1</sup>Instituto de Informática – Universidade Federal de Goiás (UFG)

{eliseugermanosilva,brunoos,sergio}@inf.ufg.br

**Abstract.** *A care plan is a set of information that links patients to their health care. This paper presents a scenario for using a platform developed to enable the composition and coordination of care plan tasks that can be implemented as services. This platform has a domain-specific language that allows the modeling of the tasks of a care plan as a sequence of actions that is controlled from workflows.*

**Resumo.** *Um plano de cuidados consiste em um conjunto de informações que relaciona os pacientes a seus respectivos tratamentos de saúde. Este artigo apresenta um cenário para uso de uma plataforma desenvolvida para viabilizar a composição e a coordenação de tarefas de planos de cuidados que podem ser implementadas como serviços. Essa plataforma conta com uma linguagem específica de domínio que permite a modelagem das tarefas de um plano de cuidados como uma sequência de ações que é controlada a partir de workflows.*

## 1. Introdução

Um dos grandes desafios investigados em pesquisas nas áreas de ciências da saúde está relacionado à necessidade de realizar o acompanhamento contínuo de pessoas com doenças crônicas [Sabaté 2003]. Isso se deve a uma série de fatores que são intrínsecos ao problema de acompanhar continuamente e por tempo prolongado diversos aspectos da saúde humana. Esse cenário tem impulsionado o desenvolvimento de pesquisas envolvendo o uso da computação ubíqua em ambientes de assistência domiciliar à saúde, onde a partir da adição de elementos computacionais não invasivos, tem sido investigado formas de automatizar processos relacionados ao monitoramento de um paciente [ElHelw et al. 2009][Deng et al. 2011].

A assistência domiciliar à saúde pode ser entendida como um conjunto de serviços médicos e terapêuticos prestados na residência de um paciente para promover, manter ou restaurar a sua saúde [Caffrey et al. 2011]. Uma das principais motivações em torno do uso de elementos computacionais em ambientes de assistência domiciliar à saúde está relacionada à possibilidade de melhorar o engajamento entre as pessoas envolvidas em um tratamento e as tarefas desse tratamento. Nessa perspectiva, há propostas de sistemas de monitoramento remoto de pacientes que têm como principal objetivo automatizar algumas dessas tarefas, visando integrar aos ambientes de monitoramento elementos sensoriais (e.g., sensores fisiológicos) e componentes de *software* (e.g., notificadoros de eventos) [Carvalho et al. 2011][Palesandro et al. 2016].

Uma ferramenta conceitual da enfermagem chamada plano de cuidados é tipicamente utilizada para viabilizar aos profissionais de saúde realizar o gerenciamento de

informações relacionadas aos cuidados prestados aos pacientes. Sistematizar algumas tarefas de um plano de cuidados para auxiliar os envolvidos em um tratamento nas tarefas do plano (e.g., adicionando elementos de notificação) é uma das formas investigadas para inserir elementos computacionais em ambientes de monitoramento remoto de pacientes [Germano et al. 2016]. No entanto, existe uma complexidade em construir tais sistemas computacionais para representação de planos de cuidados devido a alta variabilidade de tipos de tratamentos de saúde e pela dinamicidade em torno dos requisitos de saúde de um paciente. Dessa forma, há algumas propostas de soluções computacionais adaptativas para representação de planos de cuidados [Abidi and Chen 2006][Browne et al. 2004].

Neste artigo é apresentada uma plataforma desenvolvida para viabilizar a construção de planos de cuidados como um conjunto de tarefas implementadas na forma de serviços computacionais e que são coordenadas a partir de um modelo baseado em *workflows*. Essa abordagem é apresentada a partir de uma Linguagem Específica de Domínio (do inglês, *Domain-Specific Language - DSL*) [Fowler 2010] usada para descrever as sequências em que as tarefas são executadas, e de um mecanismo de orquestração dos serviços baseados em um modelo Redes de Petri. Diante do uso da DSL e dos serviços que são registrados dinamicamente na plataforma é possível a construção de planos de cuidados adaptáveis e personalizáveis a partir da escolha e do encadeamento dos serviços de acordo com as tarefas específicas de cada tratamento.

O conteúdo deste artigo está organizado da seguinte forma: na Seção 2 é apresentada uma contextualização em torno do trabalho; na Seção 3 é abordada a proposta de *workflows* como serviços a partir de uma motivação relacionada ao domínio de saúde, uma arquitetura de uma plataforma e a forma em que os serviços de saúde são orquestrados nessa plataforma; na Seção 4 é apresentado um cenário de testes em que dados fisiológicos para diagnosticar se um paciente está com hipertensão são monitorados continuamente em um ambiente computacionalmente ubíquo; e a Seção 5 traz as considerações finais.

## 2. Contextualização

Definida em 1991 por Mark Weiser como a onipresença da computação no cotidiano das pessoas a partir da miniaturização dos dispositivos computacionais [Weiser 1991], a computação ubíqua vem se concretizando a partir do advento de novos paradigmas na computação, como a Internet das Coisas [Atzori et al. 2010]. O termo técnico “coisa” (do inglês, *things*) é utilizado para representar as entidades computacionais que podem ser classificadas de acordo com um contexto, que por sua vez, pode ser físico ou lógico [Hofer et al. 2003]. Enquanto as entidades físicas são tipicamente medidas por sensores físicos (e.g., detecção dos batimentos cardíacos usando um sensor fisiológico), as entidades lógicas são caracterizadas por ser uma composição de propriedades estimadas a partir de uma combinação de dados de diferentes tipos de sensores físicos (e.g., detecção de um estado de saúde a partir de dados obtidos por um conjunto de sensores fisiológicos).

Considerando que um dos grandes desafios da computação ubíqua consiste em tratar dados de diferentes fontes e combiná-los para obter algum resultado [Caceres and Friday 2012][Conti et al. 2012], neste artigo é apresentada uma plataforma que utiliza a abstração de serviços para oferecer mecanismos para manipulação de dados de saúde obtidos em ambientes computacionalmente ubíquos. A proposta considera que os serviços trabalham como *gateways* entre os sensores e a plataforma, criando um

nível de indireção para o acesso aos dados dos sensores a partir de APIs dos serviços. Diante disso, a plataforma oferece um conjunto de recursos para viabilizar o processo de orquestração desses serviços.

Uma orquestração pode ser vista como a composição de um conjunto de processos em que um processo central coordena os demais processos [Benyoucef et al. 2011]. Dessa forma, em uma orquestração de serviços, o serviço orquestrador controla o fluxo de execução de um conjunto de serviços na *web*. Assim, no contexto da plataforma, o desafio de combinar dados sensoriais de diferentes fontes consiste em construir um processo orquestrador que recebe, filtra e interliga os dados de um conjunto de serviços.

A forma em que as orquestrações são realizadas na plataforma ocorre por meio de *workflows*, ou seja, em uma sequência de etapas que representa a sistematização de uma tarefa. Fazendo um mapeamento para o processo de orquestração de serviços, um *workflow* representa o processo orquestrador, onde cada uma de suas etapas representa uma chamada a um serviço. Nesse sentido, executar um *workflow* corresponde ao processo de orquestrar um conjunto de serviços no qual esse *workflow* é composto.

A ideia de utilizar *workflows* em sistemas de computação orientados a serviços é algo investigado sob diversas abordagens [Tan et al. 2015]. No entanto, na plataforma apresentada neste artigo é utilizado um mecanismo diferente para criação de abstrações em relação ao uso de *workflows*, onde a própria estrutura do *workflow* é tratada como um serviço. Dessa forma, na plataforma os *workflows* possuem *interfaces* constituídas de *endpoints* que podem receber parâmetros de entrada, retornar dados e que são acessíveis a partir de protocolos de comunicação.

### **3. Workflows como Serviços**

Nesta seção são apresentadas as principais motivações que levaram à decisão de trabalhar com *workflows* como mecanismos para representar orquestrações de serviços de saúde. Além disso, são apresentados os detalhes de como essa abordagem foi utilizada do ponto de vista da arquitetura e da implementação de alguns componentes da plataforma.

#### **3.1. Motivação**

Do ponto de vista da área da saúde, principalmente no que se refere ao processo de acompanhamento de cuidados individuais de pacientes com doenças crônicas, há uma motivação em torno do uso de entidades computacionais não intrusivas (e.g., sensores) como forma de reduzir o problema de adesão aos tratamentos [Deen 2015][Vieira and Carvalho 2016]. A dificuldade de controlar a adesão desses pacientes que realizam demandas por tratamentos contínuos e por tempo prolongado é um dos principais desafios da assistência domiciliar à saúde [Sabaté 2003].

De acordo com a Organização Mundial da Saúde (OMS), a adesão de um paciente é definida como o grau de comportamento dele em relação às recomendações elaboradas por um prestador de serviços de saúde [Sabaté 2003]. Dessa forma, algumas questões relacionadas à autonomia do paciente podem influenciar no tratamento, como, esquecimentos em relação ao cumprimento de tarefas prescritas, falta de conhecimentos para lidar com os sintomas da doença, má compreensão das instruções prescritas, entre outras. Essas questões podem ser tratadas por meio da sistematização de um plano de cuidados.

Um plano de cuidados é uma ferramenta utilizada na enfermagem, cujo objetivo é permitir aos cuidadores o gerenciamento sistematizado das informações relacionadas aos cuidados prestados aos seus pacientes [Horta 1979]. Essa ideia tem sido estendida no sentido de que algumas funções dessa ferramenta podem ser automatizadas, tanto para auxiliar os profissionais de saúde no gerenciamento das atividades do plano quanto para envolver o paciente nos processos relacionados ao seu tratamento. A adição de elementos computacionais autônômicos não invasivos tem sido uma das formas investigadas para automatizar processos dessa ferramenta de saúde [Germano et al. 2016].

Visando lidar com características de dinamicidade de um plano de cuidados, a abordagem deste trabalho apoia-se na proposta de Eric Browne [Browne et al. 2004] que considera que um fluxo de tratamento de um paciente pode ser representado por um *workflow*, onde cada uma das etapas desse *workflow* possui um *subworkflow* que pode ser visto como um *workflow* com um nível de especificidade maior, como pode ser visualizado na Figura 1. Assim, considerando que em situações distintas diferentes *subworkflows* possam ser utilizados em uma mesma etapa de um *workflow*, a característica de dinamicidade do plano de cuidados pode ser obtida.

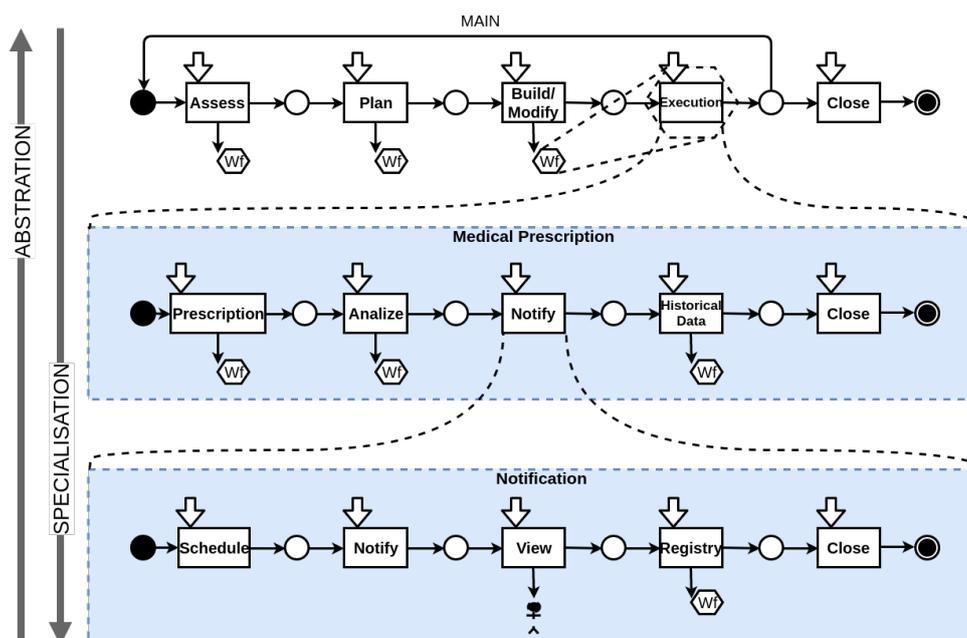


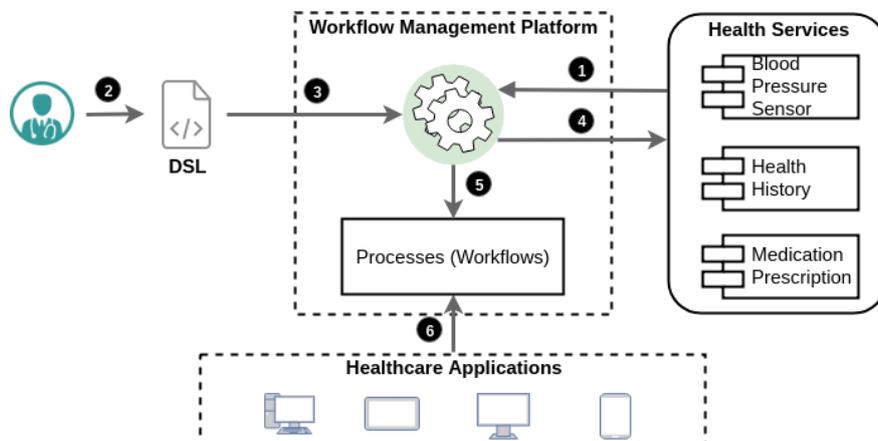
Figura 1. Modelo de *subworkflows* (adaptada de [Browne et al. 2004]).

### 3.2. Arquitetura da *Workflow Management Platform*

Tendo em vista o modelo proposto por Eric Browne (Figura 1) e considerando o fato que algumas tarefas de um plano de cuidados podem ser implementadas como serviços computacionais, desenvolvemos a *Workflow Management Platform* (WfMP) para oferecer um conjunto de mecanismos de abstração para construção e gerência de *workflows* de um plano de cuidados. Entre esses mecanismos estão o registro de serviços, a definição de *workflows* a partir de um conjunto desses serviços, e a forma na qual os *workflows* são acessados [Germano et al. 2018].

A Figura 2 apresenta uma sequência com os passos de interação entre diferentes entidades que se comunicam com a plataforma. Nesses passos estão os procedimentos

necessários para o registro de serviços (e.g., realizado por um desenvolvedor de *software*), especificação de *workflows* (e.g., realizado por um profissional de saúde utilizando uma DSL) e a utilização desses *workflows* na plataforma a partir de aplicações de saúde.



**Figura 2. Arquitetura da *Workflow Management Platform*.**

No passo 1 há um arco direcionado dos serviços de saúde para a WfMP, que representa o momento em que os serviços são registrados na plataforma. Esse procedimento deve ser realizado a partir da API (*Application Programming Interface*) da plataforma, sendo configuradas na WfMP as tarefas de baixo nível de abstração relacionadas à comunicação entre os serviços. Após o registro dos serviços, no passo 2, um usuário do domínio de saúde, utilizando um conjunto de regras e restrições de uma DSL, escreve um *script* para representar um *workflow*. Uma vez que o *workflow* é constituído de trocas de mensagens entre serviços, estes já devem ser conhecidos pelo usuário do domínio, podendo ser consultados a partir da plataforma em um passo anterior à escrita do *script*.

No passo 3, ainda seguindo o fluxo apresentado na Figura 2, um *script* é submetido à plataforma, que por meio de um analisador deve avaliar sintaticamente cada uma das regras descritas. Uma vez que não haja erros sintáticos, um modelo semântico deve ser construído, sendo que este representa a estrutura do *workflow* em um modelo de objetos. Durante a construção do modelo semântico é verificada a disponibilidade dos serviços (passo 4). Com o modelo preenchido, no passo 5 o *workflow* deve ser publicado na plataforma, podendo ser acessado pelas aplicações como um novo serviço (passo 6).

Neste artigo são apresentados aspectos da WfMP em relação a execução de *workflows* sob a perspectiva dos usuários da linguagem. Uma descrição dos componentes computacionais que realizam o processamento desses *workflows*, desde o processo de análise sintática de um *script* da DSL até o momento em que os *workflows* são disponibilizados a partir de uma API, é feita em [Germano et al. 2018].

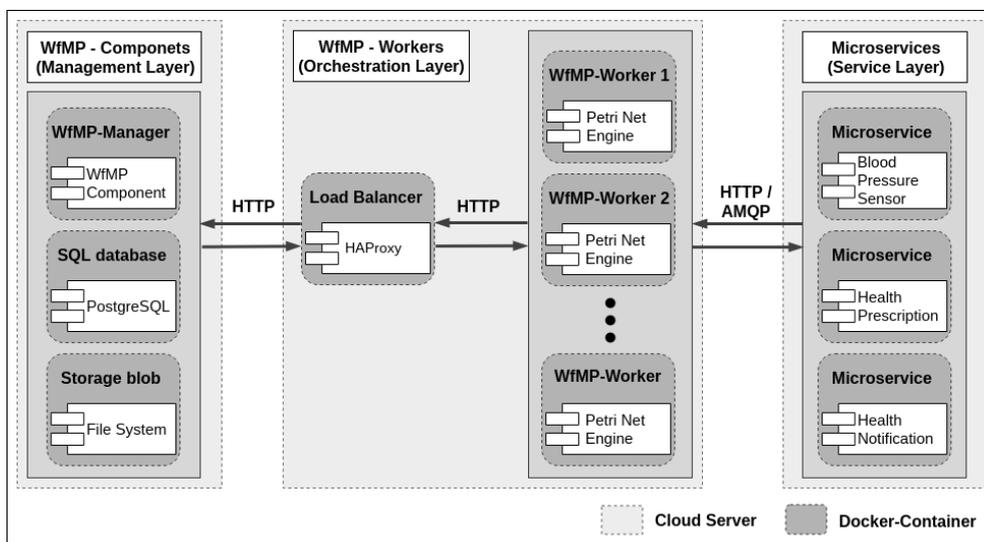
### **3.3. Orquestração dos Serviços na *Workflow Management Platform***

Na WfMP os *workflows* são modelados a partir de uma DSL, concebida seguindo as diretrizes de Martin Fowler [Fowler 2010]. De acordo com esta abordagem, uma DSL é composta por uma sintaxe e um modelo computacional que representa a semântica de um *script* (programa) construído a partir dessa sintaxe, chamado modelo semântico. O modelo semântico utilizado na WfMP é conhecido como Rede de Petri, e foi escolhido devido principalmente à sua forte aceitação para a representação de *workflows*.

De modo geral, uma Rede de Petri é um tipo de formalismo utilizado para modelagem e análise de processos. Elas são compostas por *places*, *transitions* e *tokens*. Os *places* são os elementos passivos que representam uma localização geográfica, uma fase ou um estado da rede. *Transitions* são elementos ativos, cujo objetivo é modelar os processos que são executados durante a passagem de um *token* de um *place* para outro. Os *tokens*, por sua vez, representam objetos que trafegam pelos *places* da rede, sendo os responsáveis por determinar o estado dela em tempo de execução [Murata 1989] [Salimifard and Wright 2001].

O processamento de um *script* da DSL na WfMP é feito em duas etapas. Na primeira delas o *script* passado é analisado sintaticamente e, caso não haja erros, uma árvore sintática correspondente é gerada. Em seguida, essa árvore é percorrida e uma representação do *workflow* em um modelo de objetos é criada. Essa representação é persistida em um banco de dados e, quando o *workflow* é invocado a partir da API da plataforma, o modelo de objetos é recuperado e executado de acordo com a semântica de execução de uma Rede de Petri.

A Figura 3 apresenta os principais componentes de *software* envolvidos no processo de execução dos *workflows*. Do lado esquerdo, há um servidor contendo componentes que persistem os *workflows* (*SQL database* e *Storage blob*) e gerenciam o processo de orquestração (*WfMP-Manager*). Na parte central, há um conjunto de componentes responsáveis por executar os *workflows* (*WfMP-Workers*) e um componente responsável por realizar a distribuição de cargas entre eles (*Load Balancer*). E por fim, do lado direito, há um conjunto de serviços de saúde registrados na plataforma (*Microservices*).



**Figura 3. Orquestração dos Serviços na WfMP.**

Uma vez que a WfMP é utilizada para coordenar e compor serviços de saúde, uma das principais funções da plataforma consiste em realizar a orquestração desses serviços. Como os *workflows* da plataforma são computacionalmente modelados como Redes de Petri, os *Workers* apresentados na Figura 3 são motores de execução de Redes de Petri que realizam orquestrações dos serviços conforme as redes são processadas. O processamento dos *workflows* nos *Workers* utiliza uma semântica em que os *places* representam elementos passivos e as *transitions* os elementos ativos de uma orquestração. Dessa

forma, os *places* representam os estados ociosos de um *workflow*, enquanto as *transitions* representam as chamadas a *endpoints* de serviços registrados na plataforma.

Durante o processo de orquestração de serviços na plataforma, a comunicação entre *Workers* e *Microservices* ocorrem de forma síncrona e assíncrona, sendo utilizado respectivamente para cada uma dessas formas de comunicação os protocolos HTTP (*Hypertext Transfer Protocol*) e AMQP (*Advanced Message Queuing Protocol*). Para que a orquestração ocorra, há uma série de componentes que são utilizados para viabilizar a comunicação, tais como, *middlewares* de comunicação, componentes para controle dos serviços registrados, repositórios de tipos de dados de saúde, entre outros. Uma visão geral dos principais componentes é apresentada em [Germano et al. 2018].

#### 4. Cenário de testes

Como forma de analisar a representação de possíveis *workflows* usados na área da saúde na WfMP, foi considerado um possível plano de cuidados para pacientes de hipertensão (*HypertensionCarePlan*), apresentado na Figura 4 a partir do *workflow HypertensionCarePlan* (linha A). A mesma Figura 4 apresenta os seus *subworkflows* (linhas B, C e D): *HypertensionDiagnostics*, *BloodPressureAnalyzer* e *HypertensionTreatment*. Do ponto de vista do domínio de saúde, o objetivo desse cenário consiste em representar um plano de cuidados para hipertensão dividido em duas etapas: etapa de diagnóstico e etapa de tratamento. Sendo assim, na etapa de diagnóstico os dados fisiológicos de um paciente seriam analisados, e, ocorrido algum problema, a etapa de tratamento seria executada.

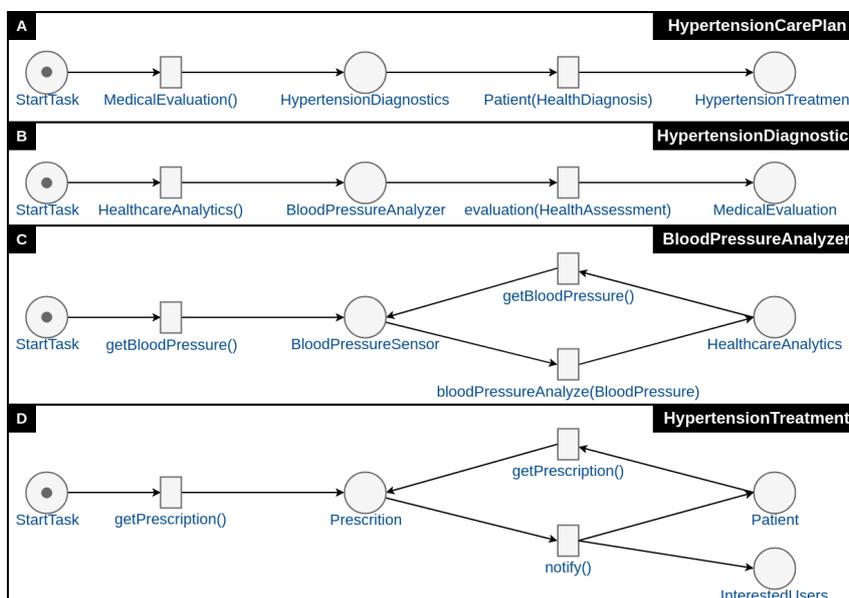


Figura 4. Exemplo de *workflow* de um Plano de Cuidados para Hipertensão.

Alguns elementos típicos de ambientes de computação ubíqua podem ser encontrados em recursos que compõem os *workflows* da Figura 4, tais como, elementos utilizados para aquisição de dados sensoriais (e.g., Serviço *BloodPressureSensor*) e elementos utilizados para realizar notificações de eventos (e.g., Serviço *Prescription*). O ponto comum entre esses elementos na plataforma é que todos eles são representados por meio de serviços. Isso permite a realização de fusões de dados de entidades de diferentes tipos, desde que haja uma compatibilidade entre as *interfaces* dos serviços que as representam.

Na Figura 5 é apresentada a estrutura de um *script* da DSL que corresponde à representação gráfica do primeiro *workflow* (*HypertensionCarePlan*) da Figura 4. No *script* um *workflow* é composto por três tarefas (*tasks*), onde a primeira delas (*start task*) representa o ponto de partida do *HypertensionCarePlan*, o qual realiza uma chamada ao endpoint *MedicalEvaluation* do serviço *HypertensionDiagnostics*. Uma vez que a chamada (*call*) da *start task* é realizada, o fluxo de controle do *workflow* é direcionado para a tarefa seguinte para receber o retorno do serviço *HypertensionDiagnostics*. Como os tipos de dados de *input* e *output* dos serviços são registrados na plataforma, quando o dado de retorno é recebido pelo *workflow* (*onReceive*), é verificado nos *cases* (a partir dos operadores *isTrue* e *isFalse*) o valor do atributo do tipo *boolean* chamado *hasProblem*. Caso o segundo *case* seja satisfeito, o *workflow* *HypertensionCarePlan* é finalizado. Se por outro lado, o primeiro *case* for satisfeito, é realizada uma chamada ao endpoint *Patient* do serviço *HypertensionTreatment* e o fluxo de controle do *workflow* é direcionado para a tarefa seguinte, a fim de receber o retorno desse serviço. Dessa forma, ao receber o dado de retorno (*onReceive*) *Prescription*, se o valor do atributo *endTreatment*, verificado pelo operador da linguagem *currentDateEqual* possuir a data atual, o *workflow* *HypertensionCarePlan* é finalizado.

```

workflow HypertensionCarePlan {
  start task {
    call MedicalEvaluation()->HypertensionDiagnostics
  }

  task HypertensionDiagnostics {
    case onReceive HealthDiagnosis {
      with isTrue(HealthDiagnosis.hasProblem) {
        call Patient(HealthDiagnosis)->HypertensionTreatment
      }
    }
    case onReceive HealthDiagnosis {
      with isFalse(HealthDiagnosis.hasProblem) {
        end.workflow
      }
    }
  }

  task HypertensionTreatment {
    case onReceive Prescription {
      with currentDateEqual(Prescription.endTreatment) {
        end.workflow
      }
    }
  }
}

```

Figura 5. Exemplo de *workflow* modelado a partir de um *script* da DSL da WfMP.

## 5. Conclusão

Com o objetivo de viabilizar a construção de serviços de saúde especializados, visando possibilitar o desenvolvimento de sistemas computacionais de saúde personalizados para cada paciente, a *Workflow Management Platform – WfMP* oferece um conjunto de ferramentas para modelagem, execução e gerenciamento de *workflows*. A proposta da plataforma consiste em permitir que especialistas do domínio de saúde, diante de um conjunto de recursos computacionais, tenham condições de escolher aqueles adequados para o auxílio no tratamento de pacientes em um dado momento.

Neste artigo é apresentada uma arquitetura em que usuários de diferentes perfis (e.g., profissionais de saúde e profissionais de computação) na plataforma têm diferentes perspectivas em relação à forma em que ocorre as orquestrações. Enquanto profissionais

de computação têm acesso, por exemplo, a APIs de manipulação de informações relacionadas a protocolos de comunicação e configurações de serviços, os profissionais de saúde têm acesso às informações dos serviços registrados, as formas de utilizá-los e a uma DSL para compor tarefas de um plano de cuidados a partir desses serviços. No artigo também são mencionados alguns aspectos inovadores da WfMP ao usar a abordagem de *workflows* como serviços para possibilitar a concepção de composições de serviços especializados de saúde a partir de serviços mais básicos, tais como, extensibilidade e reutilização de serviços em tempo de execução.

A plataforma apresentada está em estágio de finalização, sendo que seus principais componentes estão implementados e testados. Embora esses componentes foram desenvolvidos em um projeto de pesquisa multidisciplinar em um laboratório de informática em saúde que conta com pesquisadores de computação e enfermagem, alguns aspectos no que se refere ao uso da linguagem ainda precisam ser avaliados. Desde o início do projeto, há a consciência do desafio de se construir uma linguagem para o domínio de saúde que possa refletir o encadeamento de tarefas de um plano de cuidados e que, ao mesmo tempo, tenha uma sintaxe e semântica adequadas ao entendimento de pessoas que não são da área de computação. Dessa forma, algumas avaliações de usabilidade da DSL com profissionais de saúde ainda são etapas a serem realizadas. O projeto também prevê o desenvolvimento de uma DSL gráfica que seja semanticamente equivalente à DSL textual.

A WfMP é uma plataforma em potencial para realização de testes de integração de microsserviços de saúde candidatos a compor funcionalidades de planos de cuidados em ambientes de computação ubíqua. Dessa forma, poderão ser analisados cenários com componentes que automatizam tarefas de um plano de cuidados adaptável/reconfigurável a partir de *workflows* concebidos e executados na plataforma.

## Referências

- Abidi, S. S. and Chen, H. (2006). Adaptable personalized care planning via a semantic web framework. In *20th International Congress of the European Federation for Medical Informatics (MIE 2006)*, Maastricht. Citeseer.
- Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.
- Benyoucef, M., Kuziemsy, C., Afrasiabi Rad, A., and Elsabbahi, A. (2011). Modeling healthcare processes as service orchestrations and choreographies. *Business Process Management Journal*, 17(4):568–597.
- Browne, E. D., Schrefl, M., and Warren, J. R. (2004). Goal-focused self-modifying workflow in the healthcare domain. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, pages 10–pp. IEEE.
- Caceres, R. and Friday, A. (2012). Ubicomp systems at 20: Progress, opportunities, and challenges. *IEEE Pervasive Computing*, 11(1):14–21.
- Caffrey, C., Sengupta, M., Moss, A., Harris-Kojetin, L., and Valverde, R. (2011). Home health care and discharged hospice care patients: United states, 2000 and 2007. *Natl Health Stat Report*, 38:1–27.
- Carvalho, S. T., Copetti, A., and Loques Filho, O. G. (2011). Sistema de computação ubíqua na assistência domiciliar à saúde. *Journal Of Health Informatics*, 3:2.

- Conti, M., Das, S. K., Bisdikian, C., Kumar, M., Ni, L. M., Passarella, A., Roussos, G., Tröster, G., Tsudik, G., and Zambonelli, F. (2012). Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber–physical convergence. *Pervasive and Mobile Computing*, 8(1):2–21.
- Deen, M. J. (2015). Information and communications technologies for elderly ubiquitous healthcare in a smart home. *Personal and Ubiquitous Computing*, 19(3-4):573–599.
- Deng, M., Petkovic, M., Nalin, M., and Baroni, I. (2011). A home healthcare system in the cloud—addressing security and privacy challenges. In *2011 IEEE 4th International Conference on Cloud Computing*, pages 549–556. IEEE.
- ElHelw, M., Pansiot, J., McIlwraith, D., Ali, R., Lo, B., and Atallah, L. (2009). An integrated multi-sensing framework for pervasive healthcare monitoring. In *Pervasive Computing Technologies for Healthcare, 2009. 3rd Int. Conference on*, pages 1–7. IEEE.
- Fowler, M. (2010). *Domain-specific languages*. Pearson Education.
- Germano, E., Battisti, D., Ribeiro, H. A., and Carvalho, S. T. (2016). Plano de cuidados ubíquo para acompanhamento domiciliar de pacientes. In *Congresso Brasileiro de Informática em Saúde (CBIS)*, pages 849–858.
- Germano, E., Silvestre, B., Ribeiro, H. A., and Carvalho, S. T. (2018). Workflow management platform for orchestration of ubiquitous care plan services. In *Proceedings of the Euro American Conference on Telematics and Information Systems*, page 42. ACM.
- Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., Altmann, J., and Retschitzegger, W. (2003). Context-awareness on mobile devices—the hydrogen approach. In *36th annual Hawaii international conference on system sciences, 2003. Proceedings of the*, pages 10–pp. Citeseer.
- Horta, W. d. A. (1979). Processo de enfermagem. In *Processo de enfermagem*. EPU.
- Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.
- Palesandro, A., Guegan, C. G., Lacoste, M., and Bennani, N. (2016). Overcoming barriers for ubiquitous user-centric healthcare services. *IEEE Cloud Computing*, 3(6):64–74.
- Sabaté, E. (2003). *Adherence to long-term therapies: evidence for action*. World Health Organization.
- Salimifard, K. and Wright, M. (2001). Petri net-based modelling of workflow systems: An overview. *European journal of operational research*, 134(3):664–676.
- Tan, W., Zhou, M., Zhou, M., Li, H., and Weijnen, M. (2015). Service oriented workflow systems. In *Contemporary Issues in Systems Science and Engineering*, pages 645–660. Wiley/IEEE Press.
- Vieira, M. A. and Carvalho, S. T. (2016). (meta)modelagem de espaços inteligentes pessoais e espaços inteligentes fixos para aplicações ubíquas. In *Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SBCUP)*, pages 1056–1065.
- Weiser, M. (1991). The computer for the 21st century—scientific american special issue on communications. *Computers, and Networks*, pages 94–104.