

openEHR

Modelação de
processos
e de decisão lógica

Agenda de conteúdos

Módulo 1: Introdução à modelação de processos e decisão lógica

Módulo 2: *Task Planning*

Módulo 3: *Decision Logic Model*

Módulo 1: Introdução à modelação de processos e decisão lógica

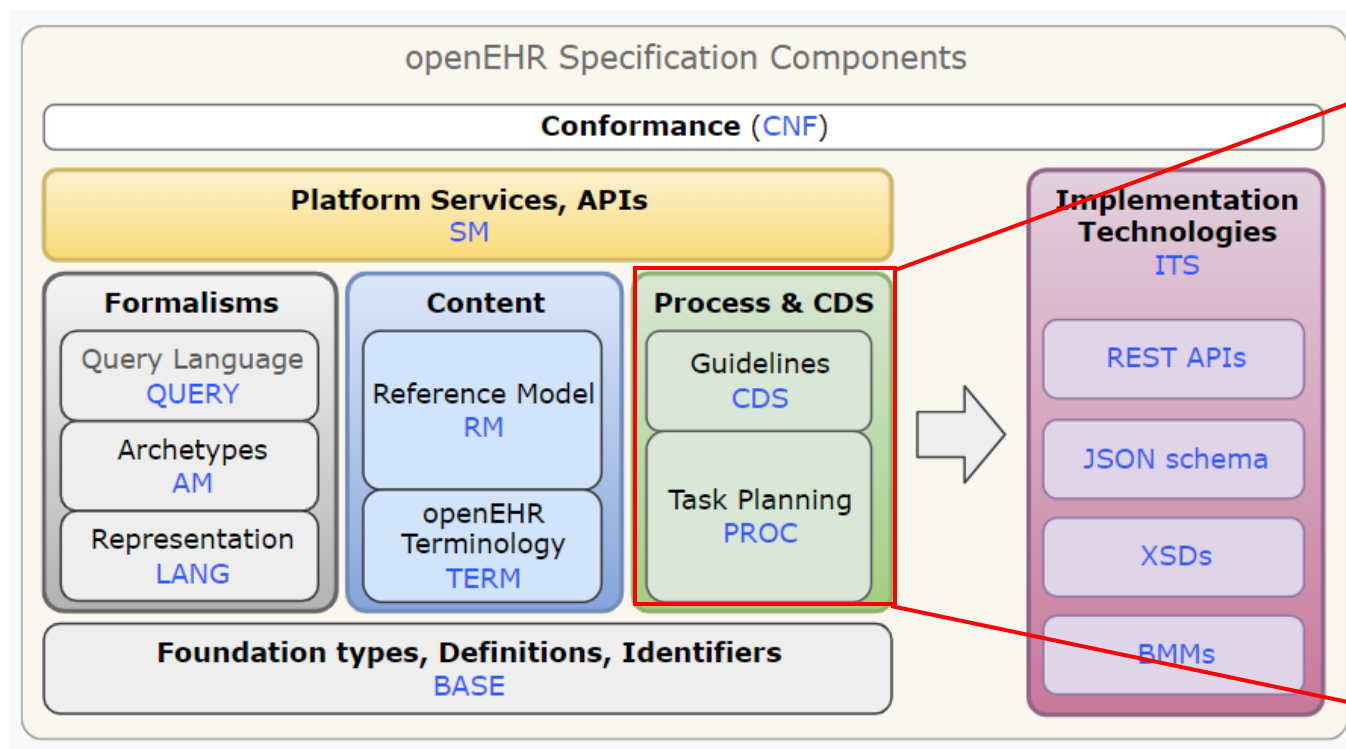
CONTEÚDO

- 1.1: Introdução
- 1.2: Enquadramento
- 1.3: Formalização dos processos
- 1.4: Paradigma de implementação
- 1.5: Arquitetura conceitual



1.1: Introdução

Especificações openEHR: processo e planeamento



- ⌚ Overview
- ⌚ Task Planning (TP)
- ⌚ TP Visual Modeling Language (TP_VML)
- ⌚ Decision Language (DL)
- ⌚ Process CDS, Guidelines and Planning Examples

1.1: Introdução

Especificações openEHR: processo e planeamento



Formalismo de definição de *planos*: [openEHR Task Planning specification \(TP\)](#);



Linguagem de *guidelines* e regras: [openEHR Decision Language specification \(DL\)](#);



Uma *linguagem de expressão*, que forma a base para escrever expressões e regras, definida pela [openEHR Expression Language \(EL\)](#);



Um serviço de *subject proxy*, que forma a ponte entre os *Plans* e as *Guidelines* como consumidores de dados do sujeito, e as fontes de dados: [openEHR Subject Proxy Service](#).



Um metamodelo subjacente que formaliza modelos de informação, linguagem de expressão e construções de decisão definido pelo [openEHR Basic Meta-Model](#);

1.1: Introdução

Relação com outros standards

FORMALISMO OPENEHR	PADRÃO HL7	PADRÃO OMG	OUTROS
Task Planning (TP)	FHIR	BPMN, CMMN, BPM+	YAWL
Decision Language (DL)	CQL, FHIRpath?	DMN	Arden, ProForma
Expression Language (EL)	CQL ELM	DMN FEEL	(numerous)
Subject Proxy Service (SPS)		(SDMN)	

1.2: Enquadramento

Melhoria de Cuidados Clínicos com Tecnologias de Informação (TI)



As tecnologias de informação não vão substituir o trabalho humano, mas auxiliar na lógica de planeamento e decisão.

- Suporte para *workflows*, tomada de decisões, lembretes e documentação;
- Abordagem das fragilidades humanas: capacidade para lidar com a complexidade dos circuitos, adaptação a mudanças, prevenção do esquecimento de passos rotineiros;
- Necessidade de suporte automatizado em situações de cuidado em equipa e de longo prazo.

1.2: Enquadramento

Implementação das melhores práticas

Formalização das melhores práticas: *care pathways*, *guidelines* de prática clínica...

Uso dos benefícios de recursos computacionais para uma fácil integração em aplicações clínicas

Simulação, teste e verificação formal para uma adoção mais rápida de melhores práticas

1.2: Enquadramento

Modelo cognitivo

Paradigma do copiloto

Os sistemas de planeamento e suporte à decisão na saúde desempenham um papel de copilotos, proporcionando assistência sem substituir o processamento cognitivo dos profissionais. Semelhante a um sistema de navegação, adaptam-se a desvios, recalculando trajetórias conforme necessário. O resultados de regras são considerados recomendações, sujeitas a anulação, com a opção de justificação.

1.2: Enquadramento

Modelo cognitivo – Integração com EHR

Em sistemas de fluxo de trabalho genéricos, a falta de um sistema dedicado para registar informações detalhadas sobre um sujeito pode limitar a eficiência.

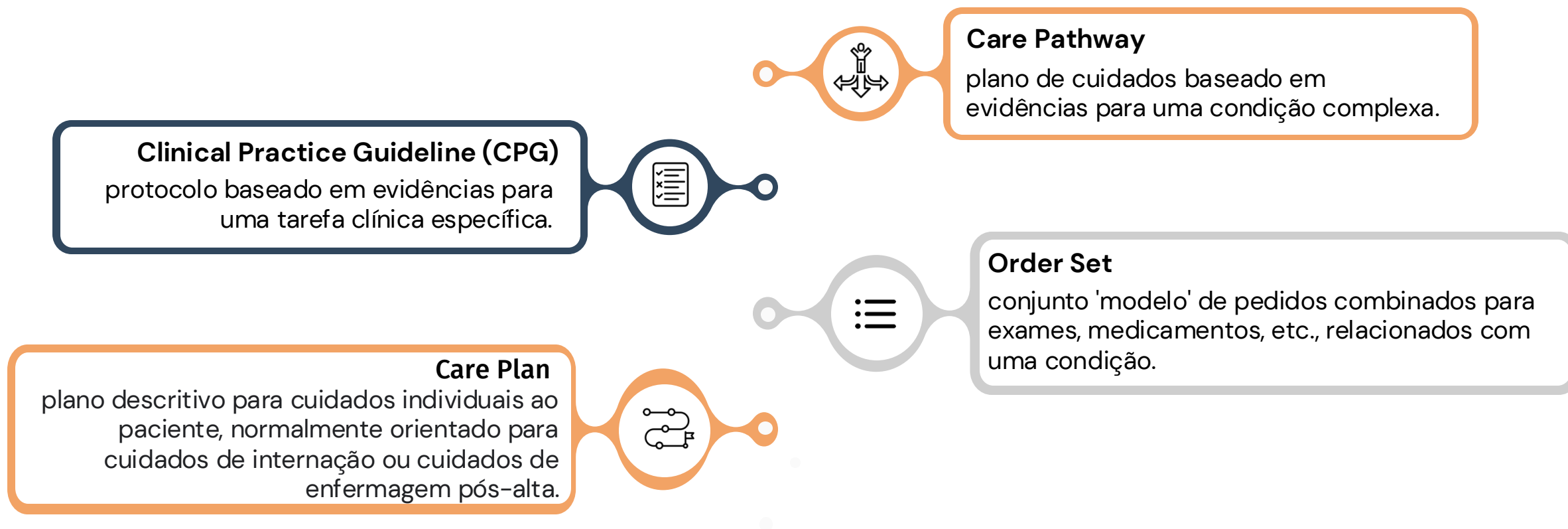
Planos clínicos demandam representações formais que acomodem dados precisos e descrições detalhadas de ações.

A integração estreita entre interações de registos de pacientes e tarefas é crucial para reduzir a carga documental dos clínicos.

A realização ideal da automação em saúde incluiria conjuntos de dados padronizados, mas desafios práticos requerem estratégias flexíveis para integração em ambientes diversos, garantindo associações eficazes entre informações de tarefas e registos de pacientes.

1.3: Formalização dos processos

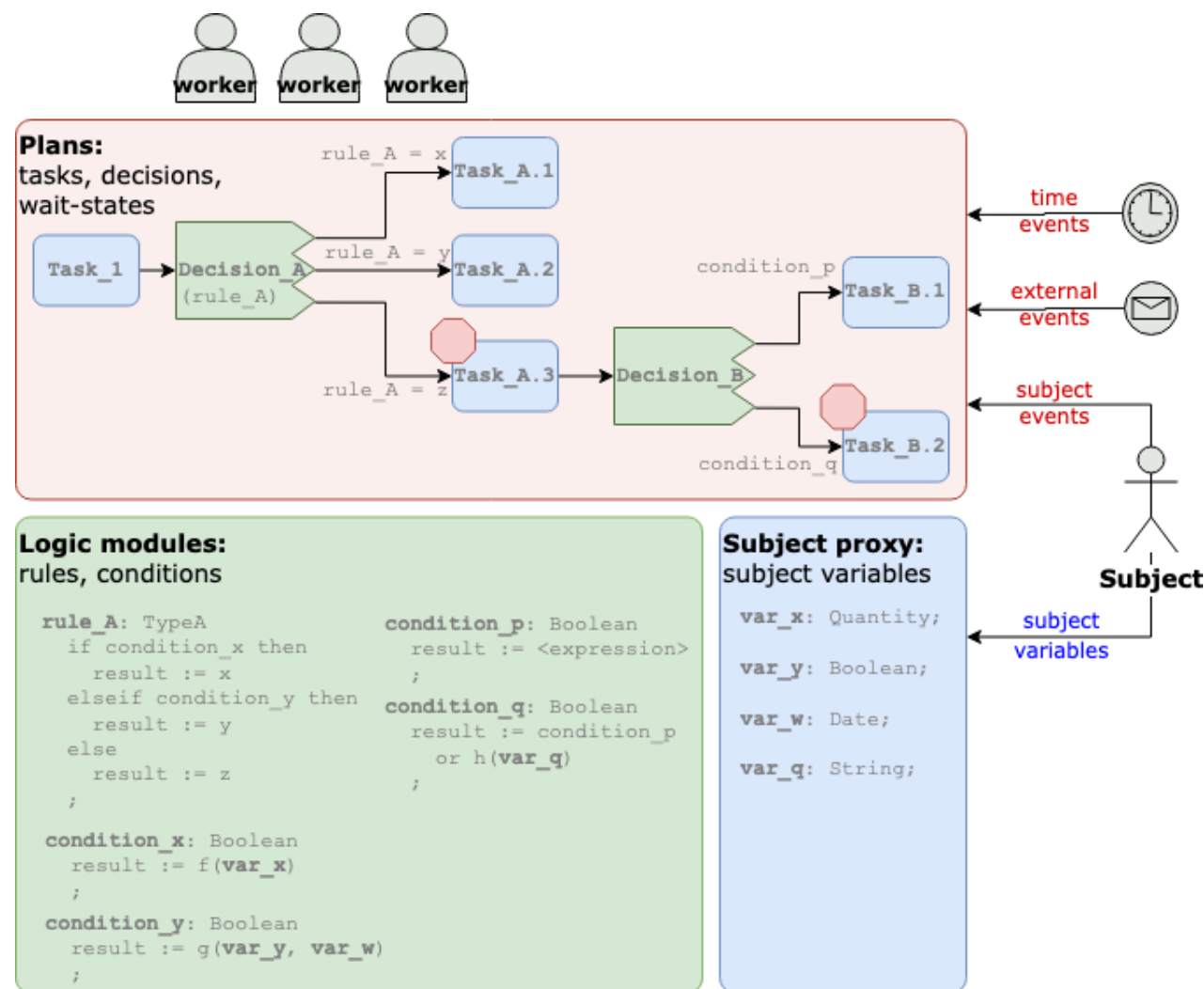
Artefactos formalizáveis na área da saúde



1.3: Formalização dos processos

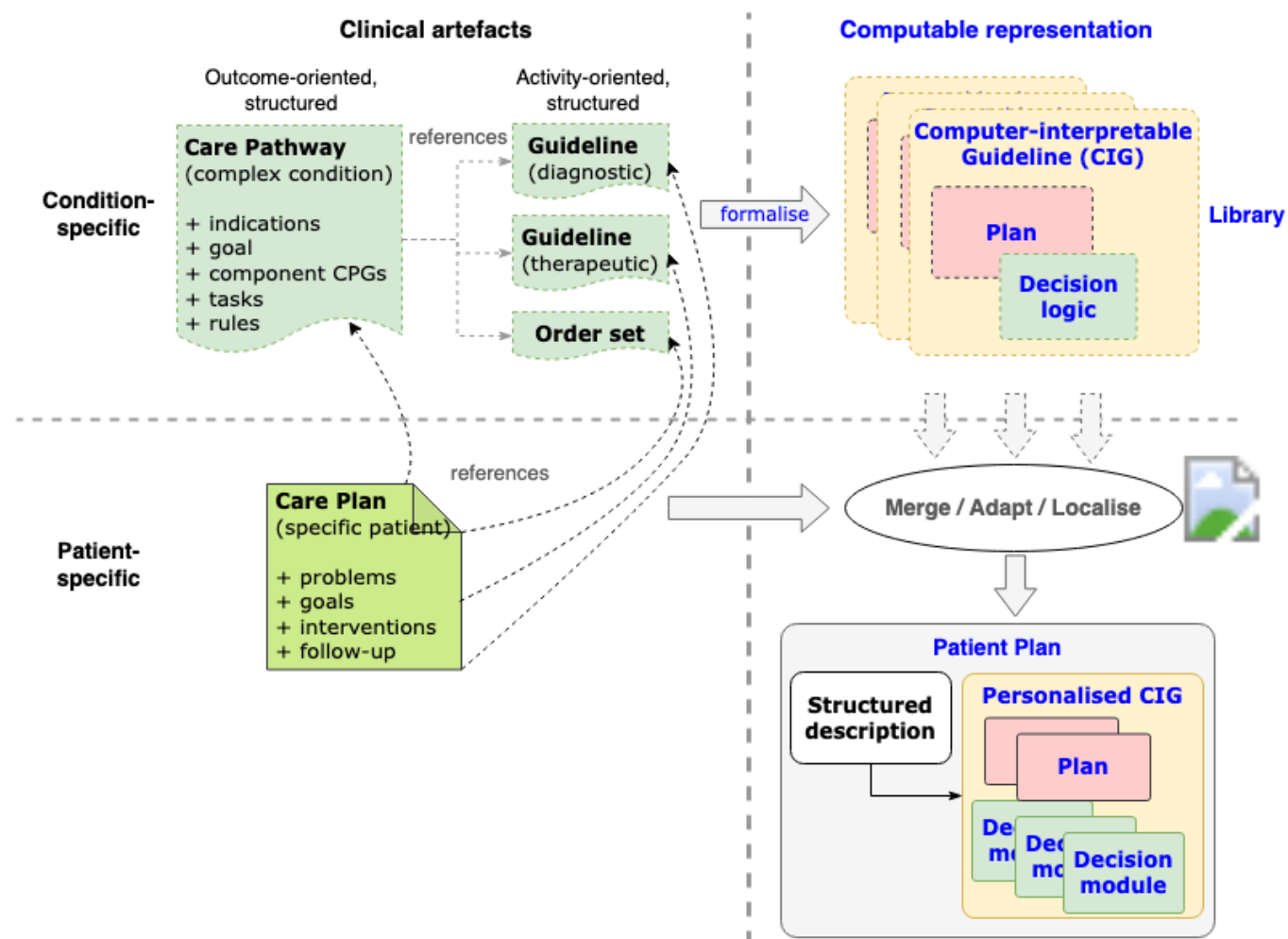
Modelo conceptual das *guidelines*

- **planos** que consistem em:
 - *tarefas*;
 - *decisões*;
 - *condições de espera*.
- **módulos lógicos** contendo:
 - *condições*;
 - *regras*.
- **subject proxy**

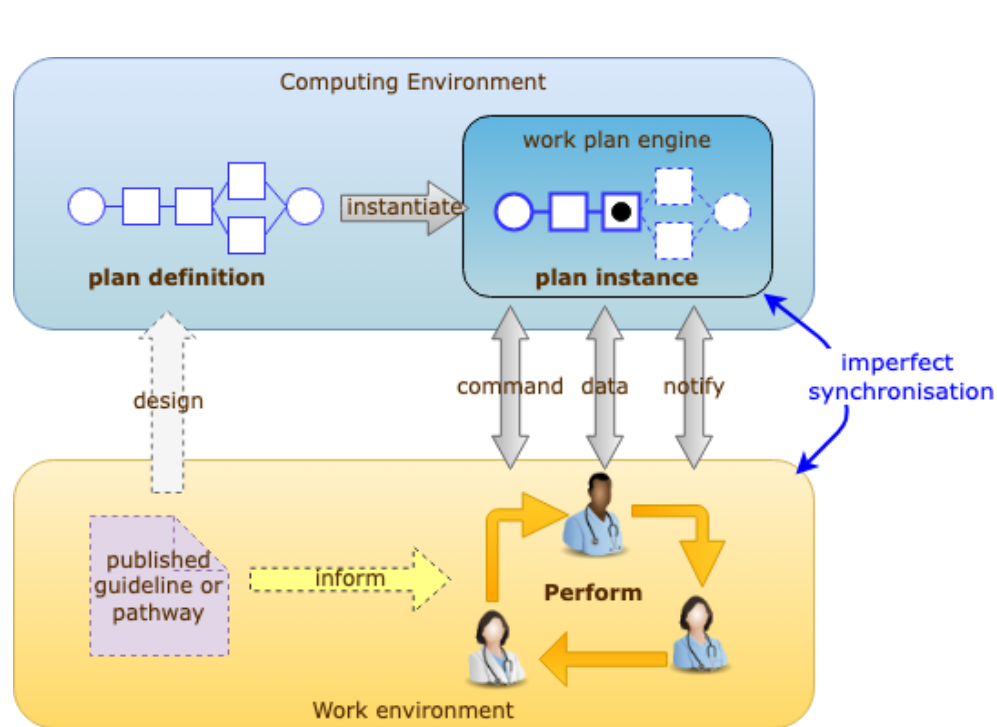


1.3: Formalização dos processos

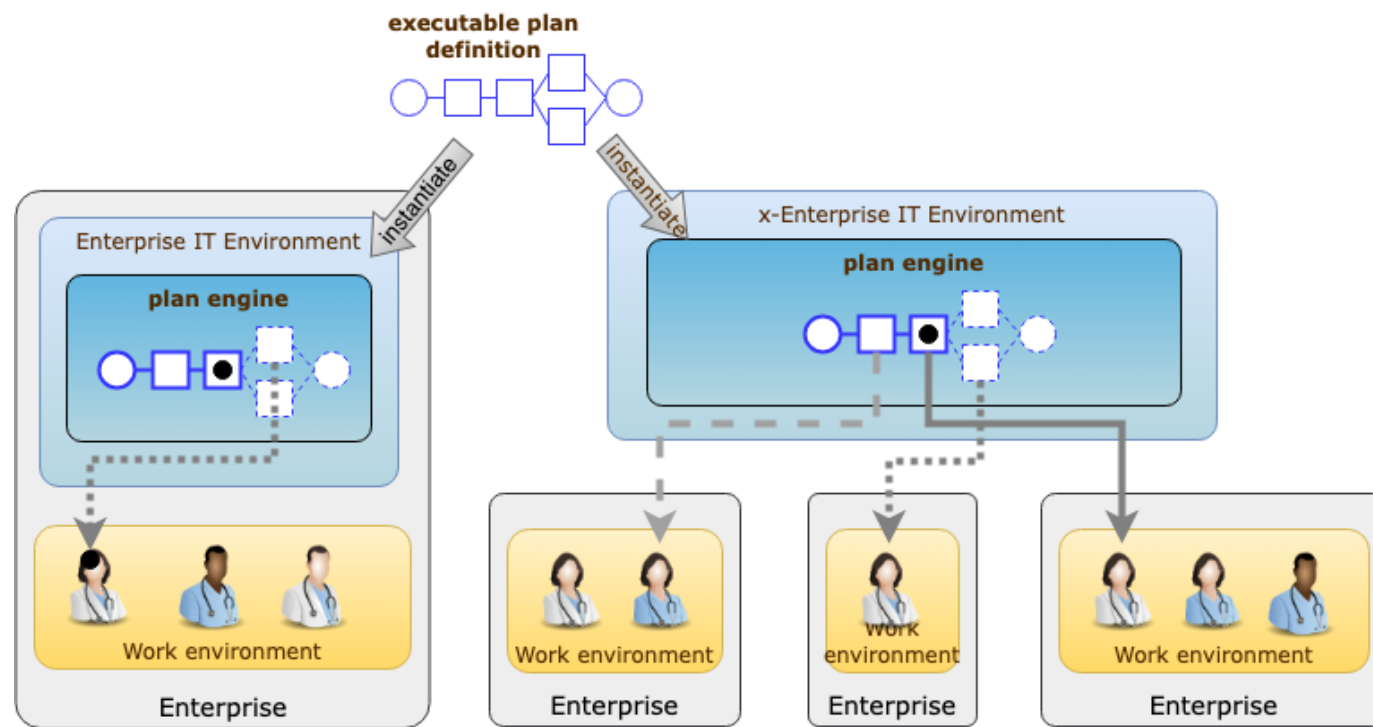
Visão geral



1.4: Paradigma de implementação



Paradigma de execução dos planos



Planos distribuídos

1.5: Arquitetura conceitual

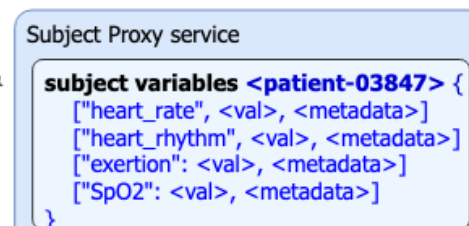
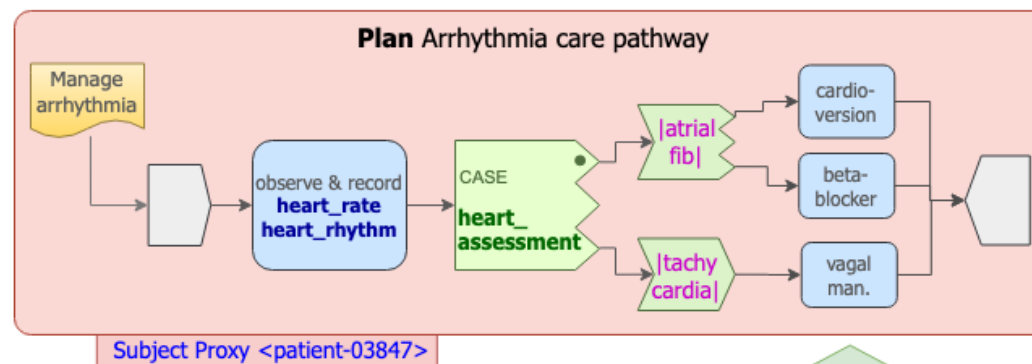
Plan definition:
structured plans
based on
guidelines,
care pathways

Subject proxy:
subject state &
events (ontic view)

Subject proxy service:
populate variables &
events of subject
- currency
- type conversion
- caching

Data access binding:
bind proxy
variables to
data sources

Data sources:
EHR, messages,
devices,
live users
(epistemic
resources)



DLM openEHR.cardiology_basic.v1.0.4

input

heart_rate: Quantity
heart_rhythm: Terminology_term
exertion: Terminology_term

reference

Afib_heart_rate: Quantity = 120 /min
High_heart_rate: Quantity = 100 /min

conditions

has_atrial_fibrillation:
heart_rhythm = [erratic] and
heart_rate >= Afib_heart_rate

has_tachycardia:

heart_rate >= High_heart_rate and
exertion = [at rest]

rules

heart_assessment: Terminology_term

=====

has_atrial_fibrillation: [atrial fibrillation]

has_tachycardia: [tachycardia]

=====

*: [normal]

=====

Decision logic module:
conditions & rules,
guideline logic

Módulo 2: *Task Planning*

CONTEÚDO

- 2.1: Introdução
- 2.2: Princípios de design
- 2.3: Implementação
- 2.4: Ferramentas
- 2.5: Exemplos

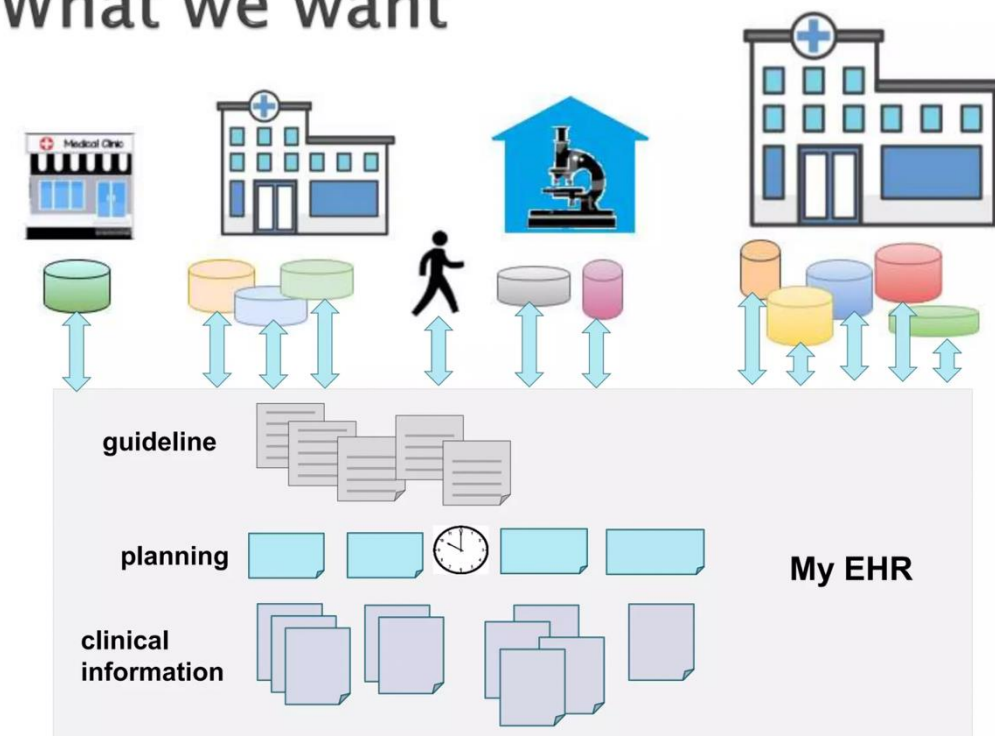


2.1: Introdução

O que os profissionais de saúde querem?

- Um co-piloto para as guidelines;
- Uma visão do processo;
- Em que parte do processo nos encontramos?
- Quem está responsável pelo doente atualmente?
- Todas as requisições e estado das intervenções;
- Todos os exames, notas, resultados (EHR).

What we want



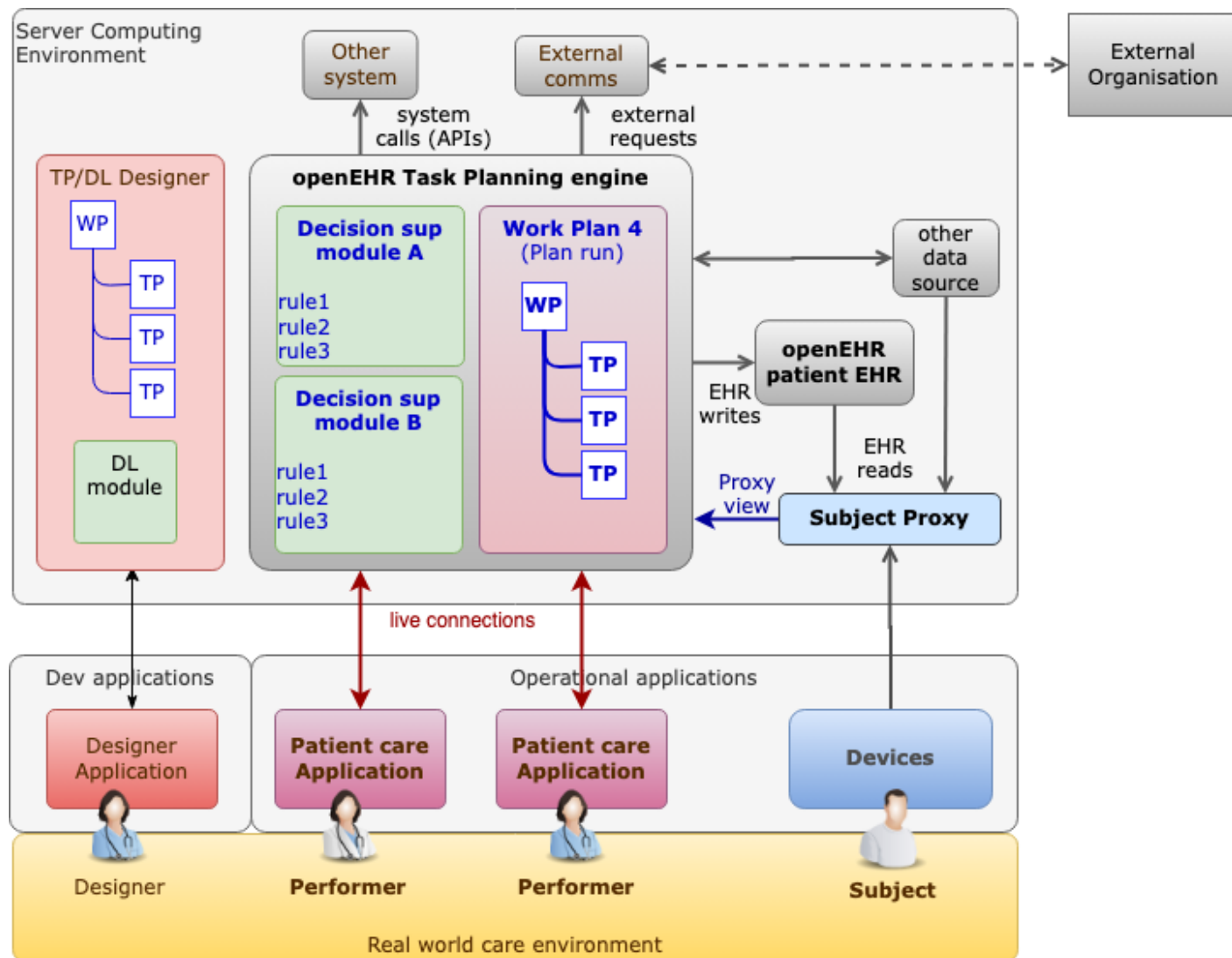
2.1: Introdução

Requisitos técnicos

- planos clínicos de longo prazo;
- lembretes/itens de *checklist* para ações básicas que são recorretamente perdidas ou esquecidas;
- conjuntos complexos de ações definidas por percursos clínicos;
- pontos de decisão que determinam qual caminho seguir;
- ações que exigem aprovação;
- coordenação de trabalhadores em equipas distribuídas;
- ações que resultem no registro de algo no EHR;
- ações de vários níveis de granularidade.

2.2: Princípios de design

Contexto computacional



2.2: Princípios de design

Definição vs execução

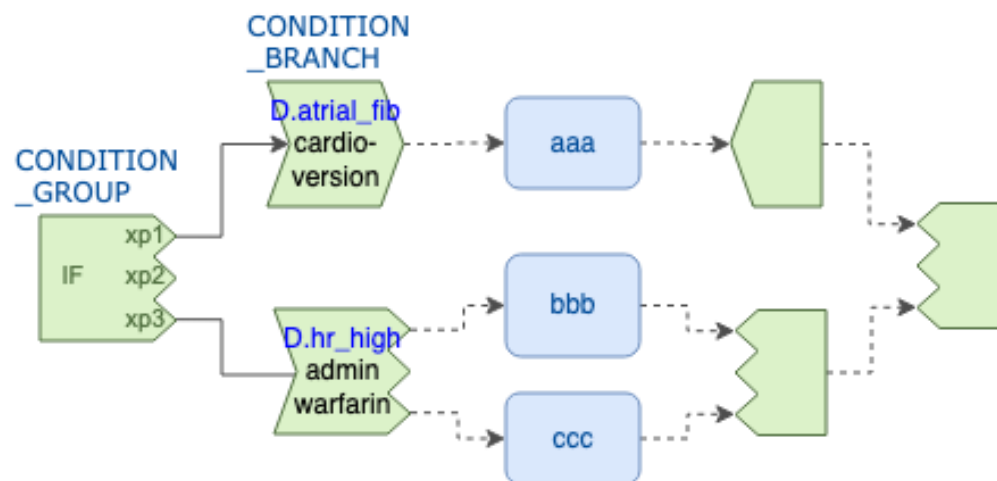
- **Template do plano**, representando *Planos* e *componentes de Plano* reutilizáveis, a partir dos quais podem ser geradas 'definições' de Plano;
- **Definição do plano**, representação concreta de um determinado Plano que pode ser executado diretamente;
- **Execução do plano**, a execução do plano, que pode ser de longa duração, por ex. dias, meses, anos;
- **Execução de tarefas**, a execução em *runtime* de parte(s) de um Plano durante sessões de utilizadores, por meio do uso de aplicações, etc.

2.2: Princípios de design

TP-VML

Linguagem de modelação visual para o modelo de planeamento de tarefas openEHR. Projetado para permitir que representações visuais eficientes de *Task Plans* sejam criadas com ferramentas configuradas adequadamente.

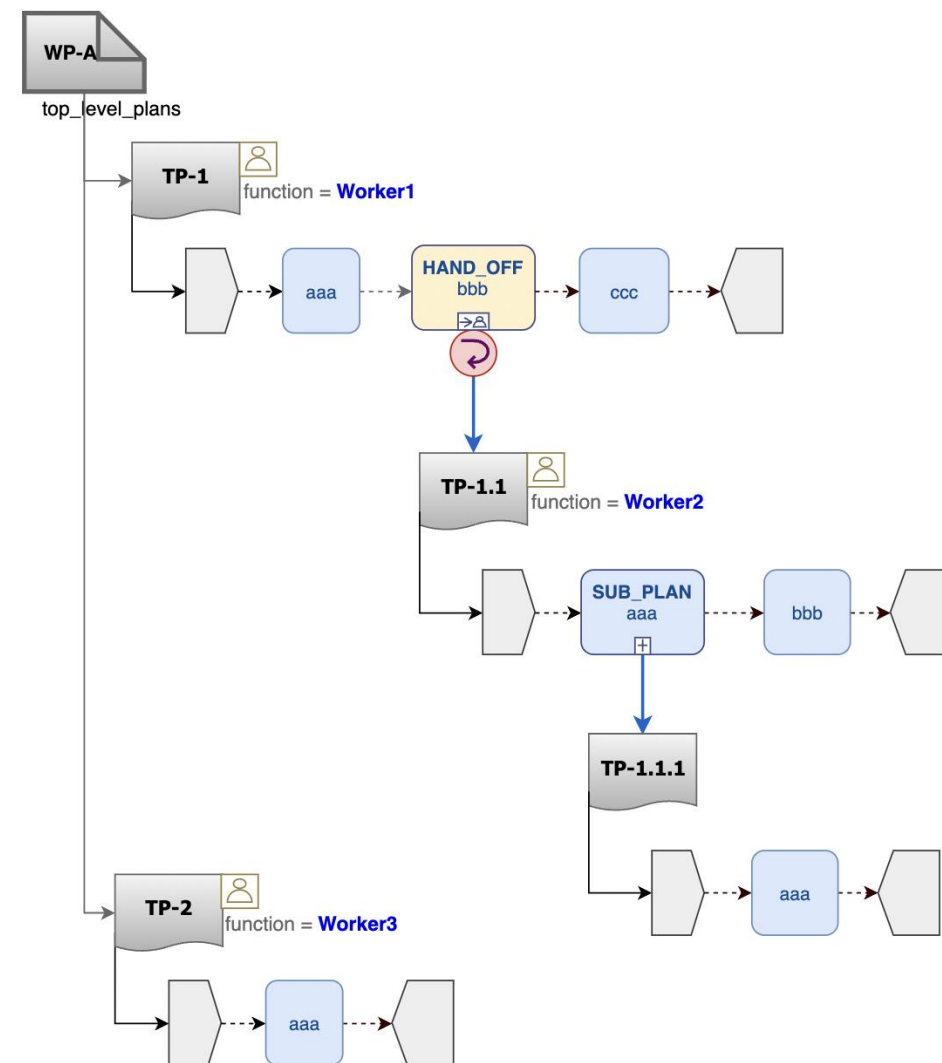
EXEMPLO DE UTILIZAÇÃO:



2.2: Princípios de design

Work Plan (WP)

- Composto por **um ou mais *Task Plans***.
- Define o trabalho a ser executado por um ou mais trabalhadores para alcançar um objetivo específico em relação a um único assunto de cuidado.
- Dentro de um WP, cada *task plan* incluído é uma definição de trabalho num contexto específico, realizado por um *principal performer* e possivelmente outros participantes.



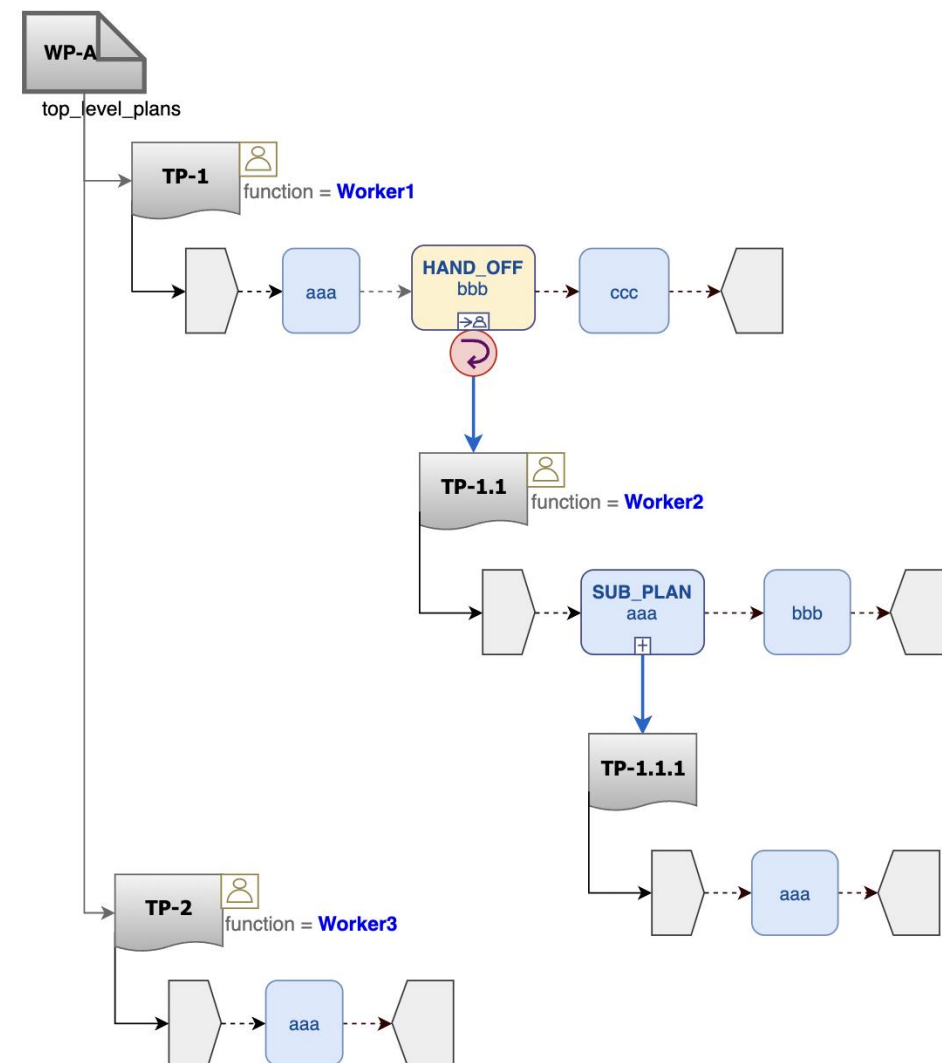
2.2: Princípios de design

Work Plan (WP)

ATRIBUTOS:

- **plans**: referencia todos os *Task Plans*;
- **top_level_plans**: O conjunto de *Task Plans* que estão ativos na ativação do WP;

(...)



2.2: Princípios de design

Work Plan (WP)

ATRIBUTOS:

- **calendar**: um calendário global contendo entradas relacionadas a este *Work Plan*, por ex. compromissos, feriados etc;
- **timeline**: o cronograma global do Plano (e, portanto, do sujeito) no qual as tarefas planeadas são fixadas, com tempos especificados como deslocamentos do ponto zero;
- **event_wait_states**: uma lista de referência de todas as instâncias de espera de *Event* definidas no Plano;
- **order_list**: uma tabela de referências aos pedidos que estão sendo rastreados no Plano;

2.2: Princípios de design

Work Plan (WP)

ATRIBUTOS:

- **context**: um contexto de dados para o Plano como um todo, que permite que variáveis externas (como itens de dados do paciente) sejam rastreadas e atualizadas;

variables

List<CONTEXT_VARIABLE>

- Sinais vitais do paciente;
- Características demográficas;
- “tempo desde evento AVC”;
- (...)

expressions

List<CONTEXT_EXPRESSION>

- $\$bp_systolic - \$bp_diastolic$
- $\$weight / (\$height * \$height)$

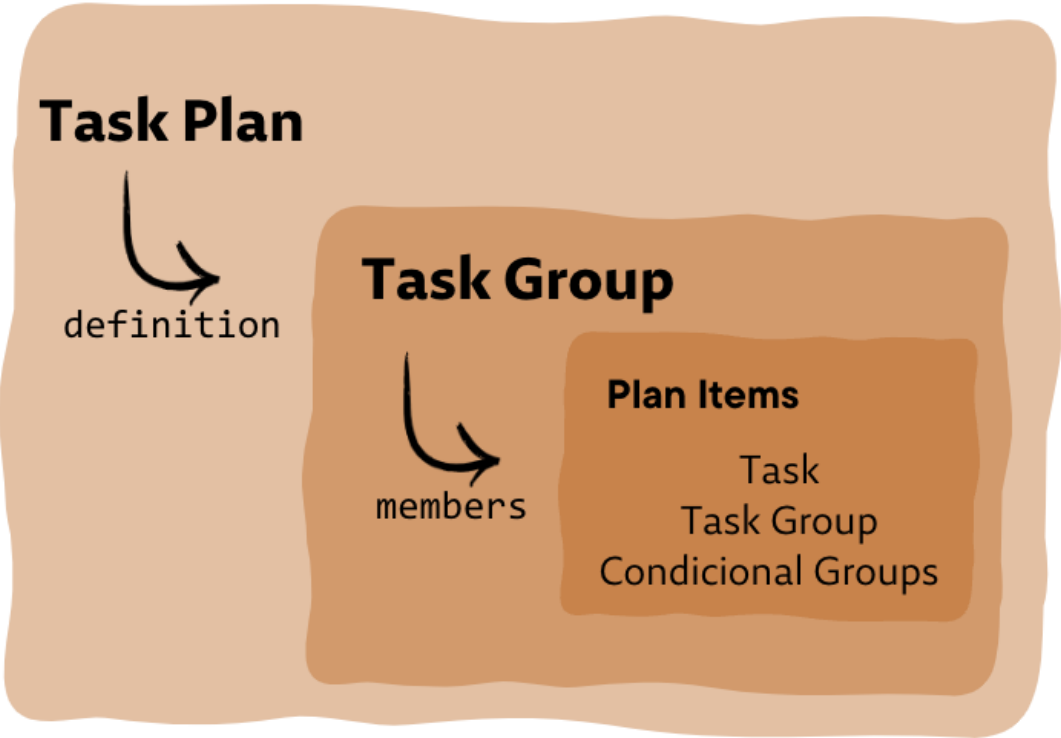
constants

List<CONTEXT_CONSTANT>

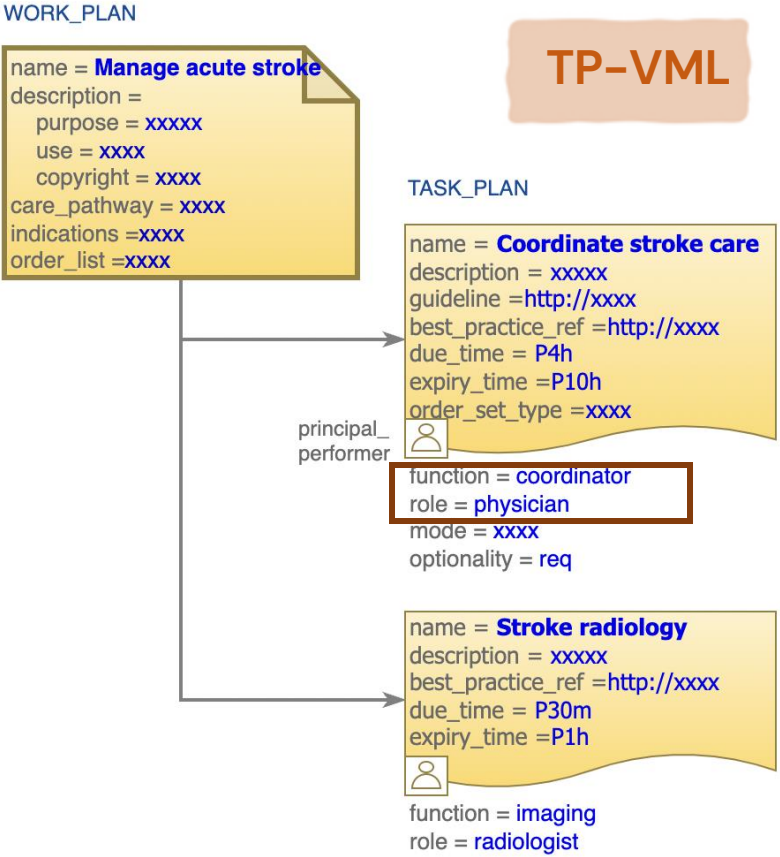
- Dados imutáveis

2.2: Princípios de design

Task Plan



Work Plan,
Task Plan



2.2: Princípios de design

Referências

WORK PLAN

- *care_pathway*
- *care_plan*

TASK PLAN

- *guideline*
- *best_practice_ref*

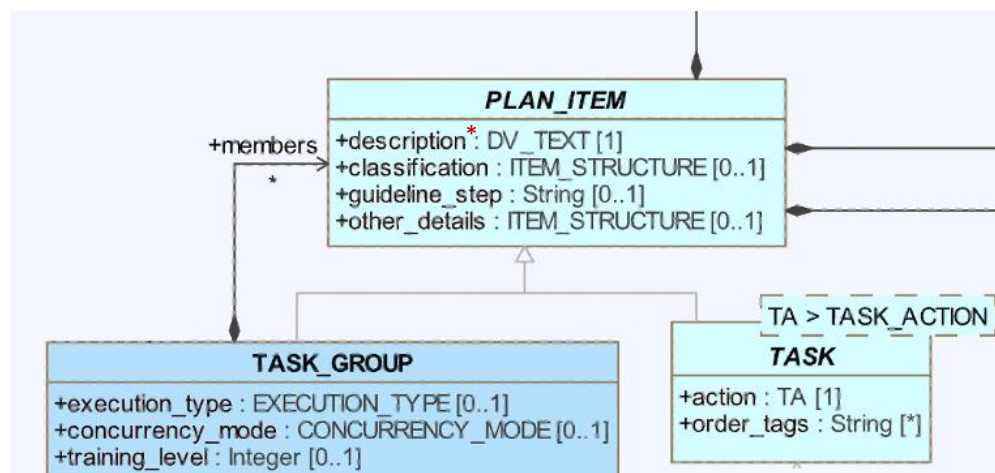
PLAN ITEM

- *guideline_step*

2.2: Princípios de design

Plan Items

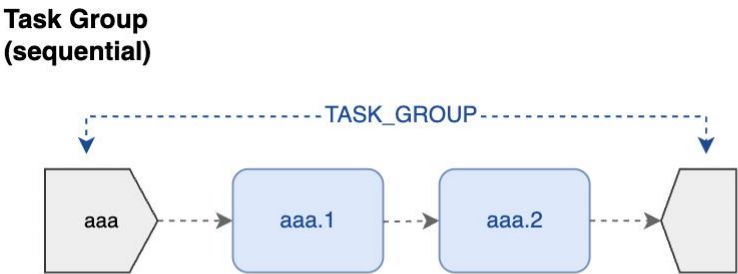
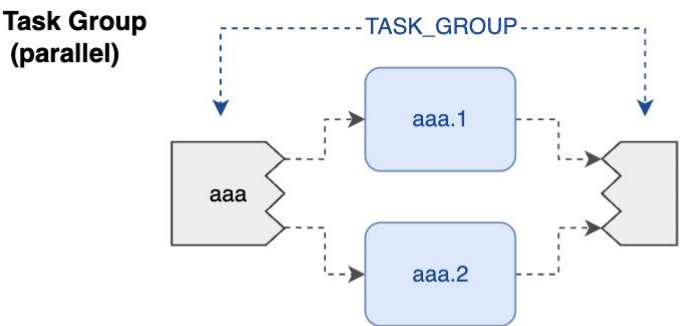
A classe *PLAN_ITEM* é a mãe de todos os elementos refinados de um Task Plan, que são *TASKs* ou *TASK_GROUPS*. Possui um atributo de *description* obrigatório.



2.2: Princípios de design

Plan Items

TASK GROUP



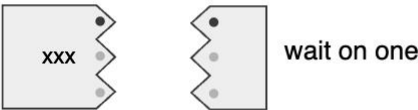
AND logic: follow all



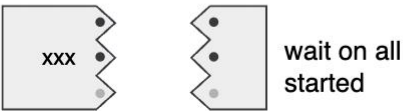
XOR logic: follow one



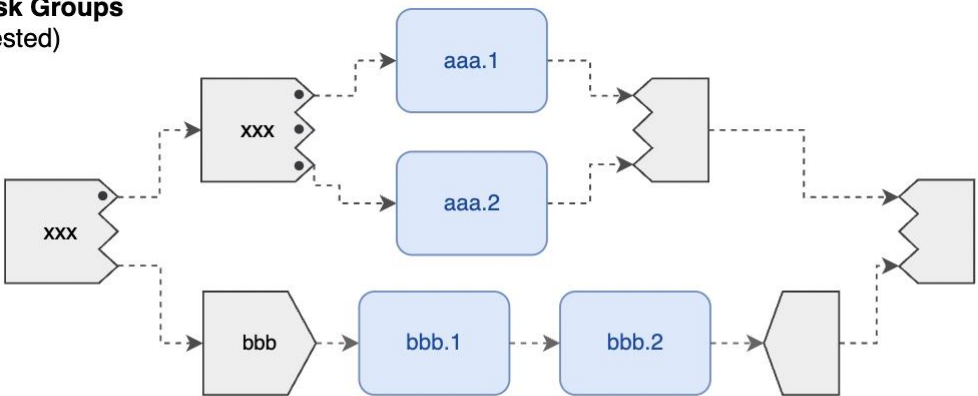
OR logic: follow ≥ 1 ,
complete when first path
started completes



OR logic: follow > 1 ,
complete when all paths
started complete



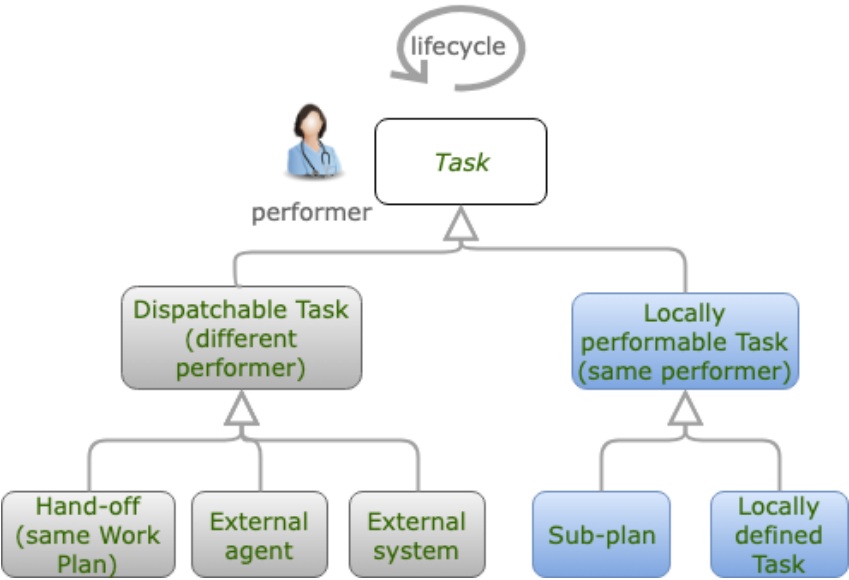
**Task Groups
(nested)**



2.2: Princípios de design

Plan Items

Tasks

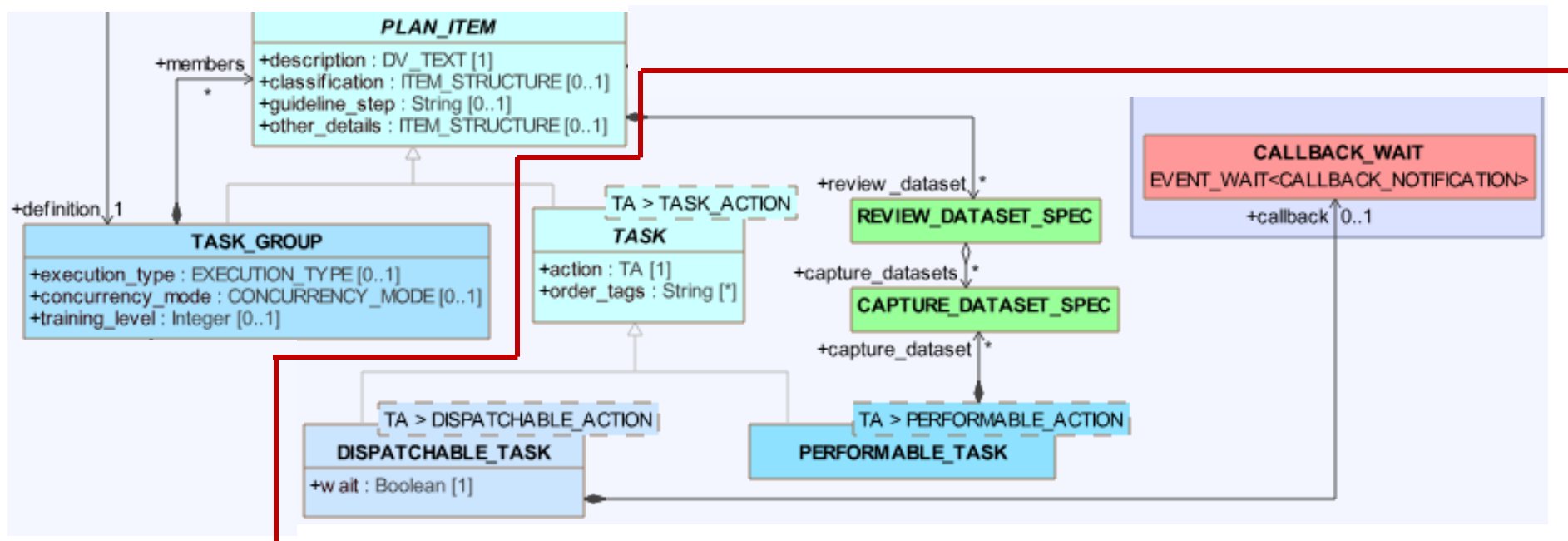


Tipo de Task	Descrição	Lifecycle state	Atributos
Performable	Tarefa executada pelo performer atual	Avança por ações do performer (ator do mundo real)	<i>capture_dataset</i> : Template e/ou formulário de conjunto de dados associado, através do qual os dados a serem capturados durante esta tarefa serão realmente inseridos.
Dispatchable	Uma tarefa cujo trabalho é “despachado” para ser executado por outro performer ou sistema.	<i>wait</i> = True: Avança pelo recebimento de uma notificação <i>callback</i> <i>wait</i> = False: Avança automaticamente para <i>completed</i>	<i>wait</i> : sinalizador que indica se se deve esperar ou continuar o Plano atual após a notificação de despacho ter sido alcançada. <i>callback</i> : gestor opcional da notificação de <i>callback</i>

2.2: Princípios de design

Plan Items

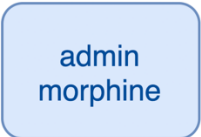
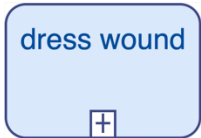
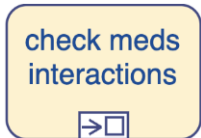
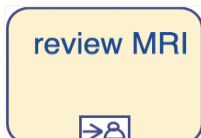

Tasks



2.2: Princípios de design

Plan Items

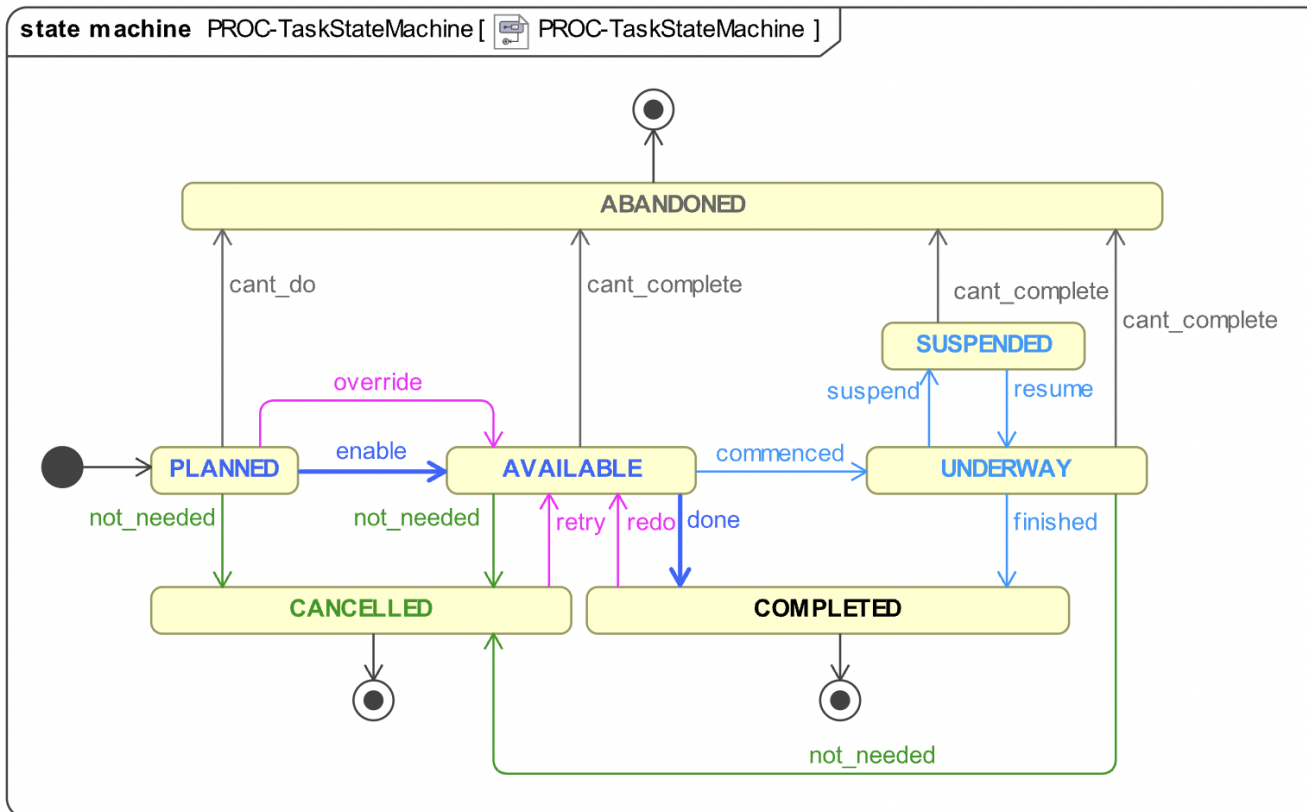
Tasks → Task Actions

Tipo	TP-VML	Descrição	Atributos
DEFINED_ACTION		Tarefa definida em linha, a ser executada pelo <i>principal performer</i> do plano;	<i>prototype</i> : List<ENTRY> <i>optionality</i> : VALIDITY_KIND
SUB_PLAN		Tarefa que representa outro plano a ser executado pelo <i>performer</i> atual	<i>target</i> : TASK_PLAN (herdado do LINKED_PLAN)
SYSTEM_REQUEST		Tarefa que consiste numa solicitação a um sistema computacional, como uma consulta de dados, em nome do <i>performer</i> atual;	<i>system_call</i> : SYSTEM_CALL
HAND_OFF		Tarefa que passa para outro <i>task plan</i> do mesmo WP, tendo um <i>performer</i> diferente.	<i>target</i> : TASK_PLAN (herdado do LINKED_PLAN)
EXTERNAL_REQUEST		Tarefa que consiste num pedido a uma entidade organizacional externa que está fora WP atual e do seu ambiente de execução, em nome do <i>performer</i> atual.	<i>organisation</i> : PARTY_PROXY <i>request_id</i> : String <i>other_details</i> : ITEM_STRUCTURE

2.2: Princípios de design

Plan Items

Lifecycle state



planned – tarefa definida mas não disponível para execução;

available – tarefa pronta para execução;

cancelled – tarefa cancelada antes ou durante execução (não necessária);

abandoned – tarefa abortada por impossibilidade;

underway – tarefa foi iniciada ou retomada do estado *suspended*;

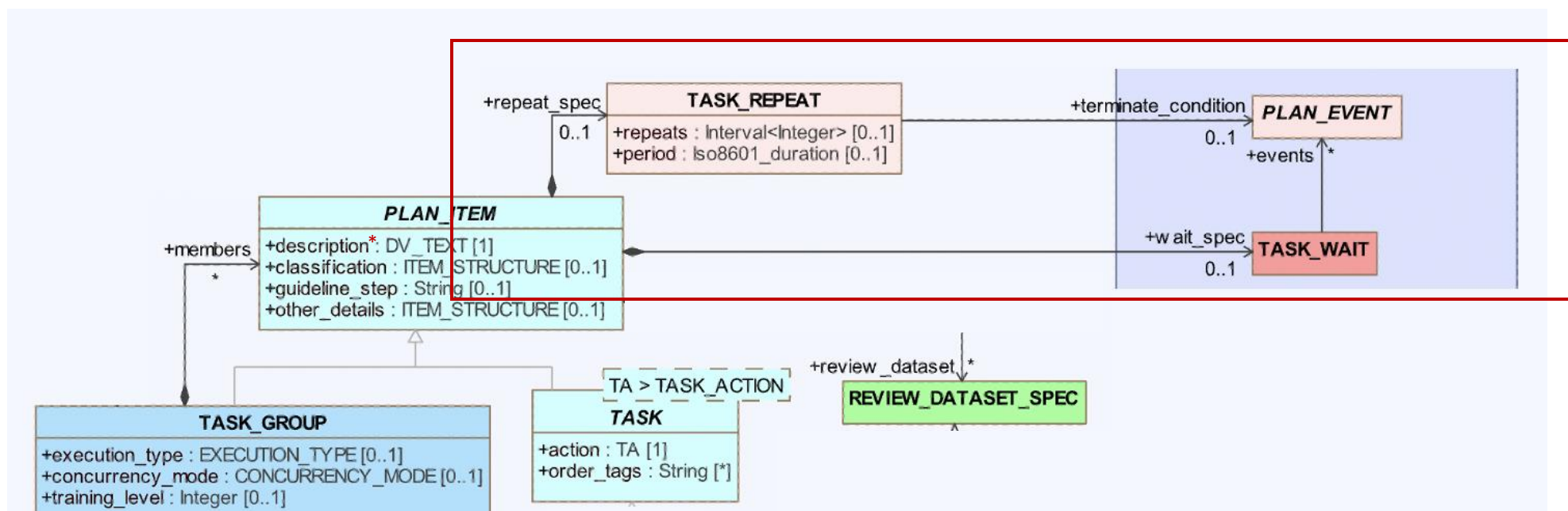
suspended – tarefa colocada em espera;

completed – tarefa concluída da maneira pretendida.

2.2: Princípios de design

Plan Items

Um *PLAN_ITEM* também possui dois atributos opcionais que controlam o comportamento temporal dos elementos do TP: *wait_spec* e *repeat_spec*.

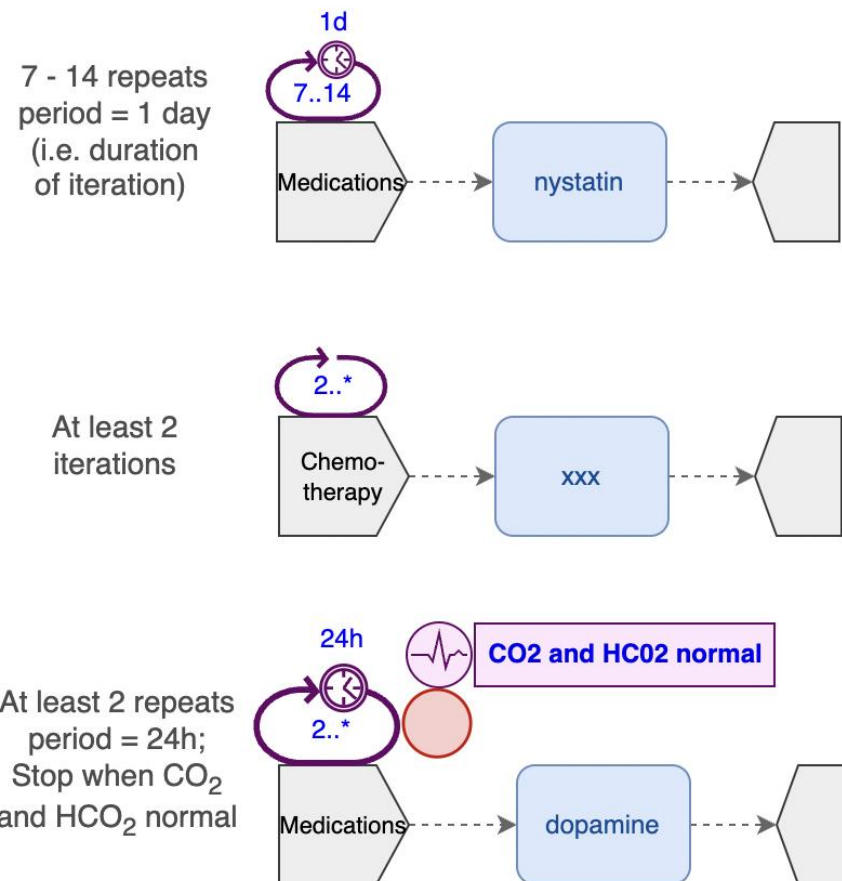


2.2: Princípios de design

Plan Items

REPETIÇÃO:

- **repeats:** Range de repetições. Se o limite inferior for maior que zero, será necessária pelo menos uma repetição. Um intervalo 0..3 significa que até 3 repetições são possíveis, mas não obrigatórias.
- **terminate_condition:** Estabelece uma condição para quando a repetição deve parar. Se definido, é testado antes de cada repetição acontecer, após o número mínimo de repetições ser atingido.
- **period:** Período opcional de repetição, ou seja, duração entre o início de cada clone PLAN_ITEM na forma materializada.



2.2: Princípios de design

Plan Items

TEMPO E ESTADOS DE ESPERA

Desvio Relativo (<i>relative offset</i>)	<i>timer event</i> evento causado por um temporizador que expira	<i>state trigger</i> condição baseada numa ou mais variáveis observadas disponíveis no ambiente de computação	<i>task transition</i> evento gerado pela transição de estado de uma tarefa durante a execução até o momento
Tempo absoluto (<i>absolute time</i>)			
Momento em que ocorre um evento	<i>callback notification</i> evento gerado por uma notificação recebida na conclusão de uma tarefa enviada para um TP diferente ou para um sistema externo	<i>system notification</i> evento associado à notificação de um sistema	<i>manual notification</i> evento externo sinalizado manualmente ao sistema por um utilizador

2.2: Princípios de design

EVENTOS

Não determinísticos

Eventos que podem nunca acontecer

TASK_TRANSITION



task_id = Task-1234
transitions = abandoned, complete

Evento gerado pela transição do ciclo de vida de outra Tarefa, como transição para cancelada ou concluída.

STATE_TRIGGER



expression =
\$systolic_bp > 160.0

Tipo de evento que representa uma expressão referente a uma ou mais variáveis rastreadas, atingindo um valor ou ultrapassando um valor limite.

MANUAL_NOTIFICATION



description =
"CA CT scan result ready"

Um evento que é notificado manualmente ao mecanismo de execução do plano por um utilizador.

SYSTEM_NOTIFICATION



system_id = paed.amsgen.org.nl
notification_type = result ready

Um evento que é notificado ao mecanismo de execução do plano por um sistema.

CALLBACK_NOTIFICATION



Notificação de retorno de chamada conectada a uma *dispatchable task* (transferência, solicitação externa, solicitação do sistema).

Determinísticos

Eventos com ocorrência garantida num momento conhecido

TIMER_EVENT



duration = P1h30m

Evento gerado pela expiração de um *timer* que foi lançado em algum momento anterior

TIMELINE_MOMENT



offset = P2d
fixed_time = 13:45:00

Evento gerado pelo relógio do sistema que atinge um ponto de tempo fixo na linha do tempo do WP, especificado por um deslocamento da origem do WP (substituível) mais um horário fixo opcional no dia.

CALENDAR_EVENT



entry_id = Cal-item-1234
time = 2019-02-15 07:30:00

Evento gerado pelo relógio do sistema que atinge um evento no calendário global do Plano, que é especificado em tempo absoluto, independente do cronograma do WP.

2.2: Princípios de design

Plan Items

EVENTOS

Instâncias de todos estes tipos de eventos, por si só, apenas identificam o tipo e origem de um evento. Eventos podem ser aguardados de duas maneiras:



TASK_WAIT associado a qualquer *Plan Item*, através do atributo *wait_spec*;

instâncias de **EVENT_WAIT<T>**, que é uma classe genérica a partir da qual tipos concretos podem ser construídos com base no tipo de *Event* a ser “escutado”.

2.2: Princípios de design

Plan Items

TASK WAIT

A classe TASK_WAIT modela um estado de espera e define como especificar os detalhes de quando uma tarefa deve começar relativamente a qualquer um dos tipos de eventos descritos anteriormente.

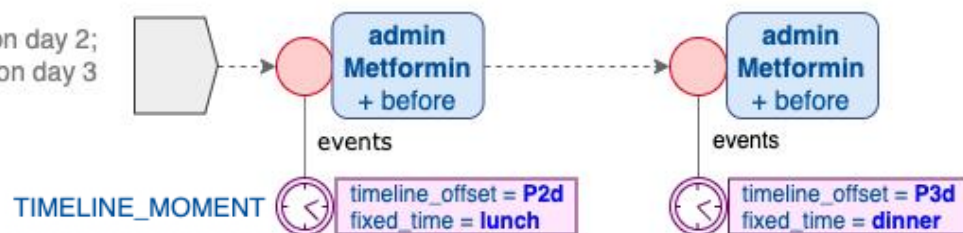
- **events:** lista de evento(s) que define(m) o ponto no tempo pretendido;
- **event_relation:** relação temporal com o evento do ponto ou período no tempo que está a ser definido por este timing spec.
- **start_window:** define um *delay* aceitável dentro do qual a tarefa deverá começar;
- **timeout:** define como deve ser lidado com o facto de nenhum evento ser recebido dentro de um período de tempo.

2.2: Princípios de design

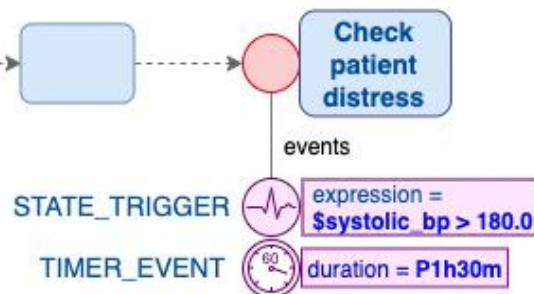
Plan Items

TASK WAIT

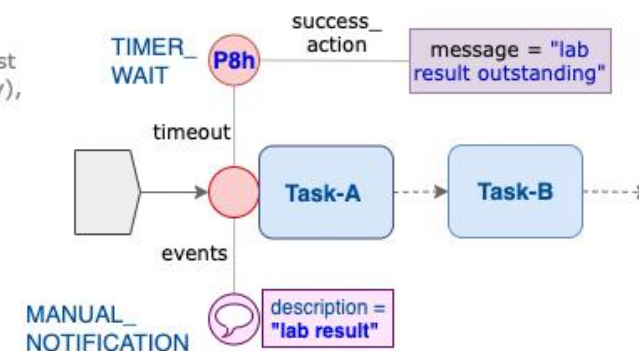
Administer Metformin before lunch on day 2;
administer Metformin before dinner on day 3



Check patient distress
1 hour 30 mins after previous Task is finished
OR
if systolic BP > 160 mm[Hg]



Task-A available when test
result received (manually),
with timeout = 8h.



Perform on day 2 of plan, at 13:30,
with reminder at 15, 20 and 25 mins,
and a message "xx overdue"



2.2: Princípios de design

Plan Items

EVENT_WAIT

success_action (EVENT_ACTION): Ação a ser executada se algum evento escutado retornar um resultado de sucesso.

timeout (TIMER_WAIT): define o que fazer se nenhum outro evento for recebido dentro de um determinado tempo.



TIMER_WAIT

event (TIMER_EVENT):
Evento em qual o estado de espera se baseia

REMINDER

event (PLAN_EVENT):
Evento que vai gerar o reminder

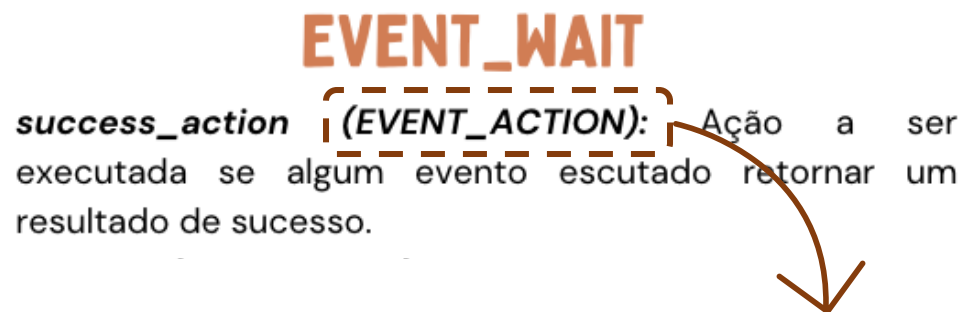
CALLBACK_WAIT

event (CALLBACK_NOTIFICATION):
Evento em qual o estado de espera se baseia

fail_action (EVENT_ACTION): Ação a ser executada se algum evento “escutado” retornar um resultado de falha.

2.2: Princípios de design

Plan Items



Ação(s) a serem executadas ao receber um evento com um estado específico. Quatro coisas possíveis podem ser especificadas:

- uma mensagem;
- uma chamada de sistema;
- o próximo estado da tarefa ou outro local de retomada (*receiver_thread_next_state*);
- uma *resume_action* específica indicando a retomada em um local específico do Plano.

2.2: Princípios de design

Plan Items

CALLBACK

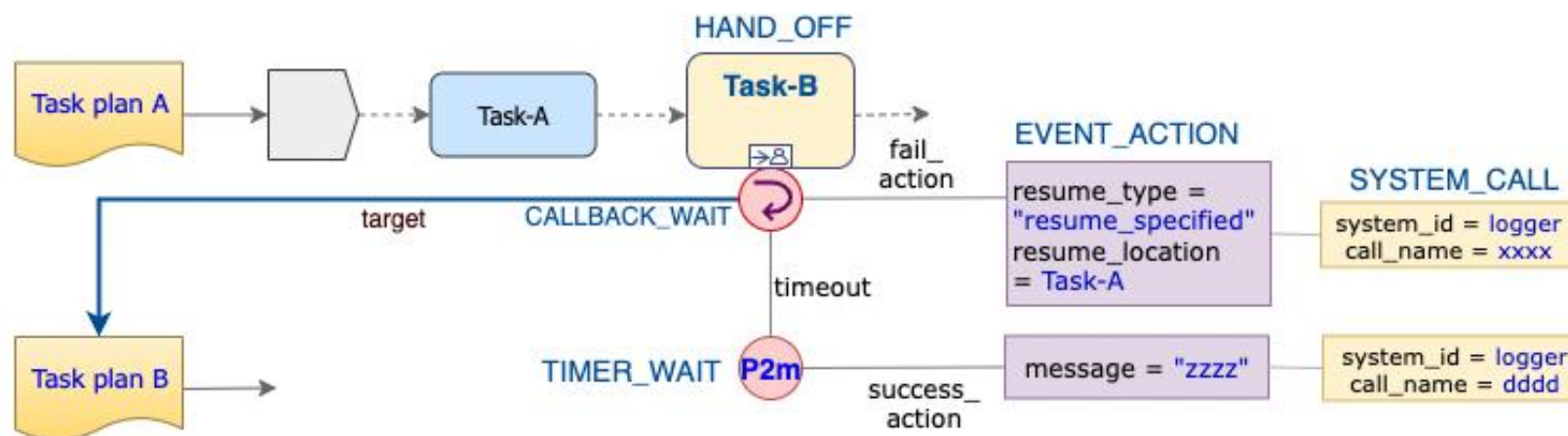
Um *callback* é o mecanismo para definir o que acontece quando o controlo retorna para uma *dispatchable task* do seu destino.

1. a tarefa começa no estado *planned*;
2. quando o ponto de execução atinge a tarefa e qualquer estado de TASK_WAIT tiver sido encerrado, a tarefa fica *available*.
3. a operação de despacho da tarefa é realizada de forma automática, exceto no External Request;
4. se DISPATCHABLE_TASK.wait for falso: a tarefa transita para o estado concluído.
5. se DISPATCHABLE_TASK.wait for verdadeiro: a tarefa entra no estado *underway* até que ocorra callback (ou *timeout*), momento em que o próximo estado dependerá dos detalhes do callback;

2.2: Princípios de design

Plan Items

CALLBACK



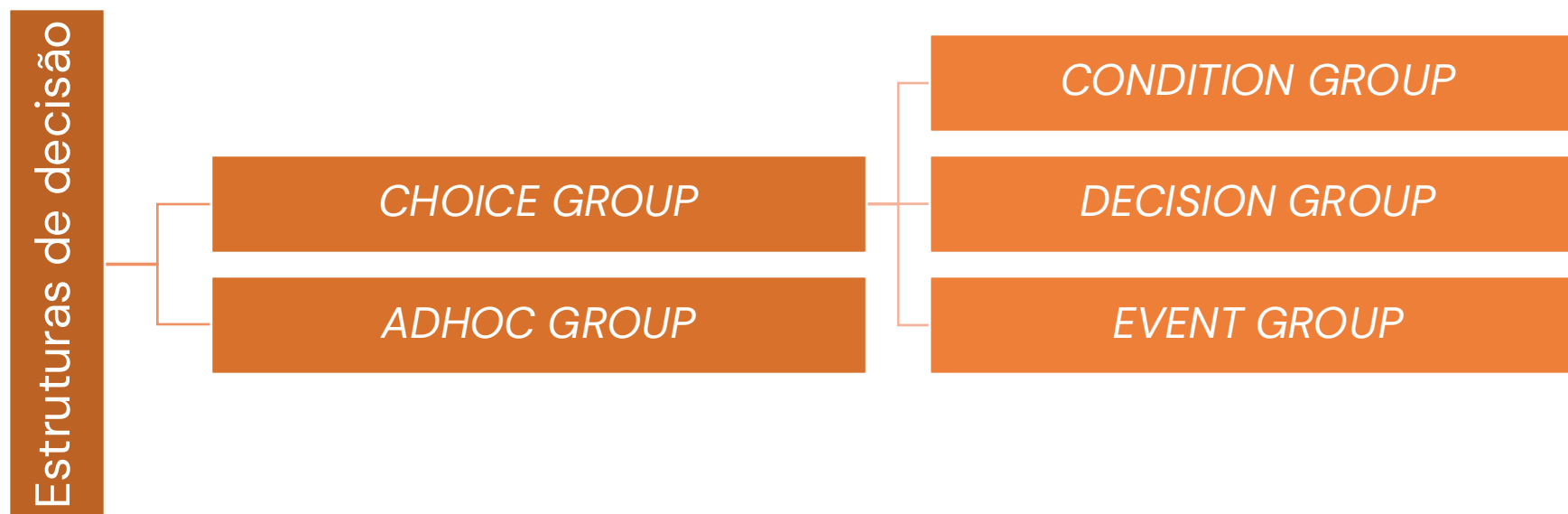
Task-B blocks; hands-off to Plan-B;
callback handling:

- *success*: set lifecycle_state to completed; progress to next Task (no success_action needed)
- *fail*: set lifecycle_state to abandoned; log result & resume at Task-A
- *timeout* (5 mins): log result & display message.

2.2: Princípios de design

Estruturas de decisão

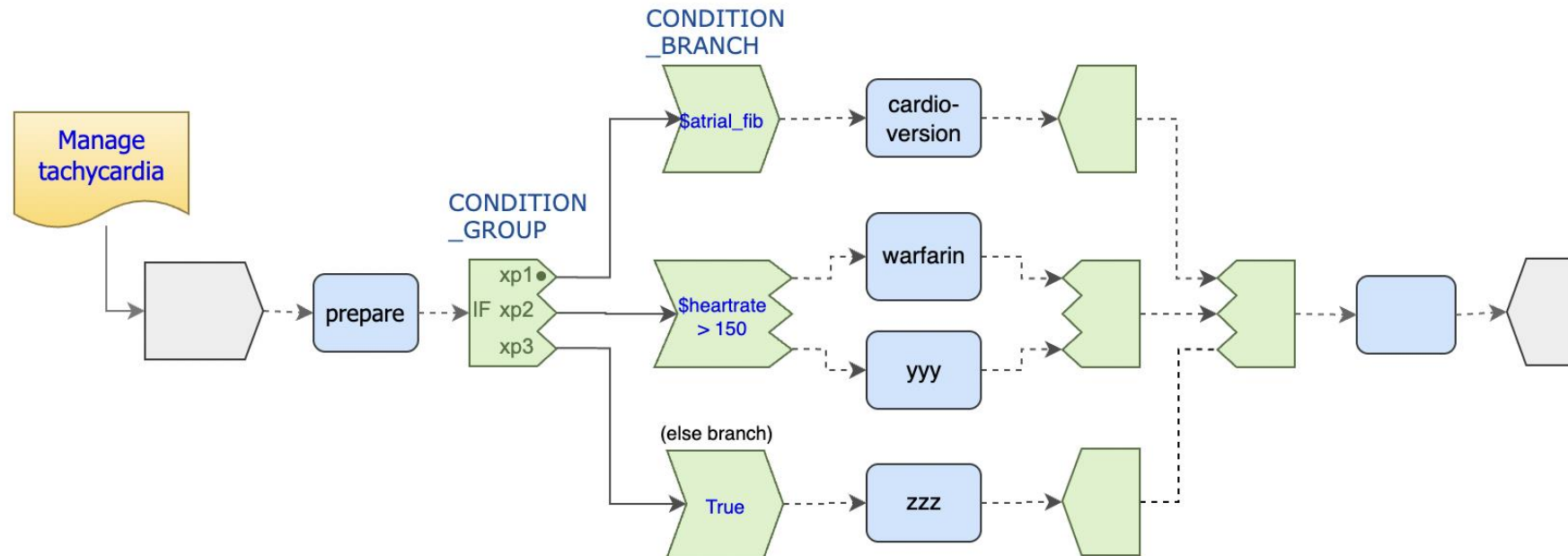
A maioria dos TPs inclui estruturas de decisão, nas quais uma das múltiplas ramificações pode ser seguida dependendo de condições específicas. Existem quatro variedades dessa estrutura, que são definidas como descendentes específicos da classe *TASK_GROUP*.



2.2: Princípios de design

Estruturas de decisão

Condition group (if/elseif/else)



Herdado do
CHOICE_GROUP

CONDITION_GROUP

override_type: OVERRIDE_TYPE (allowed, allowed_with_reason, prohibited)

timeout: TIMER_WAIT

members: List<CONDITION_BRANCH>

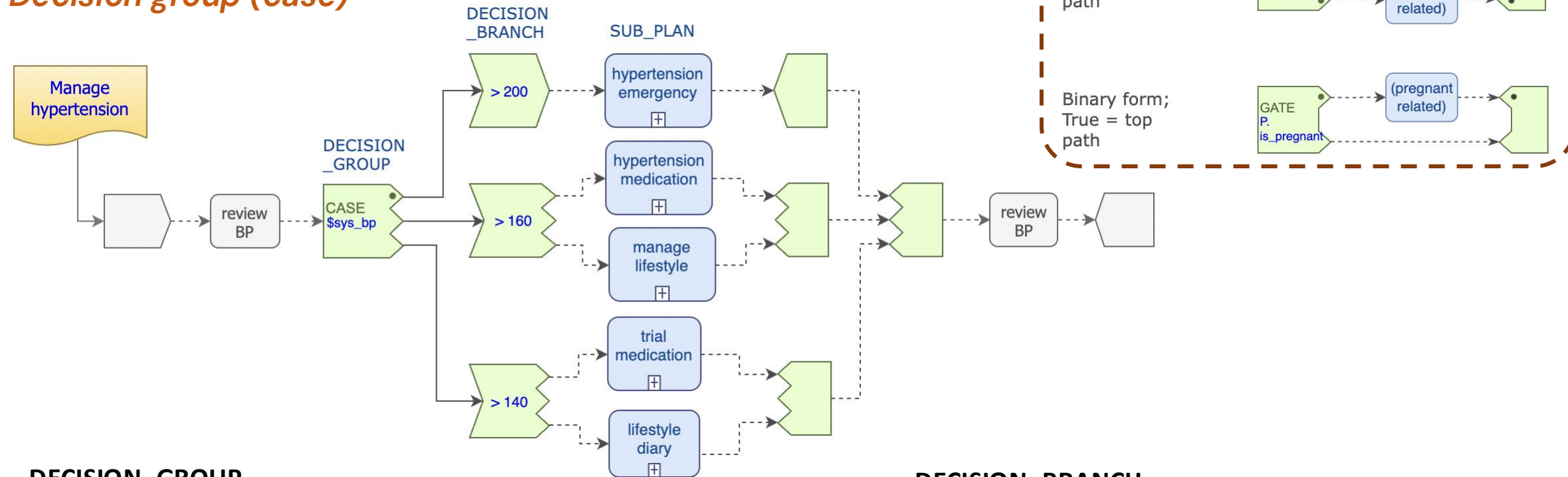
CONDITION_BRANCH

condition: BOOLEAN_CONTEXT_EXPRESSION

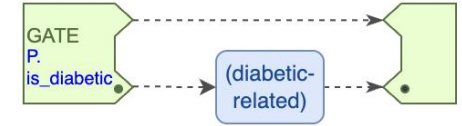
2.2: Princípios de design

Estruturas de decisão

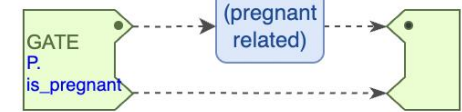
Decision group (case)



Binary form;
True = bottom
path



Binary form;
True = top
path



Herdeado do
CHOICE_GROUP

DECISION_GROUP

override_type: OVERRIDE_TYPE (allowed, allowed_with_reason, prohibited)
timeout: TIMER_WAIT

value: CONTEXT_EXPRESSION

members: List<DECISION_BRANCH>

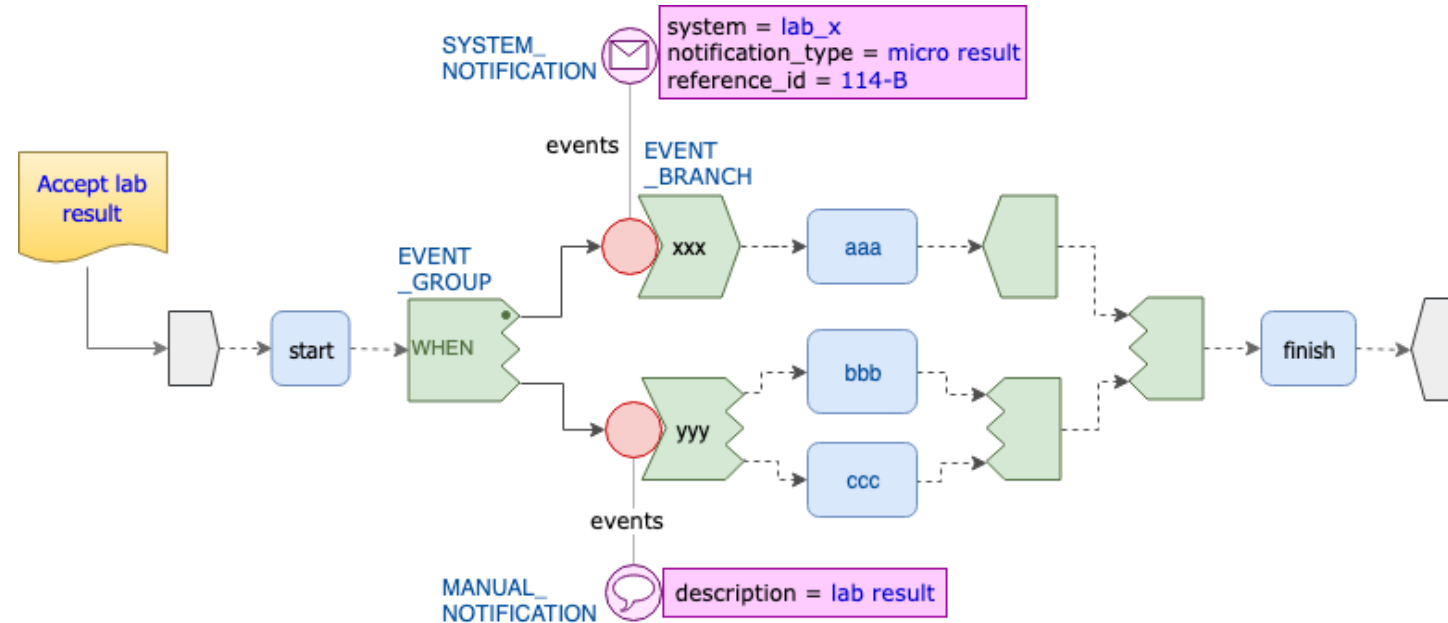
DECISION_BRANCH

value_constraint: BOOLEAN_CONTEXT_EXPRESSION

2.2: Princípios de design

Estruturas de decisão

Event group (when/then)



Herdado do
CHOICE_GROUP

EVENT_GROUP

override_type: OVERRIDE_TYPE (allowed, allowed_with_reason, prohibited)
timeout: TIMER_WAIT

members: List<EVENT_BRANCH>

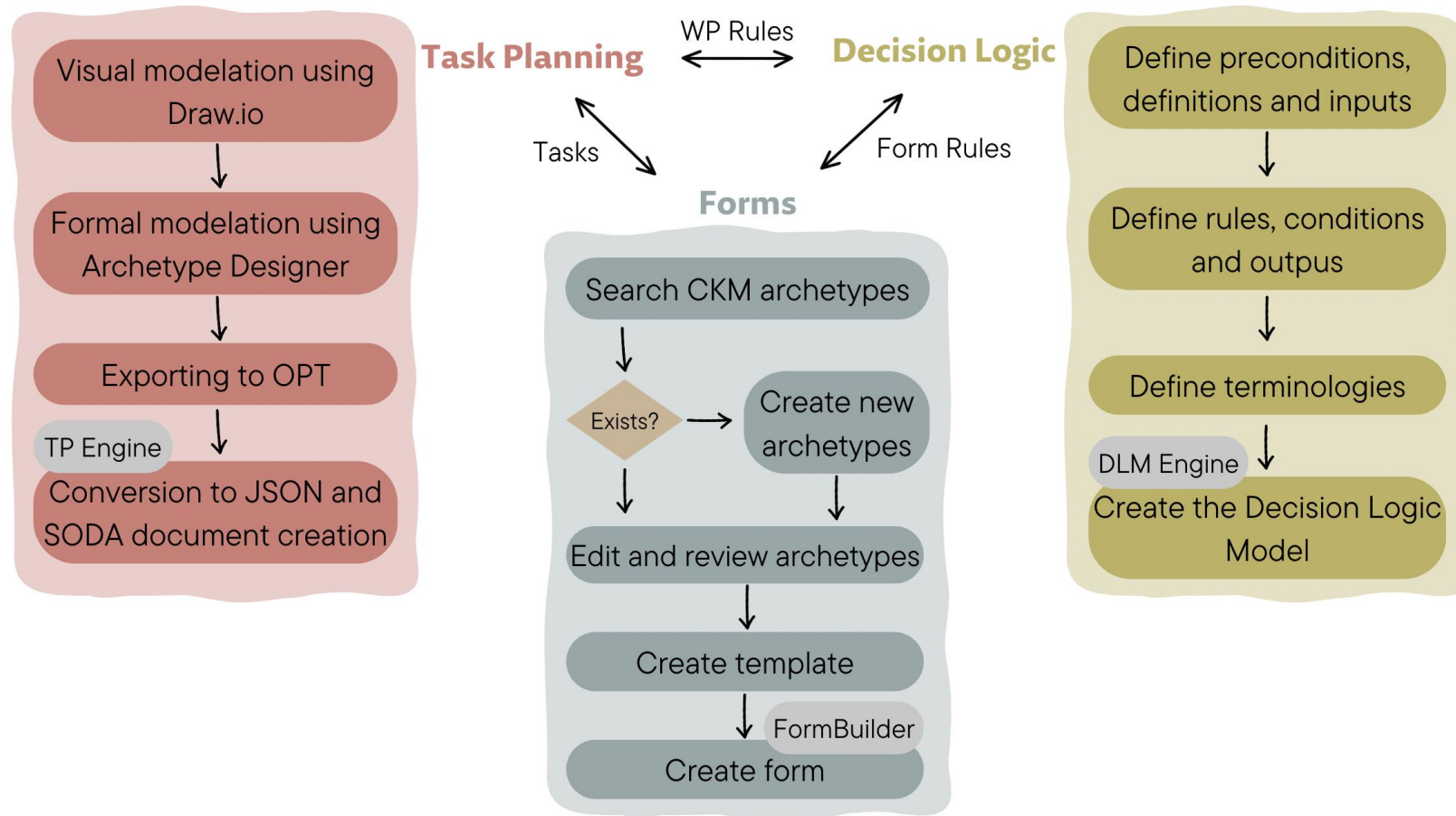
*pelo menos uns dos ramos tem de ter: wait_spec.timeout != Void

EVENT_BRANCH

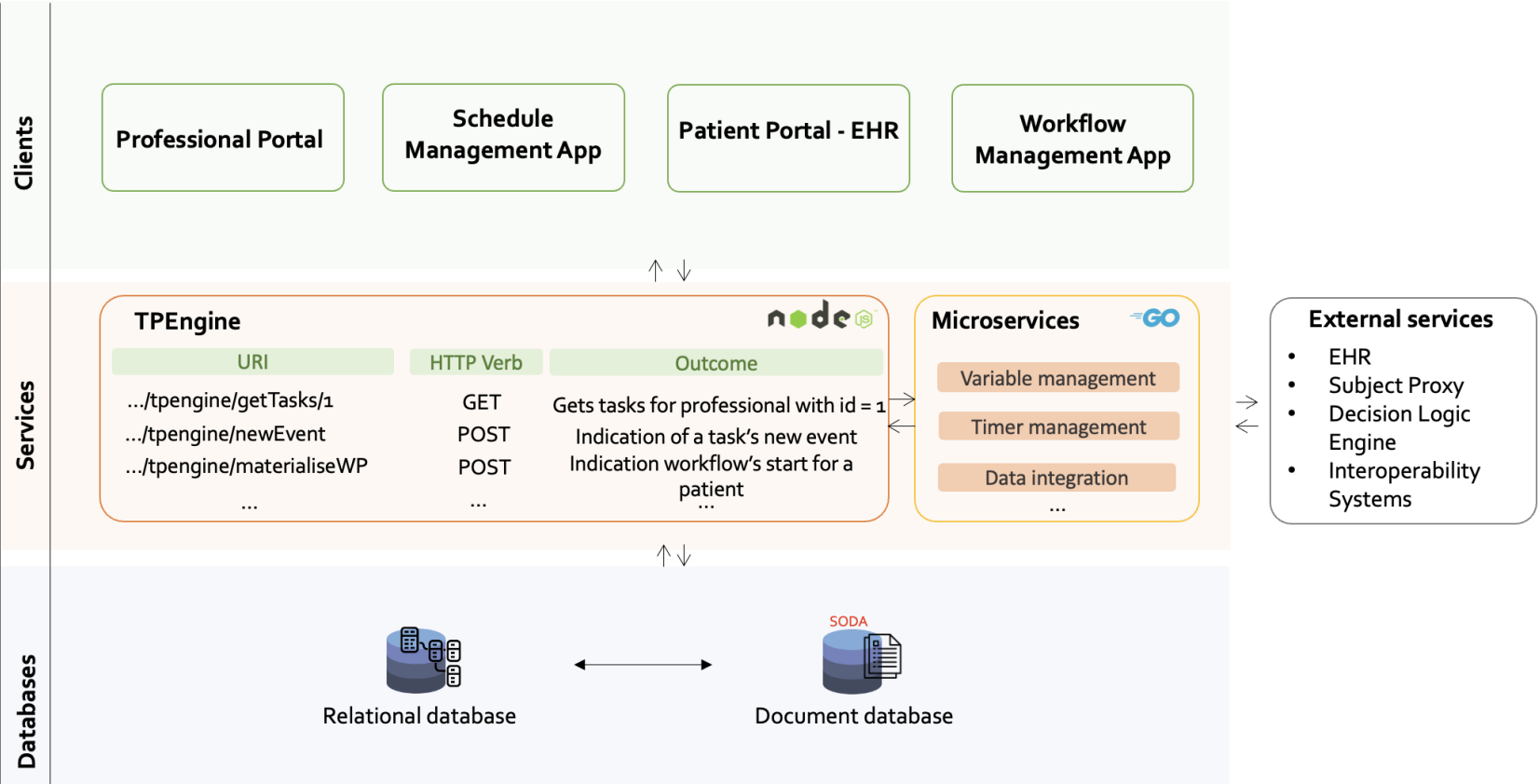
(**wait_spec** herdado do TASK_GROUP tem de estar preenchido:
wait_spec != Void)

2.3: Implementação

Ecosystem



2.3: Implementação Ecossistema



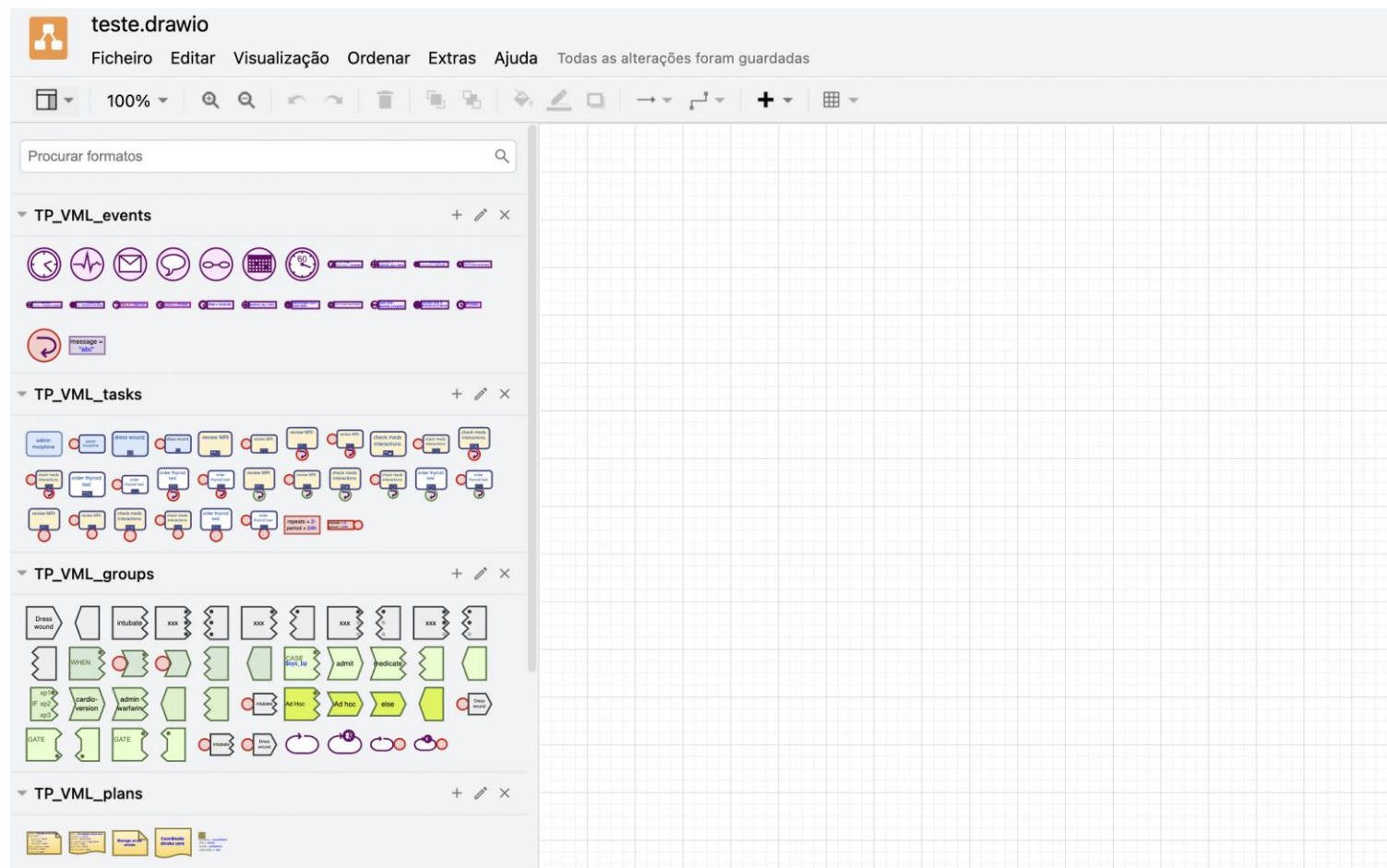
2.4: Ferramentas

[Draw.io](https://draw.io)

**Elementos
OpenEHR para
download**

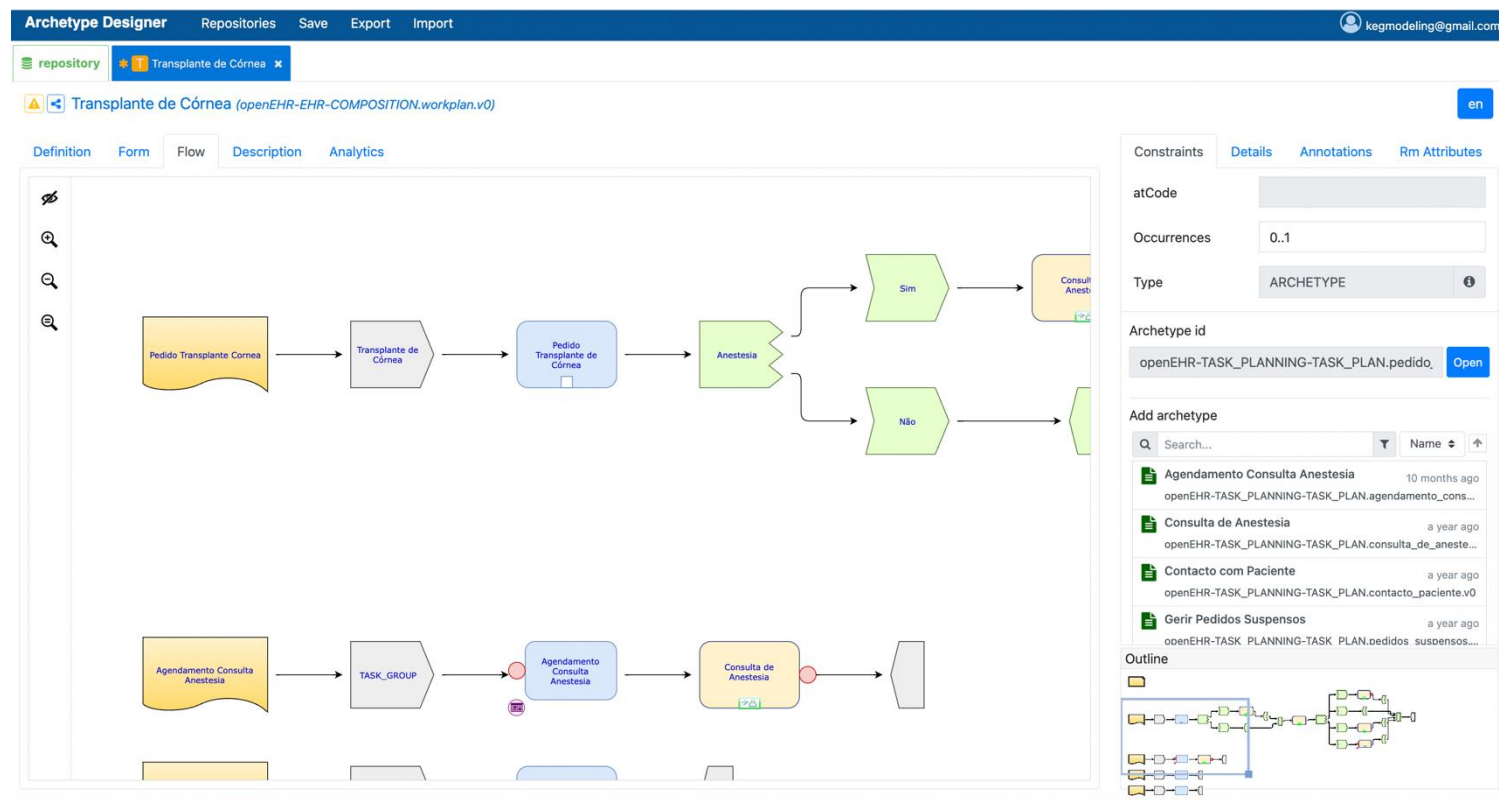
- [TP_VML events](#)
- [TP_VML tasks](#)
- [TP_VML groups](#)
- [TP_VML plans](#)

arrastar
→



2.4: Ferramentas

Archetype Designer



2.4: Ferramentas

Archetype Designer

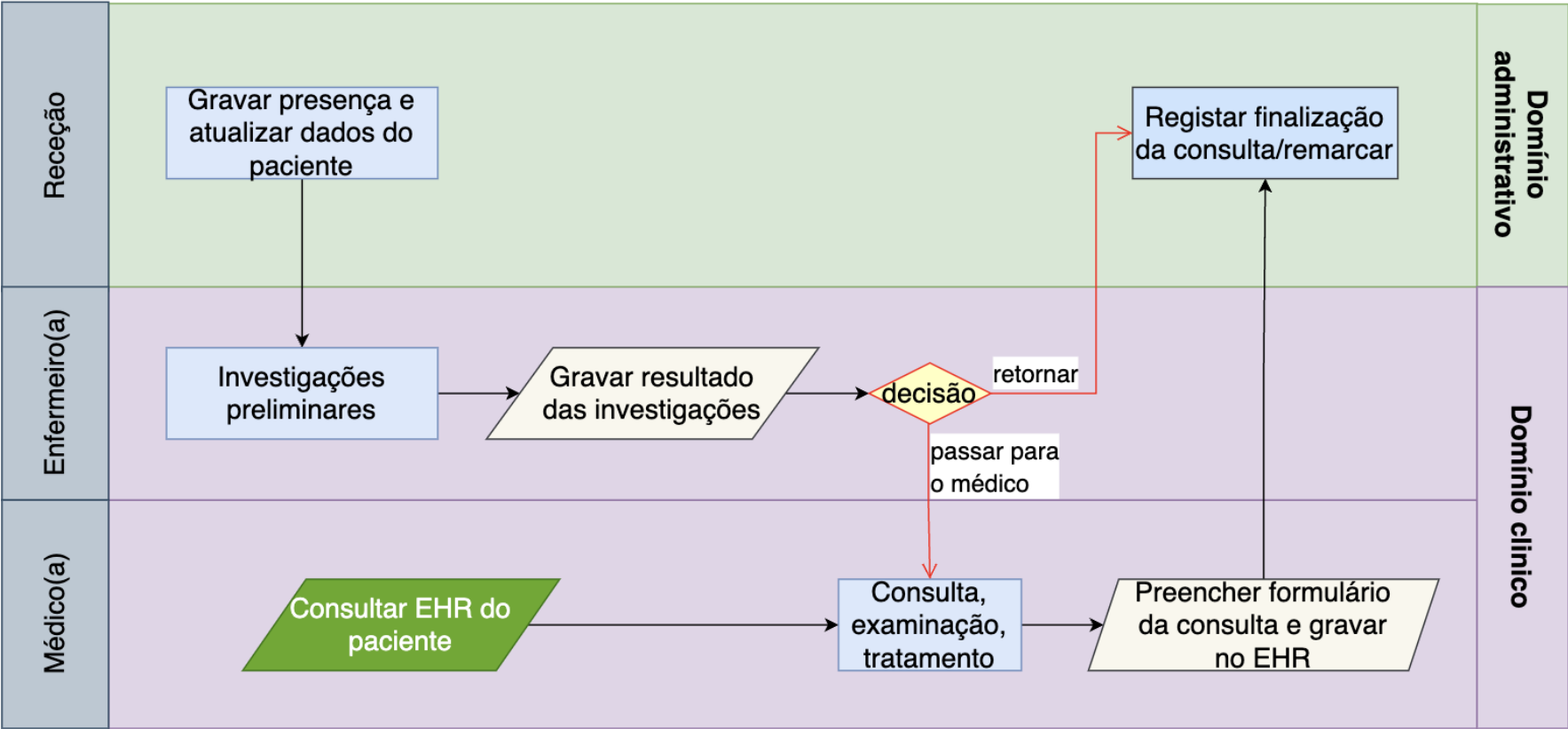
MODELAÇÃO



2.5: Exemplos

Workflow de consultas oftalmológicas

Requisitos



2.5: Exemplos

Workflow de consultas oftalmológicas

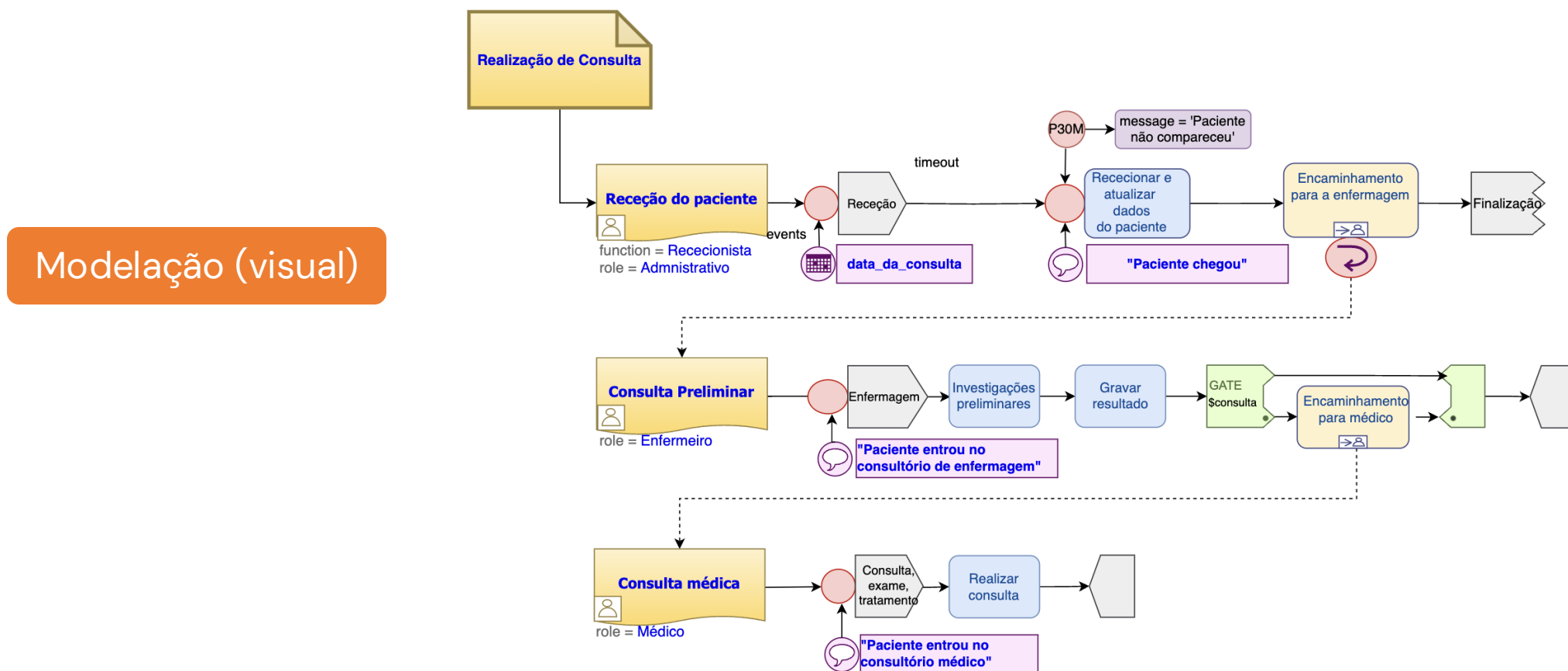
Requisitos

(continuação)

- A receção do paciente só deve ficar disponível quando chegar a data do agendamento da consulta;
- O paciente tem uma tolerância de 30 minutos, após isso a consulta fica sem efeito e deve ser lançada uma mensagem;
- A realização das consultas estão dependentes da chegada do paciente aos respetivos consultórios.
- O WP deve ter como constante a especialidade do caso em questão: oftalmologia.
- As investigações preliminares devem incluir o preenchimento do *form* dos sinais vitais

2.5: Exemplos

Workflow de consultas oftalmológicas



2.5: Exemplos

Workflow de consultas oftalmológicas

Modelação (formal)

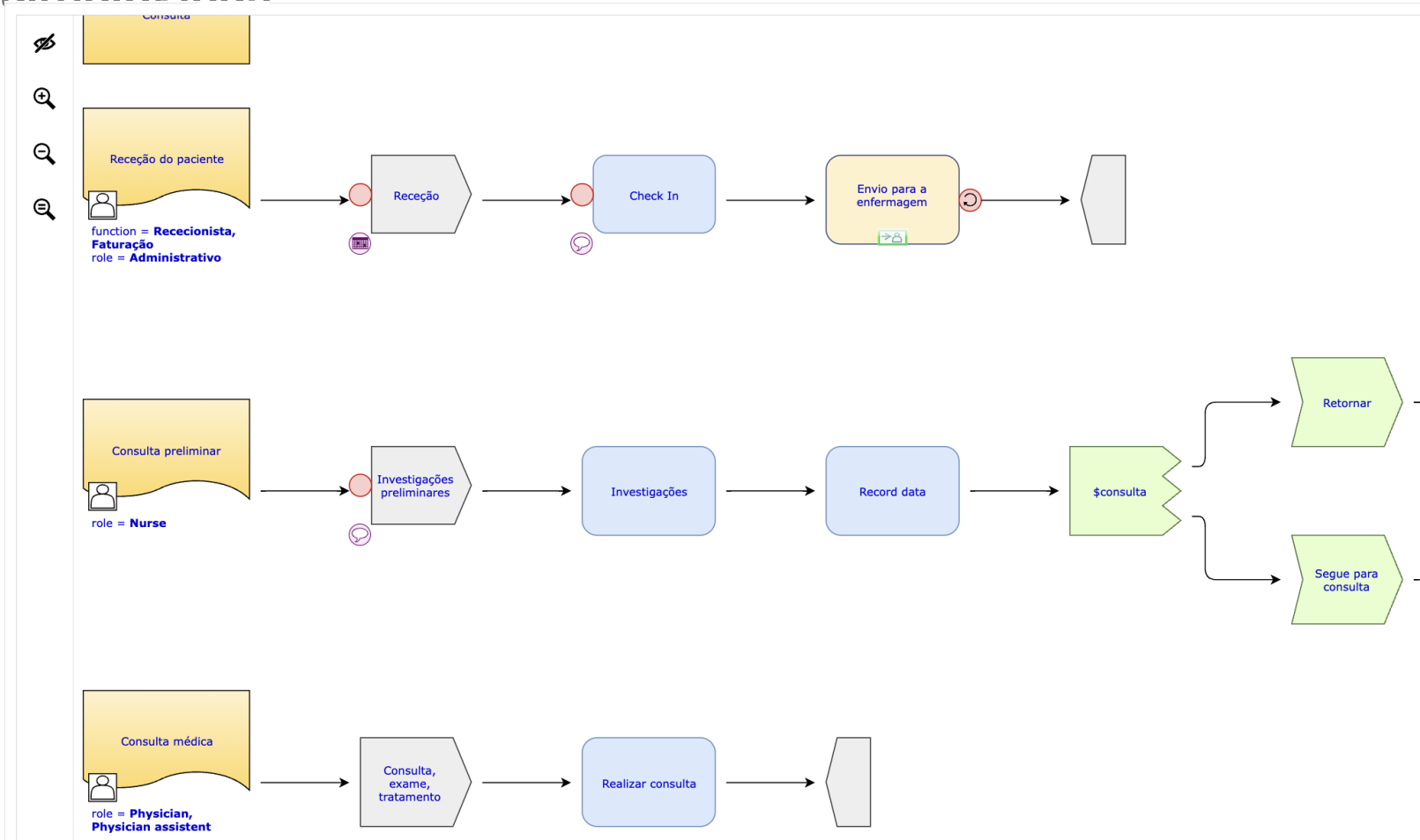
```
[-] Consulta NAME (from: 'generic_work_plan')
  [-] → top_level_plans
    [-] openSlot
      [-] [ ] Recepção do paciente Δ [0..*] to [0..1]
        [-] → definition
          [-] [ ] Recepção
            [-] → members
              [+][ ] Rececionar e atualizar dados do paciente NAME (from: 'Check In')
            [-] [ ] Envio para a enfermagem
              [-] → action
                [-] ▷ HAND_OFF
                  [-] → target
                    [-] [ ] TASK_PLAN
                      [-] [ ] Consulta preliminar Δ [0..*] to [0..1]
```

(...)

2.5: Exemplos

Workflow de consultas oftalmológicas

Modelação (formal)



Links Úteis

- [CDS, Guidelines and Planning Overview](#)
- [Task Planning \(TP\) Specification](#)
- [openEHR CDS, Guidelines and Planning Examples](#)
- [Task Planning Visual Modelling Language \(TP-VML\)](#)