

Modelos Transacionais e a Evolução dos Sistemas Computacionais

Concurso Professor Titular IME-USP
Prova de Erudição
Edital ATAc 018/2018

Candidato: João Eduardo Ferreira

Disciplinas:

MAC5760 - Introdução aos Sistemas de Bancos de Dados

MAC5861 - Modelagem de Banco de Dados

MAC5865 - Tópicos em Ciência e Engenharia de Dados

20 de maio de 2019



- 1 Introdução e desafios
- 2 Os limites dos modelos transacionais
- 3 WED-flow e processos complexos
- 4 Contextualização da contribuição
- 5 Próximos passos
- 6 Conclusão

- 1 Introdução e desafios
- 2 Os limites dos modelos transacionais
- 3 WED-flow e processos complexos
- 4 Contextualização da contribuição
- 5 Próximos passos
- 6 Conclusão

De 33 Zettabytes em 2018 para 175 Zettabytes em 2025¹

“O pior desafio não é armazenar, mas como processar todo esse material para extrair informação relevante [como registro histórico].”

Prof. Dr. Marco Antonio Casanova

¹Reinsel, D.; Gantz, J.; Rydning, J. The Digitalization of the World From Edge to Core. IDC, white paper, November 2018.

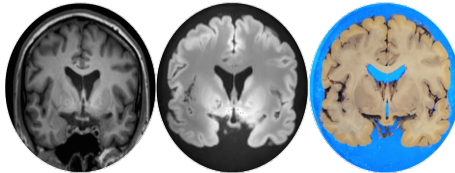
Gerenciamento de pesquisas baseadas em autópsias: SISAUT¹



Coleta de materiais



Entrevista



Processamento de materiais

¹[Ferreira et al., 2017, IEEE-CIC]



Sala de Entrevistas

Diagrama de um layout de loja. Uma seção é rotulada "Recepção". Há ícones de pessoas e uma seta indicando o fluxo de entrada.

Sala de Autópsia

Diagrama de um laboratório com uma lupa e um símbolo de biohazard.

Laboratório



Process. de Materiais

```

graph LR
    A[Analisa Material] --> B[Executa Procedimentos]
  
```



Fluxo de execução (Dependência Intraprocesso)

Dependência Interprocessos

Fluxo de execução

(Dependência
Intraproces

Dependência Interprocessos

Prover um modelo e uma plataforma de software para desenvolvimento de novas aplicações que envolvem processos complexos com sincronizações internas e externas e que requerem:

- ① tratamento de **transações com escalas de tempo distintas**;
- ② preservação de **dependências entre os componentes dos processos** para garantir corretude;
- ③ aproveitamento de **processamento paralelo** para atingir alta eficiência.

- 1 Introdução e desafios
- 2 Os limites dos modelos transacionais**
- 3 WED-flow e processos complexos
- 4 Contextualização da contribuição
- 5 Próximos passos
- 6 Conclusão

Definição (1 - **Transação**)

Uma transação $\mathcal{T}_i$ é uma ordem parcial estrita com ordenação $<_i$ tal que:

- $\mathcal{T}_i \subseteq \{r_i[x], w_i[x] \mid x \text{ é um dado}\} \cup \{a_i, c_i\}$
Ou seja, $\mathcal{T}_i$ é um conjunto finito composto apenas de operações de leitura, operações de escrita e operações de terminação, a_i ou c_i ;
- $\mathcal{T}_i$ possui uma e apenas uma operação de terminação:
 $c_i \in \mathcal{T}_i \iff a_i \notin \mathcal{T}_i$;
- a operação de terminação t_i é a última operação da transação:
 $p_i <_i t_i$ para toda operação p_i , $t_i = a_i$ ou $t_i = c_i$;
- se $r_i[x], w_i[x] \in \mathcal{T}_i$, então $r_i[x] <_i w_i[x]$ ou $w_i[x] <_i r_i[x]$ (operações sobre o mesmo dado devem ser ordenadas).

¹[Gray et al., 1975, VLDB] [Berstein et al, 1987] [Ferreira and Finger, 2000]

Definição (2 - **Escalação seriável**)

Um escalonamento é dito seriável se a projeção confirmada de $\mathcal{E}$ é equivalente a um escalonamento serial. Ou seja, se existe um escalonamento serial $\mathcal{E}_s$ tal que $C(\mathcal{E}) \equiv \mathcal{E}_s$.

¹[Gray et al., 1975, VLDB] [Berstein et al, 1987] [Ferreira and Finger, 2000]

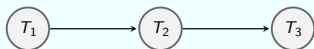
O grafo $GS(\mathcal{E})$ é um grafo dirigido construído da seguinte maneira:

- os nós de $GS(\mathcal{E})$ são as transações confirmadas de $\mathcal{E}$;
- para cada par de operações conflitantes p_i e q_j , tal que $p_i <_{\mathcal{E}} q_j$, adicionamos a aresta $T_i \rightarrow T_j$ a $GS(\mathcal{E})$.

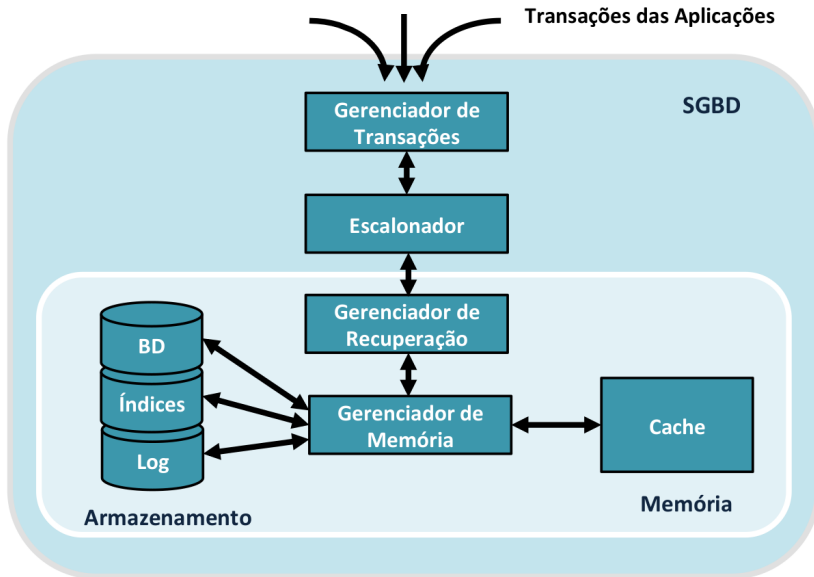
Teorema (**Condição de Seriabilidade**)

Um escalonamento $\mathcal{E}$ é seriável se e somente se $GS(\mathcal{E})$ é acíclico.

Exemplos



¹[Gray et al., 1975, VLDB] [Berstein et al, 1987] [Ferreira and Finger, 2000]



Transações clássicas

- Excelente modelo para otimização da execução de transações atômicas, recuperação de falhas com um formalismo robusto.
- **Aplicação limitada quando usada** em aplicações avançadas de banco de dados (**Transações Longas**)

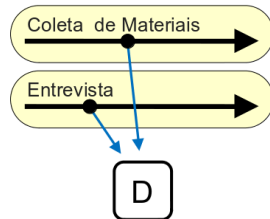
- 
- Longos períodos para realizar *commits*
 - Múltiplos passos sobre bancos de dados heterogêneos
 - Exposição de resultados intermediários

Isolamento

Muito estrito para **prover compartilhamento de dados** entre transações.

Objetivo desejado

Relaxar ou modificar as propriedades transacionais de modo a obter um **modelo para transações longas**.



Beyond Isolation: Research Opportunities in Declarative Data-Driven Coordination

Lucja Kot, Nitin Gupta, Sudip Roy, Johannes Gehrke, and Christoph Koch

Cornell University
Ithaca, NY 14853, USA

{lucja, niting, sudip}@cs.cornell.edu, {johannes, christoph}@cs.cornell.edu

ABSTRACT

There are many database applications that require users to coordinate and communicate. Friends want to coordinate travel plans, students want to jointly enroll in the schedules of courses, and busy professionals want to coordinate their schedules. These tasks are difficult to program using existing abstractions provided by database systems because, in addition to the traditional ACID properties provided by a system they all require some type of *coordination* between users. This is fundamentally incompatible with isolation in the classical ACID properties.

In this position paper, we argue that it is time for the database community to look beyond isolation towards principled and elegant abstractions that allow for communication and coordination between some notion of (suitably generalized) transactions. This new area of *declarative data-driven*

system they all require some type of *coordination* between users. This is fundamentally incompatible with isolation in the classical ACID properties.

In this position paper, we argue that it is time for the database community to look beyond isolation towards principled and elegant abstractions that allow for communication and coordination between some notion of (suitably generalized) transactions. This new area of *declarative data-driven*

collaboration. How does this coordination happen today? Typically, it starts with a lengthy phone or email conversation to decide on the itinerary. Then, one person books tickets for everyone, and there is another round of emails to sort out the finances. Or everybody arranges to book at about the same time, hoping that the airline seats do not fill up in the meanwhile.

Coordination is not just inefficient, but also not just inefficient.

1

Estendendo a Seriabilidade
ajuste Menor

- ➡ **Não expõe** resultados intermediários a outras transações
- ➡ **Seriabilidade ainda é o critério de correção**

2

Relaxando a Seriabilidade
permite confirmações (*commit*)
parciais

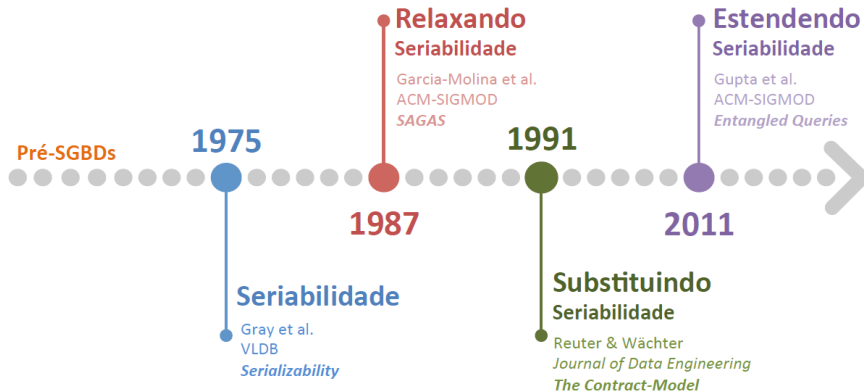
- ➡ **Expõe** resultados intermediários a outras transações
- ➡ **A Seriabilidade é relaxada** considerando-se a compensação das transações

3

Substituindo a Seriabilidade

- ➡ **Expõe** resultados intermediários a outras transações
- ➡ **Seriabilidade é ignorada e um critério alternativo de correção deve ser definido**

¹[Schwerz, 2016] [Ferreira et al., 2019]



¹[Ferreira et al., 2019]

1

Estendendo a Seriabilidade ajuste Menor

- ➡ Não expõe resultados intermediários a outras transações
- ➡ Seriabilidade ainda é o critério de correção

Relaxando a Seriabilidade permite confirmações (*commit*) parciais

2

- ➡ **Expõe** resultados intermediários a outras transações
- ➡ A **Seriabilidade é relaxada** considerando-se a compensação das transações

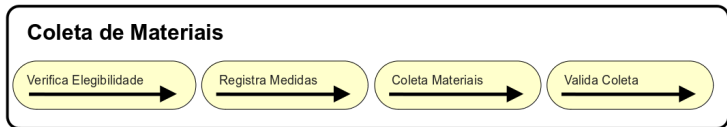
3

Substituindo a Seriabilidade

- ➡ Expõe resultados intermediários a outras transações
- ➡ Seriabilidade é ignorada e **um critério alternativo de correção deve ser definido**

¹[Schwerz, 2016] [Ferreira et al., 2019]

- Transação é uma sequência de subtransações (passos SAGAs)
 - ▶ viola o isolamento do ponto de vista global da transação



- Garante a atomicidade por passos compensáveis: $T_i \rightarrow T_i^{-1}$
- Adequado para transações longas
- Principal objetivo
 - ▶ permite *commit* parciais
- Desvantagem
 - ▶ não é possível garantir **consistência** nas dependências de dados entre **transações longas**.

¹[Garcia-Molina et al., 1987, ACM-SIGMOD]

Linguagens WF-net são poderosas na especificação de dependência e terminação. **Entretanto**, existem dificuldades em expressar interações entre processos e passos de recuperação em larga escala usando WF-net sem estados de dados explícitos, eventos e transições.

1

Estendendo a Seriabilidade
ajuste Menor

- ➔ **Não expõe** resultados intermediários a outras transações
- ➔ **Seriabilidade ainda é o critério de correção**

2

Relaxando a Seriabilidade
permite confirmações (*commit*)
parciais

- ➔ **Expõe** resultados intermediários a outras transações
- ➔ A **Seriabilidade é relaxada** considerando-se a compensação das transações

3

Substituindo a Seriabilidade

- ➔ **Expõe** resultados intermediários a outras transações
- ➔ Seriabilidade é ignorada e **um critério alternativo de correção** deve ser definido

- Transações clássicas que comunicam entre si por meio de **consultas entrelaçadas**.
- Consultas entrelaçadas permitem interação entre consultas no banco de dados.

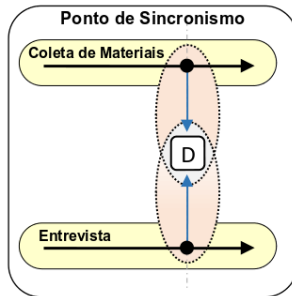
```
SELECT 'Kramer', fno INTO ANSWER Reservation
WHERE
fno IN (SELECT fno FROM Flights WHERE dest='Paris')
AND ('Jerry', fno) IN ANSWER Reservation
CHOOSE 1
```

```
SELECT 'Jerry', fno INTO ANSWER Reservation
WHERE
fno IN (SELECT fno FROM Flights F, Airlines A WHERE
      F.dest='Paris' and F.fno = A.fno
      AND A.airline = 'United' )
AND ('Kramer', fno) IN ANSWER Reservation
CHOOSE 1
```

- ▶ “Reservation” é um nome para a relação virtual que contém as respostas para as consultas ao banco de dados.

¹[Gupta et al., 2011a, ACM-SIGMOD] [Gupta et al., 2011b, VLDB]

- Organiza a execução de transações em lotes discretos.
- Cada lote é uma execução de um conjunto de transações entrelaçadas.
- Possíveis estados na execução:
 - ▶ Bloqueado
 - ▶ Abortado
 - ▶ Confirmação entrelaçada (*commit*)



Transações entrelaçadas proveem mecanismos para que transações clássicas comuniquem entre si por meio de [consultas entrelaçadas](#).
Entretanto, temos dificuldades para tratar **transações longas**.

1

Estendendo a Seriabilidade
ajuste Menor

- ➡ **Não expõe** resultados intermediários a outras transações
- ➡ **Seriabilidade ainda é o critério de correção**

2

Relaxando a Seriabilidade
permite confirmações (*commit*)
parciais

- ➡ **Expõe** resultados intermediários a outras transações
- ➡ **A Seriabilidade é relaxada** considerando-se a compensação das transações

3

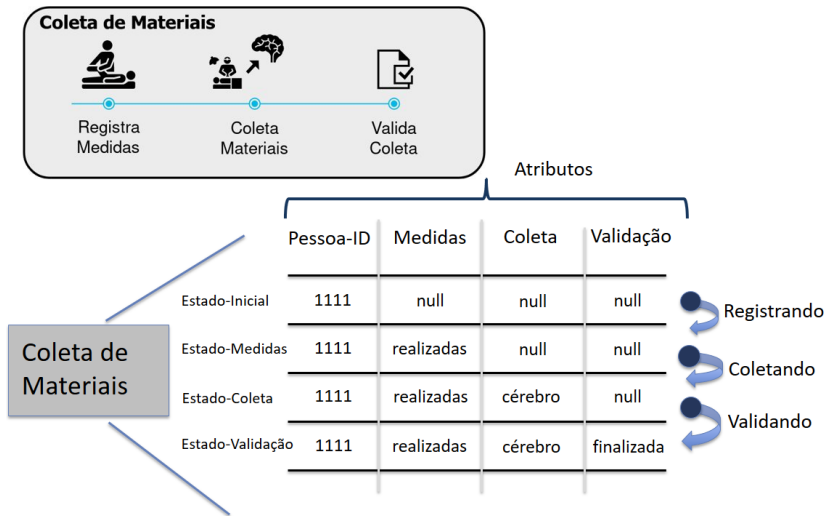
Substituindo a Seriabilidade

- ➡ **Expõe** resultados intermediários a outras transações
- ➡ **Seriabilidade é ignorada e um critério alternativo de correção deve ser definido**

É possível usufruir, ao mesmo tempo, dos benefícios (comunicação e commits parciais) proporcionados pelos modelos transacionais que estendem e relaxam a seriabilidade?

- 1 Introdução e desafios
- 2 Os limites dos modelos transacionais
- 3 WED-flow e processos complexos**
- 4 Contextualização da contribuição
- 5 Próximos passos
- 6 Conclusão

- (**W**ork, **E**vent-based, **D**ata)-flow: **WED-flow**
- é um modelo de transações longas que inclui regras de Evento, Condição e Ação (ECA) para apoiar o desenvolvimento de sistemas computacionais que envolvem processos complexos;
- possui uma plataforma de software que implementa o referido modelo.



¹[Ferreira et al., 2010, OTM-CoopIS] [Ferreira et al., 2012, IEEE-ICWS]

Definição (3 - **WED-attributes**)

Denotamos por WED-attributes de uma aplicação a tupla

$$\mathcal{A} = \langle a_1, a_2, \dots, a_n \rangle,$$

em que cada a_i (com $1 \leq i \leq n$) é um atributo de interesse para a aplicação em questão.

Exemplo (WED-attributes)

$$\mathcal{A} = \langle \text{'idade'}, \text{'sexo'}, \text{'Elegibilidade_confirmada'} \rangle$$

Definição (4 - **WED-state**)

Conjunto de valores de atributos que definem um estado da execução de uma aplicação concreta.

$$s_i \in \mathcal{S} \mid s_i = (a_1 = 'v_1', a_2 = 'v_2', \dots, a_n = 'v_n')$$

Exemplo (WED-state)

$$s_1 = ('idade' = 18, 'sexo' = 'M', 'Elegibilidade_confirmada' = \text{NULL})$$

Definição (5 - **WED-condition**)

Predicado lógico definido sobre um conjunto específico de WED-states. Captura eventos (mudanças de estados de dados) e habilita o disparo de WED-transitions como parte da execução do *workflow*.

$$c_j \in \mathcal{C} \mid c_j \text{ is true}$$

Exemplo (WED-condition)

Captura o evento “Família Aceita Entrevista”:

$c_1 =$ ‘idade’ is not NULL AND ‘sexo’ is not NULL AND
‘Elegibilidade_confirmada’ is NULL



Definição (6 - **WED-transition**)

Passo da execução da aplicação que transforma uma entrada (WED-state) em uma saída (WED-state); sua execução é encapsulada em uma transação atômica como um passo SAGA.

$$t_k \in \mathcal{T} \mid t_k(s_i) \rightarrow s_j$$

Exemplo (WED-transition)

t_1 = Verifica Elegibilidade

Essa transação classifica, com base nos dados do óbito, se o caso é elegível para projetos de pesquisa. Ele define 'Elegibilidade_confirmada' como 'Verdadeiro' ou 'Falso'.



Definição (7 - **WED-trigger**)

Vínculo entre WED-condition e WED-transition; quando a WED-condition de uma WED-trigger é satisfeita, sua WED-transition associada é executada.

$$g_l \in \mathcal{G} \mid g_l = (c_j, t_k) \text{ em que } c_j \in \mathcal{C} \text{ e } t_k \in \mathcal{T}$$

Exemplo (WED-trigger)

$$(c_1, t_1),$$

em que:

c_1 = 'idade' is not NULL AND 'sexo' is not NULL AND
'Elegibilidade_confirmada' is NULL,
 t_1 = Verifica Elegibilidade.



Definição (8 - **WED-flow**)

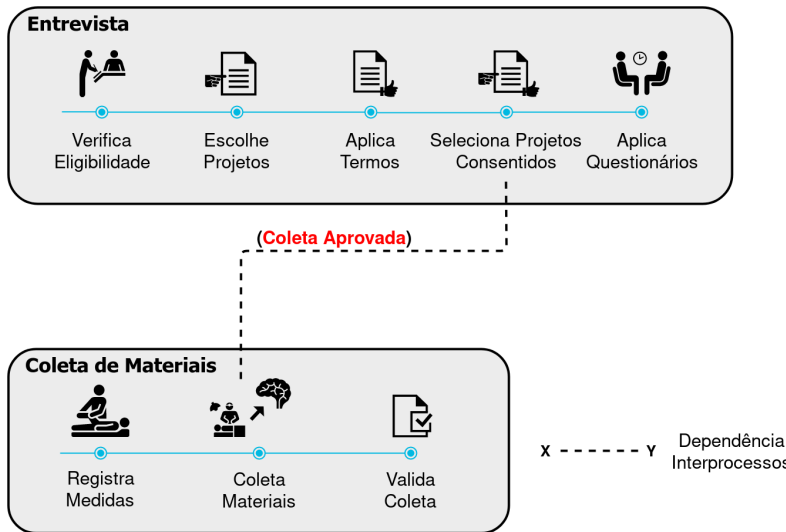
Um WED-flow é uma 3-tupla:

$$\langle \mathcal{G}', c_i, c_f \rangle,$$

em que $\mathcal{G}' \subseteq \mathcal{G}$ e $c_i, c_f \in \mathcal{C}$.

A condição c_i define o conjunto de WED-states iniciais válidos, enquanto c_f define o conjunto de WED-states finais válidos para o fluxo.

Um WED-flow pode ser visto como uma composição de WED-transitions que são disparadas por WED-states que satisfazem suas WED-conditions associadas. Cada WED-transition disparada produz um WED-state como saída, o qual pode disparar a execução de outras WED-transitions do WED-flow.

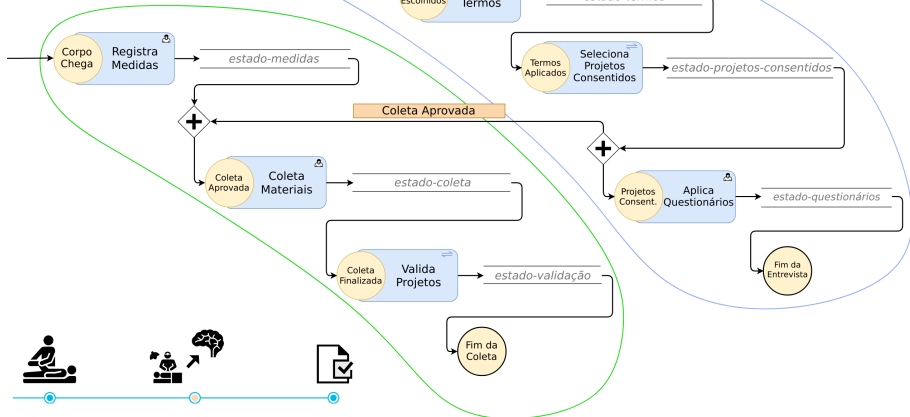


¹[Ferreira et al., 2017, IEEE-CIC]

Entrevista



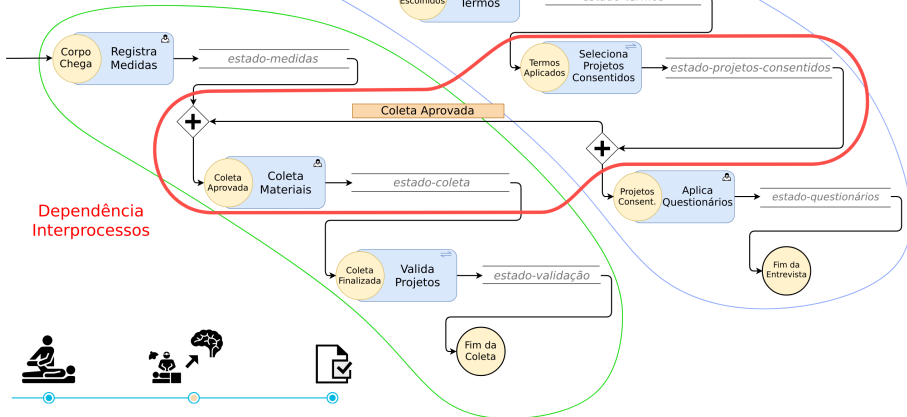
Coleta de Materiais



Entrevista

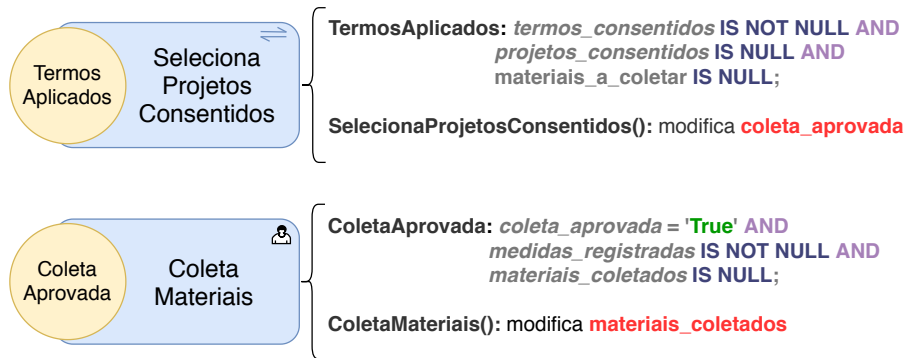


Coleta de Materiais



Dependência Interprocessos

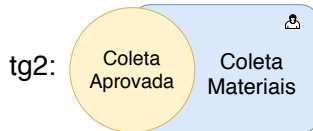
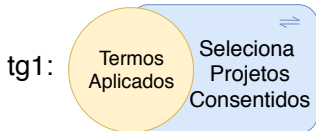
Especificação WED-flow da dependência interprocesso:



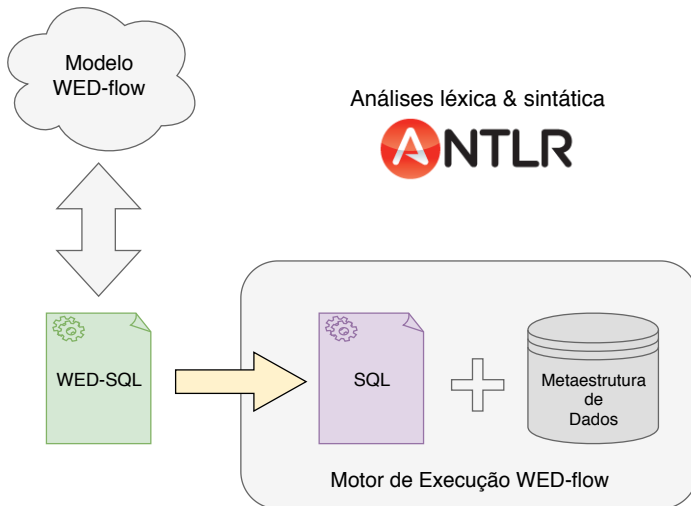
¹[Ferreira et al., 2017, IEEE-CIC] [Schwerz et al., 2018, IEEE-Journal-TSC]

Exemplo (WED-SQL)

```
create wed-attribute projetos_consentidos, termos_consentidos,  
materiais_a_coletar, medidas_registradas, coleta_aprovada, materiais_coletados;  
create wed-condition termos_aplicados as (termos_consentidos is not null ) and  
projetos_consentidos is null and materiais_a_coletar is null;  
create wed-condition coleta_aprovada as coleta_aprovada and  
medidas_registradas is not null and materiais_coletados is null;  
create wed-transition seleciona_projetos_consentidos;  
create wed-transition coleta_materiais;  
create wed-trigger tg1 as (termos_aplicados, seleciona_projetos_consentidos);  
create wed-trigger tg2 as (coleta_aprovada, coleta_materiais);
```



¹[Padilha et al., 2018, IEEE-ICWS]



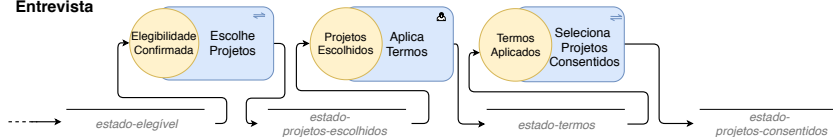
¹[Padilha et al., 2018, IEEE-ICWS; <http://www.data.ime.usp.br>]

Definição (9 - **Caminho de Recuperação**)

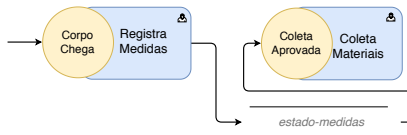
- “backward”: caminho de recuperação composto por passos (SAGAS) de compensação que reverterem a execução do WED-flow a um estado equivalente ao anterior;
- “forward”: caminho de recuperação composto por passos que geram o próximo estado de dados do WED-flow.

¹[Ferreira et al., 2012, IEEE-ICWS] [Schwerz et al., 2018, IEEE-Journal-TSC]

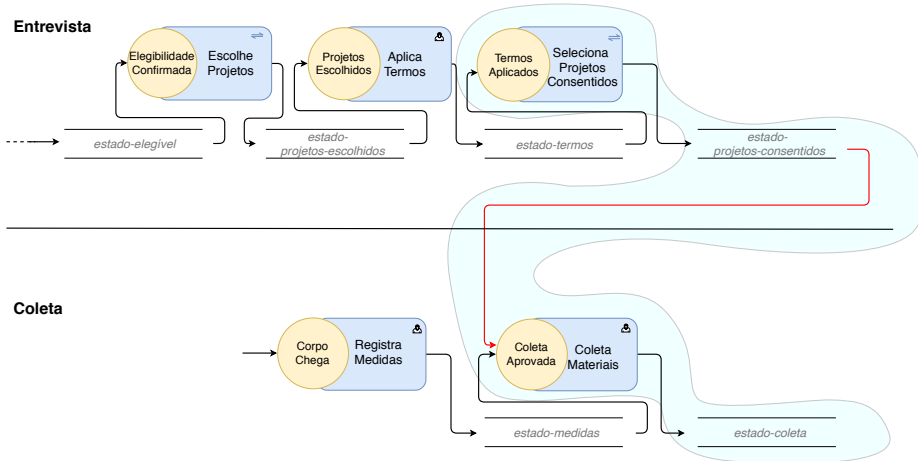
Entrevista



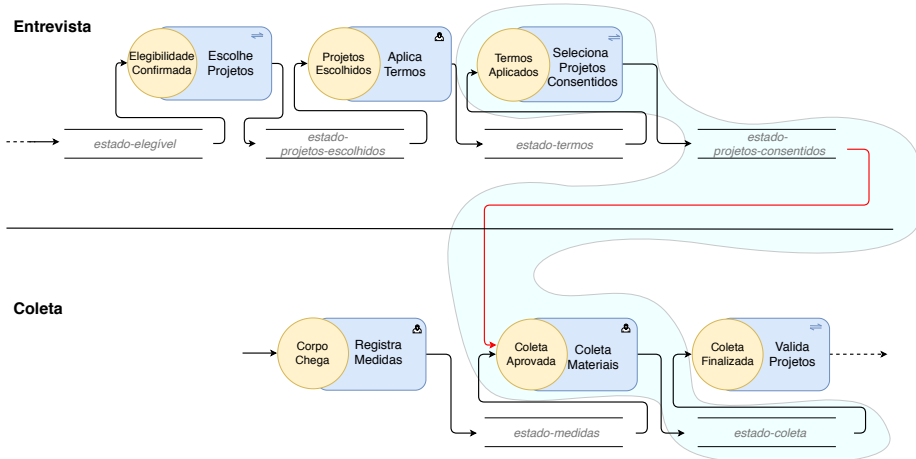
Coleta



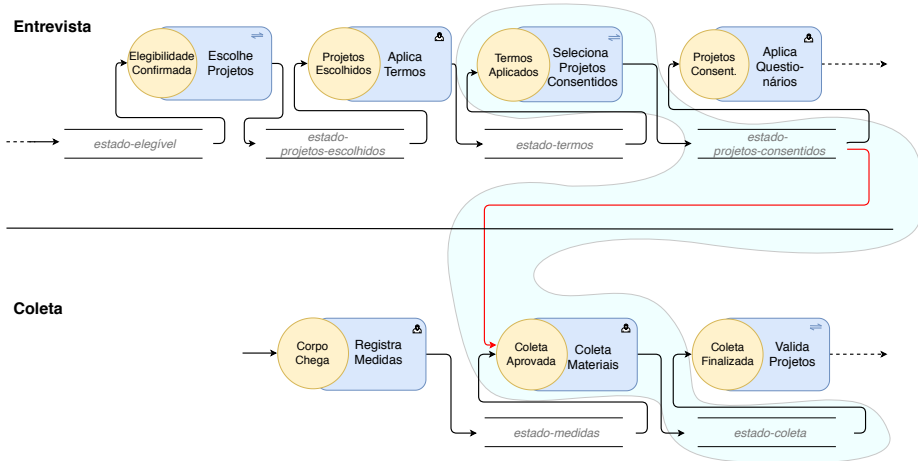
¹[Ferreira et al., 2012, IEEE-ICWS] [Schwerz et al., 2018, IEEE-Journal-TSC]



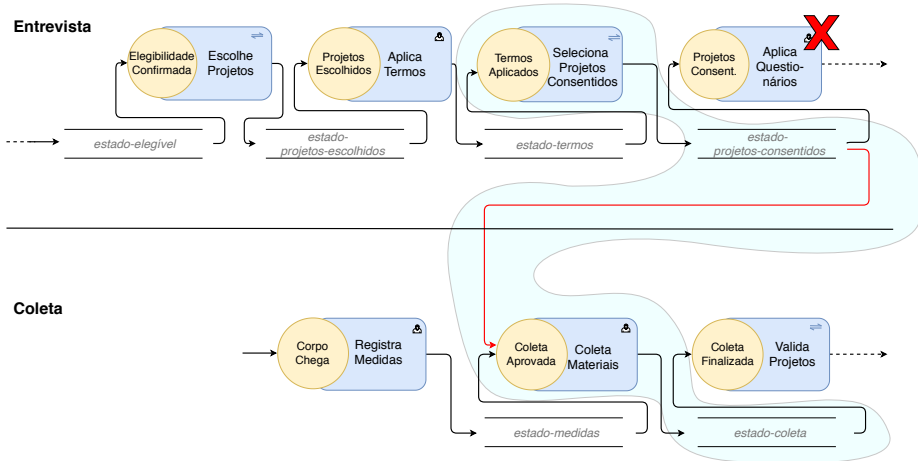
¹[Ferreira et al., 2012, IEEE-ICWS] [Schwerz et al., 2018, IEEE-Journal-TSC]



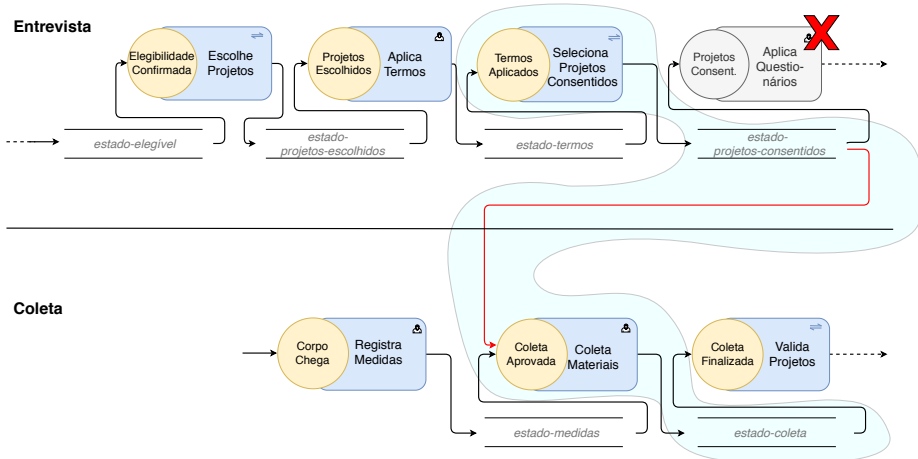
¹[Ferreira et al., 2012, IEEE-ICWS] [Schwerz et al., 2018, IEEE-Journal-TSC]



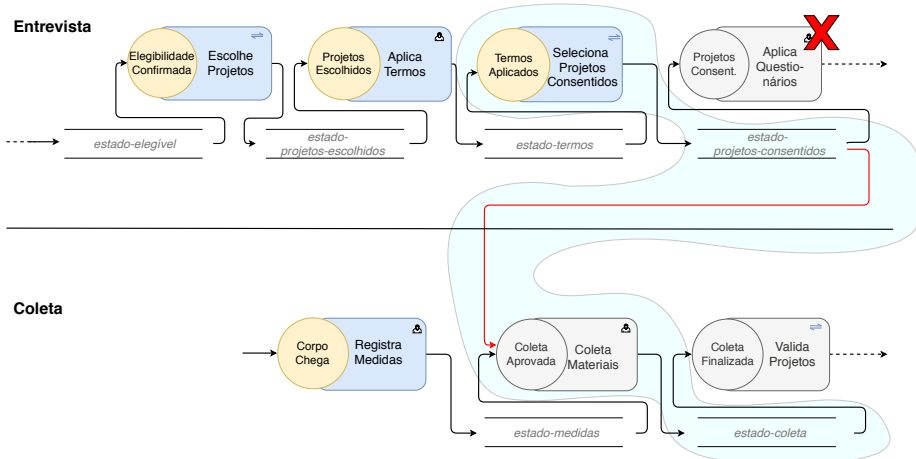
¹[Ferreira et al., 2012, IEEE-ICWS] [Schwerz et al., 2018, IEEE-Journal-TSC]



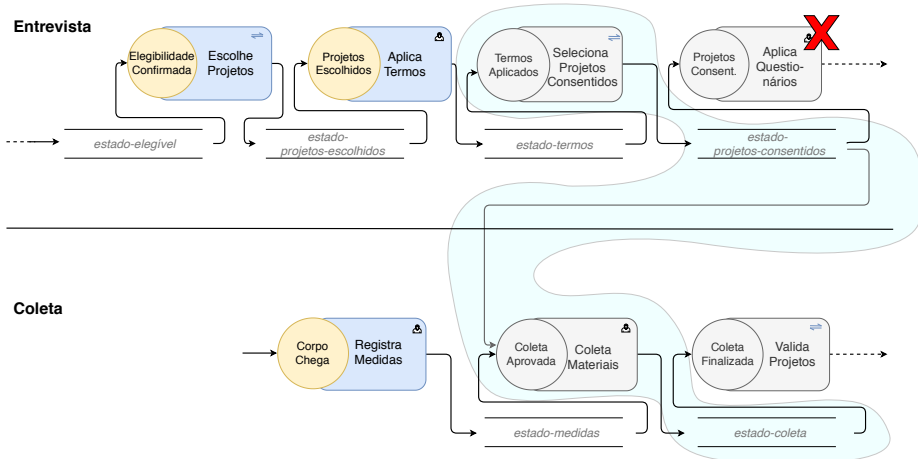
¹[Ferreira et al., 2012, IEEE-ICWS] [Schwerz et al., 2018, IEEE-Journal-TSC]



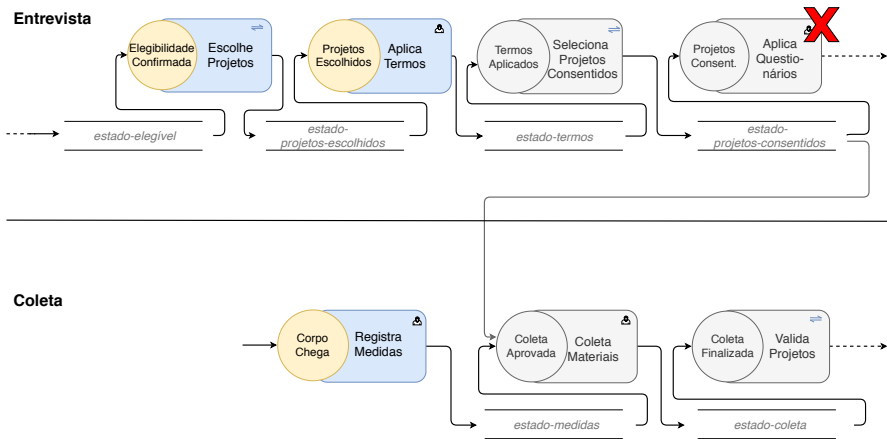
¹[Ferreira et al., 2012, IEEE-ICWS] [Schwerz et al., 2018, IEEE-Journal-TSC]



¹[Ferreira et al., 2012, IEEE-ICWS] [Schwerz et al., 2018, IEEE-Journal-TSC]



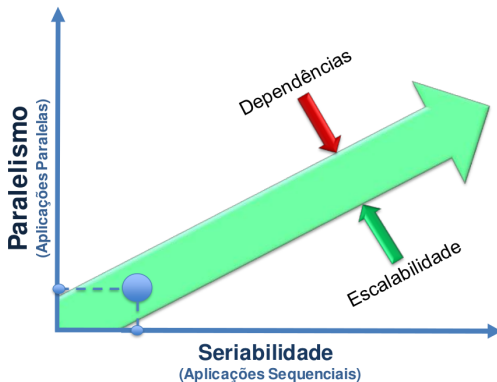
¹[Ferreira et al., 2012, IEEE-ICWS] [Schwerz et al., 2018, IEEE-Journal-TSC]



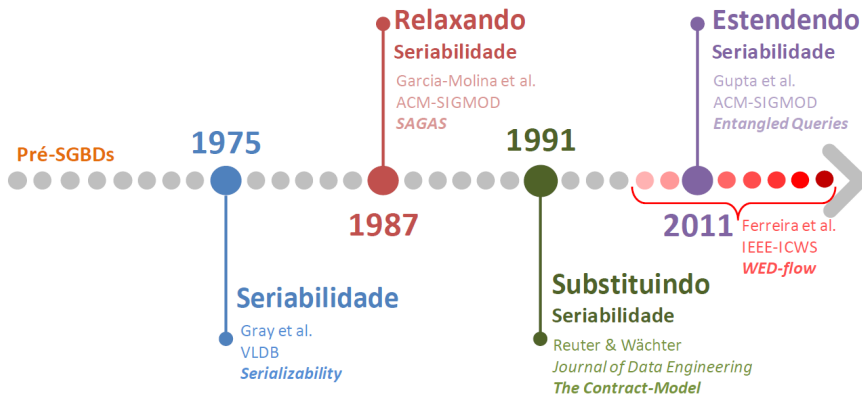
¹[Ferreira et al., 2012, IEEE-ICWS] [Schwerz et al., 2018, IEEE-Journal-TSC]

As especificações WED-flow:

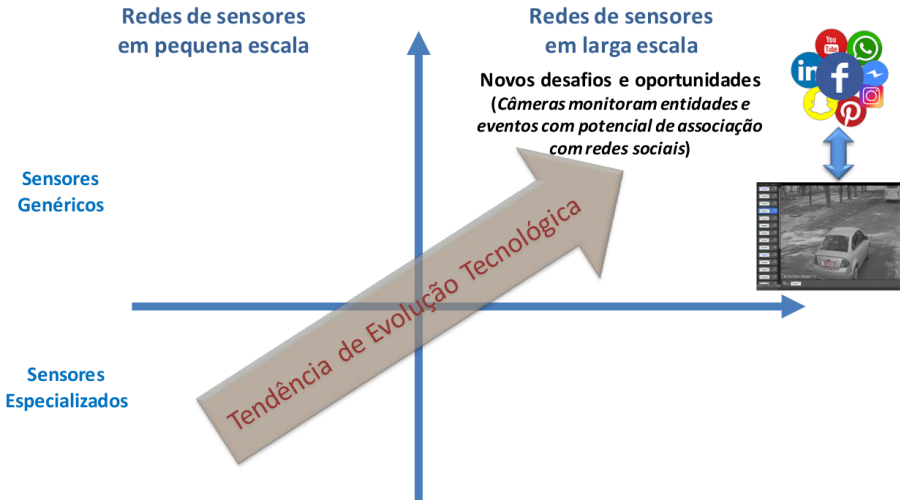
- incluem **propriedades** dos sistemas transacionais e regras ECA;
- garantem **consistências e dependências** dos dados;
- são **traduzidas** para código SQL executável.

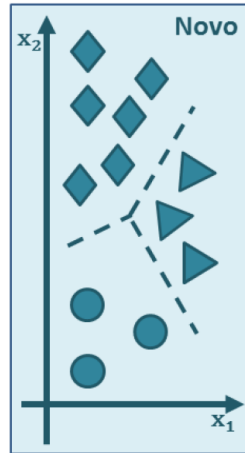
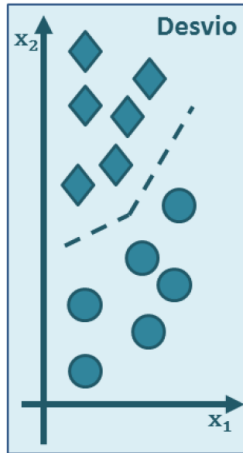
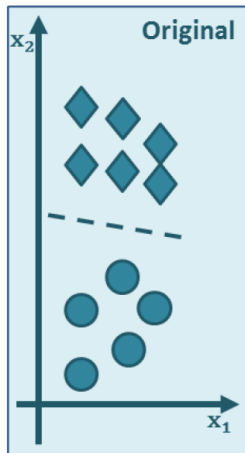


- 1 Introdução e desafios
- 2 Os limites dos modelos transacionais
- 3 WED-flow e processos complexos
- 4 Contextualização da contribuição**
- 5 Próximos passos
- 6 Conclusão



- 1 Introdução e desafios
- 2 Os limites dos modelos transacionais
- 3 WED-flow e processos complexos
- 4 Contextualização da contribuição
- 5 Próximos passos**
- 6 Conclusão



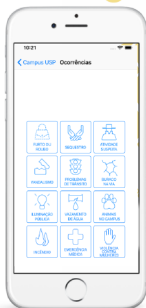


$$X \triangleq \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in R^2$$

$$Y \in \{0, 1\}$$

Distribuição de probabilidade conjunta: $P(X, Y) = P(Y|X) \cdot P(X)$

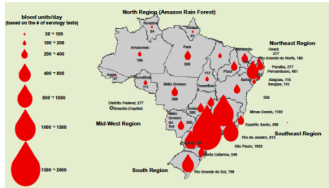
¹[Sethi and Kantardzic, 2017, ACM-Journal]



App e-Campus



REDS-NIH

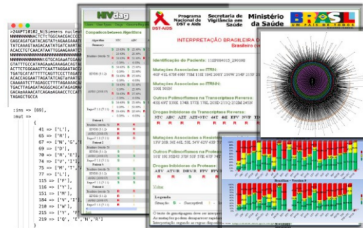


[Carneiro-Proietti et al., 2007]

[Sabino et al., 2011]

[Takecian et al., 2013]

HIV-Brazil

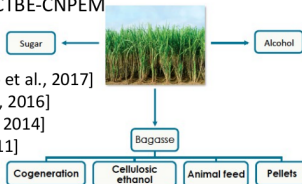


[Barreto et al., 2006]

[Araujo et al., 2006, 2011]

[Cintha et al., 2015]

CTBE-CNPEN



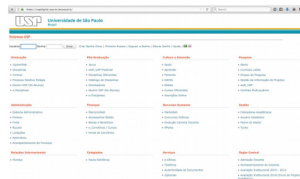
[Yancy-Caballero et al., 2017]

[Driemeier et al., 2016]

[Driemeier et al., 2014]

[Souza et al., 2011]

Portal da USP



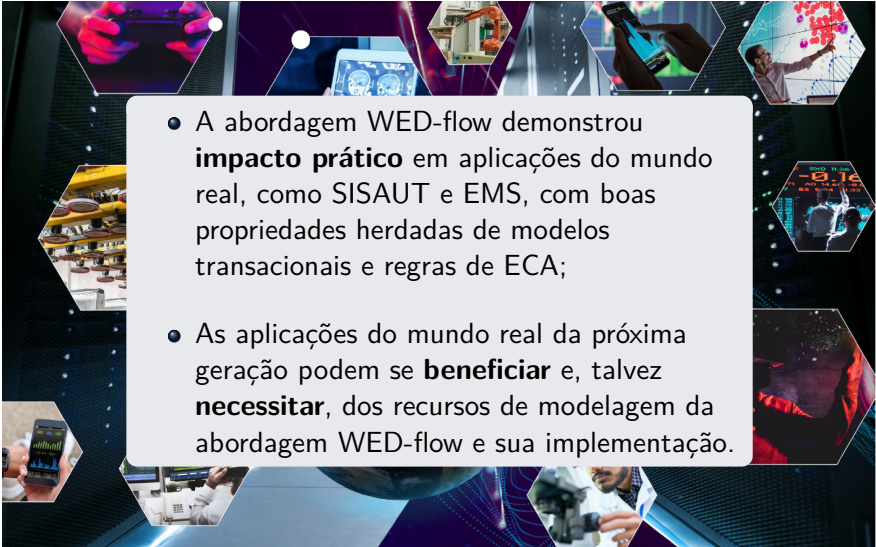
- FAPESP (Temático) - 2015/01587-0.

Storage, Modeling and Analysis of Dynamical Systems for e-Science Applications.

- INOVA; PRP-USP - Edital Grandes Projetos 668/2018.

Smart Detection and Tracking of Events and Entities in Cybernetic Physical Systems.

- 1 Introdução e desafios
- 2 Os limites dos modelos transacionais
- 3 WED-flow e processos complexos
- 4 Contextualização da contribuição
- 5 Próximos passos
- 6 Conclusão**

- 
- A collage of various images related to technology and industry, including hands holding a smartphone, a medical scanner, a factory floor, a person using a tablet, a person pointing at a screen, a person holding a smartphone, a person using a computer, and a person holding a smartphone. The images are arranged in a circular pattern around a central text box.
- A abordagem WED-flow demonstrou **impacto prático** em aplicações do mundo real, como SISAUT e EMS, com boas propriedades herdadas de modelos transacionais e regras de ECA;
 - As aplicações do mundo real da próxima geração podem se **beneficiar** e, talvez **necessitar**, dos recursos de modelagem da abordagem WED-flow e sua implementação.

- Estudantes, Pesquisadores (coautores e associados)
- FAPESP, CNPq, CAPES, NIH
- PRP-USP, IME-USP



IME-USP

Referências



J.Gray, R. Lorie, G.Putzulo and I.Traiger (1975)

Granularity of locks in shared databases.

In Proceeding of the 1st International Conference on Very Large Database (VLDB), pages 428-451.



P.A.Bernstein, V. Hadzilacos, and N.Goodman (1987)

Concurrency Control and Recovery in Database Systems.

Addison-Wesley Pub. Co. Inc., Reading, MA



H. Garcia-Molina, K. Salem (1987)

SAGAS.

Proceedings of the 1987 ACM SIGMOD International Conference on Management of Data (SIGMOD), pages 249-259.



Reuter, A.; Wächter, Helmut, (1991)

The Contract-Model.

Journal of Data Engineering, Volume 14 Issue 1, March 1991, pages 39-43, IEEE Computer Society Press.



Jim Gray and Andreas Reuter (1993)

Transaction Processing: Concepts and Techniques.

The Morgan Kaufmann Series in Data Management Systems, 1069 pages.



João Eduardo Ferreira and Marcelo Finger (2000)

Controle de Concorrência e Distribuição de Dados: a teoria clássica, suas limitações e extensões modernas.

X Escola de Computação, São Paulo, Brazil, ISBN 3540735216, 180 pages.



L. V. Araújo, M. A. Soares, S. M. Oliveira, P. Chequer, A. Tanuri, E. C. Sabino and J. E. Ferreira (2006).

DBCollHIV: a database system for collaborative HIV analysis in Brazil.

Genet Mol Res 5.1 (2006), issn: 1676-5680, p. 203-215.



Claudia C. Barreto, Luciano V. Araujo, Joao E. Ferreira and Ester C. Sabino (2006).

Trends in Antiretroviral Drug Resistance and Clade Distributions Among HIV-1-Infected Blood Donors in Sao Paulo, Brazil.

AIDS Journal of Acquired Immune Deficiency Syndromes 41.3, issn: 1525-4135, pg 338-341.

Referências



Anna Bárbara Carneiro-Proietti et al. (2007).

Demographic profile of blood donors at three major Brazilian blood centers: results from the International REDS-II study, 2007 to 2008.

Transfusion 50.4 (2009).



João E. Ferreira, Osvaldo K. Takai and Calton Pu. (2010)

Reducing Exception Handling Complexity in Business Process Modeling and Implementation: The WED-Flow Approach. *On the Move to Meaningful Internet Systems: OTM 2010. OTM 2010. Lecture Notes in Computer Science*, vol 6426.

Springer, Berlin, Heidelberg, ISBN 978-3-642-16934-2 , pg 150-167.



Kot, Lucja and Gupta, Nitin and Roy, Sudip and Gehrke, Johannes and Koch, Christoph. (2010).

Beyond Isolation: Research Opportunities in Declarative Data-Driven Coordination.

SIGMOD Record. 39. 27-32. 10.1145/1860702.1860706. .



Luciano Araujo, Simon Malkowski, Kelly R. Braghefo, Maria P. Bueno, Mayana Zatz, Calton Pu and João E. Ferreira (2011).

A rigorous approach to facilitate and guarantee the correctness of the genetic testing management in human genome information systems.

BMC Genomics. 2011;12 Suppl 4(Suppl 4):S13. doi:10.1186/1471-2164-12-S4-S13.



Gupta, Nitin and Kot, Lucja and Roy, Sudip and Bender, Gabriel and Gehrke, Johannes and Koch, Christoph (2011).

Entangled Queries: Enabling Declarative Data-driven Coordination.

Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, ISBN 978-1-4503-0661-4, p 673-684.



Gupta, Nitin and Nikolic, Milos and Roy, Sudip and Bender, Gabriel and Kot, Lucja and Gehrke, Johannes and Koch, Christoph, (2011).

Entangled Transactions.

Proceedings of the VLDB Endowment.

Referências



Ester C. Sabino, Thelma T. González, Anna B. Carneiro-Proieg, Moussa Sarr, João E. Ferreira, Divaldo A. Sampaio, Nanci A. Salles, David J. Wright, Brian Custer and Michael Busch (2011).
Human immunodeficiency virus prevalence, incidence, and residual risk of transmission by transfusions at Retrovirus Epidemiology Donor Study-II blood centers in Brazil.
Transfusion 52.4 (2011), p 870-879.



Glauca Mendes Souza, Helene Berges, Stephanie Bocs, Rosanne Casu, Angelique D'Hont, J. E. Ferreira, Robert Henry, Ray Ming, Bernard Potier, Marie-Anne Sluys, Michel Vincentz and Andrew H. Paterson (2011).
The Sugarcane Genome Challenge: Strategies for Sequencing a Highly Complex Genome.
Tropical Plant Biology (Print), v. 4, p. 145-156



João Eduardo Ferreira, Kelly Rosa Braghetto, Osvaldo Kotaro Takai e Calton Pu. (2012)
Transactional recovery support for robust exception handling in business process services.
Proceedings of the 19th International Conference on Web Services (ICWS), Honolulu, HI, USA, pg 303 - 310.



Pedro L. Takecian et al. (2013).
Methodological Guidelines for Reducing the Complexity of Data Warehouse Development for Transactional Blood Bank Systems.
Decision support systems vol. 55,3 (2013), doi:10.1016/j.dss.2013.02.008, p. 728-739.



Carlos Eduardo Driemeier, Liu Yi Ling, Angelica O. Pontes, Guilherme M. Sanches, Henrique C. J. Franco, Paulo S. G. Magalhaes and J. E. Ferreira (2014).
Data Analysis Workflow for Experiments in Sugarcane Precision Agriculture.
IEEE 10th International Conference on eScience (eScience), 2014, p. 163.



Mina C. Ozahata, Ester C. Sabino, Ricardo S. Diaz, Roberto M. Cesar e João E. Ferreira (2015).
Data-intensive analysis of HIV mutations.
BMC Bioinformatics 16.1 (2015), p. 35. issn: 1471-2105.

Referências



Carlos Driemeier, Liu Yi Ling, Guilherme M. Sanches, Angélica O. Pontes, Paulo S. Graziano Magalhães and J. E. Ferreira (2016).

A computational environment to support research in sugarcane agriculture.
Computers and Electronics in Agriculture, v. 130, p. 13-19.



Rafael Liberato Roberto (2016).

Aumentando a migração de instâncias não complacentes em sistemas de informação cientes de processos.
PhD Thesis.



André Luis Schwerz (2016).

Sistemas de Informação Cientes de Processos, Robustos e Confiáveis.
PhD Thesis.



J. E. Ferreira, P. Losco Takecian, L. T. Kamaura, B. Padilha e C. Pu. (2017).

Dependency Management with WED-flow Techniques and Tools: A Case Study.
2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC).



ethi, Tegjyot Singh and Kantardzic, Mehmed, (2017).

On the Reliable Detection of Concept Drift from Streaming Unlabeled Data.
ACM-Journal of Expert Systems with Applications.



Daison Yancy-Caballero, Liu Y. Ling, Nathaly L. Archilha, J. E. Ferreira and C. Driemeier (2017).

Mineral particles in sugar cane bagasse: Localization and morphometry using microtomography analysis.
Energy & Fuels (Online), v.31, p. 12288-12296



Ferreira et al., (2018).

Integrating the University of São Paulo Security Mobile App to the Electronic Monitoring System.
IEEE-BigData 2018.

Referências



Rodrigo A. Lima, Calton Pu, Bruno Padilha, Pedro L. Takecian, Leonardo T. Kamaura e João E. Ferreira (2018).
Modeling Time-Critical Processes with WED-Flow.
Web Services - ICWS 2018 - 25th International Conference, ISBN 978-3-319-94288-9, pg 34-48



B. Padilha, R. L. Roberto, A. L. Schwerz, C. Pu and J. E. Ferreira (2018).
WED-SQL: An Intermediate Declarative Language for PAIS Execution.
ICWS. Vol. 10966. Lecture Notes in Computer Science. Springer, 2018, ISBN 978-3-319-94288-9, pg 407-421.



A. L. Schwerz, R. Liberato, C. Pu and J. E. Ferreira (2018).
Robust and Reliable Process-Aware Information Systems.
IEEE Transactions on Services Computing, issn: 1939-1374, pg 1-1



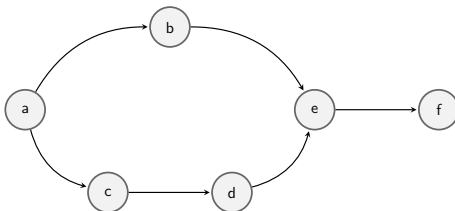
Rodrigo A. Lima et al., (2019).
Systematic Construction, Execution, and Reproduction of Complex Performance Benchmarks.
CLOUD 2019.

Transações clássicas¹

Definição (10 - **Ordem parcial estrita**)

Uma ordem parcial estrita (irreflexiva) $<$ sobre um conjunto Σ é uma relação binária em $\Sigma \times \Sigma$ (é um conjunto de pares ordenados), representada por $(\Sigma, <)$ satisfazendo as seguintes restrições:

- a ordem $<$ é transitiva: para todo $x, y, z \in \Sigma$, se $x < y$ e $y < z$ então $x < z$;
- a ordem $<$ é irreflexiva: para todo $x \in \Sigma$, $x \not< x$.

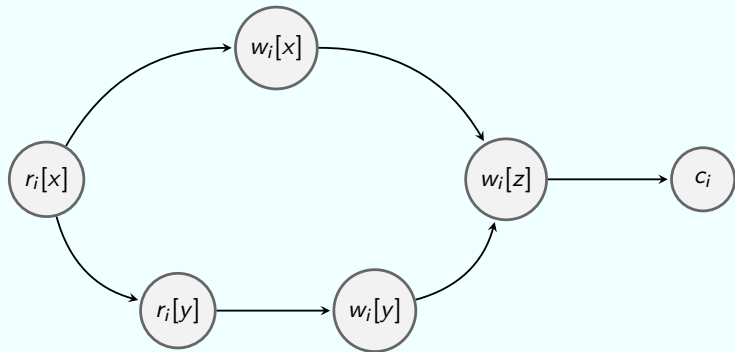


Grafo dirigido (ordem parcial transitiva) e acíclico (ordem parcial irreflexiva).

¹[Gray et al., 1975, VLDB] [Bernstein et al., 1987] [Ferreira and Finger, 2000]

Transações clássicas¹

Exemplo ($\mathcal{T}_i$)



¹[Gray et al., 1975, VLDB] [Berstein et al, 1987] [Ferreira and Finger, 2000]

Transações clássicas¹

Definição (11 - Escalonamento completo)

Um escalonamento completo sobre o conjunto de transações $\{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n\}$ é uma ordem parcial $(\mathcal{E}, <_{\mathcal{E}})$ tal que:

- $\mathcal{E} = \mathcal{T}_1 \cup \mathcal{T}_2 \cup \dots \cup \mathcal{T}_n$;
- $<_{\mathcal{E}} \supseteq <_1 \cup <_2 \cup \dots \cup <_n$; ou seja, a ordenação do escalonamento estende a ordenação das operações internas a cada transação;
- para cada par de **operações conflitantes** p_i e q_j , $p_i <_{\mathcal{E}} q_j$ ou $q_j <_{\mathcal{E}} p_i$.

Operações conflitantes de $\mathcal{T}_i$ e $\mathcal{T}_j$: $\{r_i, w_j\}$, $\{w_i, r_j\}$ e $\{w_i, w_j\}$

¹[Gray et al., 1975, VLDB] [Berstein et al, 1987] [Ferreira and Finger, 2000]

Transações clássicas

Definição (12 - Escalonamento serial)

Um escalonamento completo $\mathcal{E}$ é dito serial se, para cada par de transações $\mathcal{T}_i, \mathcal{T}_j$ em $\mathcal{E}$, todas as operações de $\mathcal{T}_i$ precedem todas as operações de $\mathcal{T}_j$ em $\mathcal{E}$, ou vice-versa.

$$\mathcal{E} = \mathcal{T}_i \cdot \mathcal{T}_j \text{ OU } \mathcal{E} = \mathcal{T}_j \cdot \mathcal{T}_i$$

Exemplo (Escalonamento serial)

T_1		T_2
$s := \text{Lê}(\text{SALDO})$	$\mathcal{E} = T_1 \cdot T_2$	$s := \text{Lê}(\text{SALDO})$
$s := s + 100$	OU	$s := s + 0,1*s$
Escreve(SALDO, s)	$\mathcal{E} = T_2 \cdot T_1$	Escreve(SALDO, s)
COMMIT ₁		COMMIT ₂

Escalonamento serial não é eficiente!

Definição (13 - Equivalência de escalonamentos)

Dois escalonamentos $\mathcal{E}$ e $\mathcal{E}'$ são equivalentes, $\mathcal{E} \equiv \mathcal{E}'$, se:

- tanto $\mathcal{E}$ quanto $\mathcal{E}'$ são definidos sobre o mesmo conjunto de transações e sobre as mesmas operações;
- ambos ordenam operações conflitantes de transações não abortadas (confirmadas ou ativas) igualmente. Ou seja, se p_i e q_j são operações conflitantes e $a_i \notin \mathcal{T}_i$, $a_j \notin \mathcal{T}_j$:

$$p_i <_{\mathcal{E}} q_j \iff p_i <_{\mathcal{E}'} q_j$$

¹[Gray et al., 1975, VLDB] [Berstein et al, 1987] [Ferreira and Finger, 2000]

Transações clássicas: um exemplo concreto

E_1	
T_1	T_2
$s := L(SALDO)$	--
$s := s + 100$	--
--	$s := L(SALDO)$
--	$s := s + 0,1*s$
$E(SALDO, s)$	--
--	$E(SALDO, s)$
$COMMIT_1$	--
--	$COMMIT_2$

Essa
operação
foi perdida
→

$s = 1.000$

$s_{fim} = 1.100$

E_1 não é seriável

E_2	
T_1	T_2
$s := L(SALDO)$	--
$s := s + 100$	--
$E(SALDO, s)$	--
--	$s := L(SALDO)$
--	$s := s + 0,1*s$
$COMMIT_1$	--
--	$E(SALDO, s)$
--	$COMMIT_2$

$s = 1.000$

$s_{fim} = 1.210$

E_2 é seriável

Classic transactions

Proof: $\mathcal{E}$ is serializable $\iff GS(\mathcal{E})$ is acyclic .

We need to find $\mathcal{E}_s \equiv \mathcal{E}$.

Since $GS(\mathcal{E})$ is acyclic, it can be topologically sorted.

Let $\mathcal{E}_s$ be a serial scheduling from a topological sort of $GS(\mathcal{E})$.

Given a couple of conflicting operations p_i, q_j , if $p_i <_{\mathcal{E}} q_j \implies T_i \rightarrow T_j$ in $GS(\mathcal{E})$, and therefore any topological sort of $GS(\mathcal{E})$ respects the ordering, that is, $p_i <_{\mathcal{E}_s} q_j$.

On the other hand, If $p_i <_{\mathcal{E}_s} q_j \implies T_i \rightarrow T_j$ in $GS(\mathcal{E})$

Since $GS(\mathcal{E})$ is acyclic, $\nexists T_j \rightarrow T_i \in GS(\mathcal{E})$. Thus $\mathcal{E}_s \equiv \mathcal{E}$. □

Classic transactions

Proof: $\mathcal{E}$ is serializable $\implies GS(\mathcal{E})$ is acyclic .

If $\mathcal{E}$ is serializable then $\exists \mathcal{E}_s$ (a serial scheduling) such that $\mathcal{E} \equiv \mathcal{E}_s$.

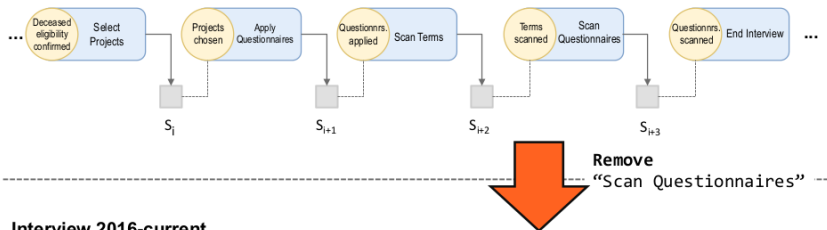
Let $T_i \rightarrow T_j \in GS(\mathcal{E})$, $p_i <_{\mathcal{E}} q_j$.

$\mathcal{E}_s$ preserves the order of conflicting operations, $p_i <_{\mathcal{E}_s} q_j$ always when $T_i \rightarrow T_j \in GS(\mathcal{E})$.

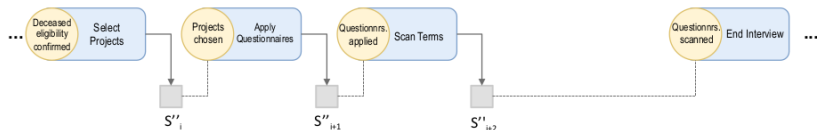
Suppose there is a cycle in $GS(\mathcal{E}) : T_{i1} \rightarrow T_{i2} \rightarrow T_{i3} \rightarrow \dots \rightarrow T_{ik} \rightarrow T_{i1}$. Then, we should have $\mathcal{E}_s : T_{i1} \rightarrow T_{i2} \rightarrow T_{i3} \rightarrow \dots \rightarrow T_{ik} \rightarrow T_{i1}$, which is a contradiction, given that T_{i1} occurs only once ($\mathcal{E}_s$ is a serial scheduling). Thus, $GS(\mathcal{E})$ is **acyclic**. □

WED-flow projection¹

Interview 2014-2016



Interview 2016-current



¹[Roberto, 2017] [Roberto et al., 2019] [Ferreira et al., 2019]

Temporal WED-flow¹

Definição (14 - **WED-attributes**)

WED-attributes + $\langle v_i, v_f, dt \rangle$

v_i and v_f define the validity period and
dt is date-time creation of attribute i.

Definição (15 - **WED-state**)

WED-state + $\langle dt \rangle$

dt is date time creation of attribute i.

Definição (16 - **WED-trigger**)

WED-trigger + $\langle v_i, v_f, dt \rangle$

v_i and v_f define the validity period and
dt is date-time creation of attribute i.

¹[Roberto, 2017] [Roberto et al., 2019] [Ferreira et al., 2019]

Definição (17 - **WED-flow SuperSchema**)

The WED-flow SuperSchema (WS) is the union of WED-flow (G), WED-attributes (A) and WED-states (S). $WS = G \cup A \cup S$

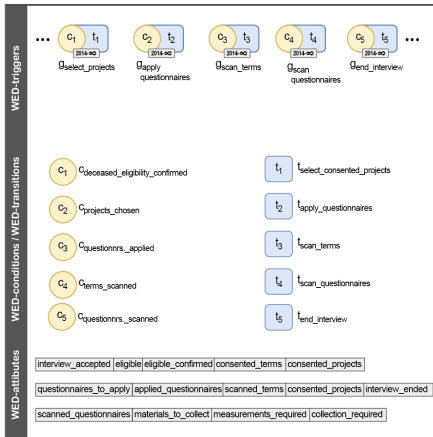
Definição (18 - **WED-flow Superschema Projection**)

The projection $\pi_{(G,2014)}$ projects the set of WED-flow Schema with their validity period using date-time dt.

¹[Roberto, 2017] [Roberto et al., 2019] [Ferreira et al., 2019]

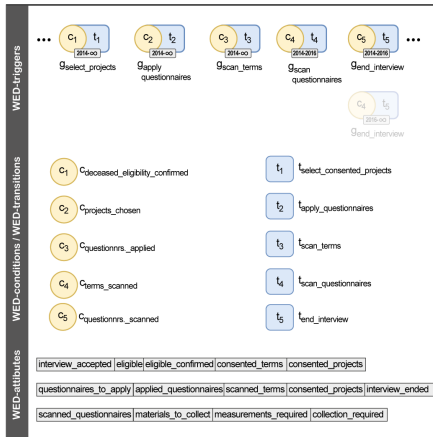
WED-flow projection¹

WED-flow



Superschema

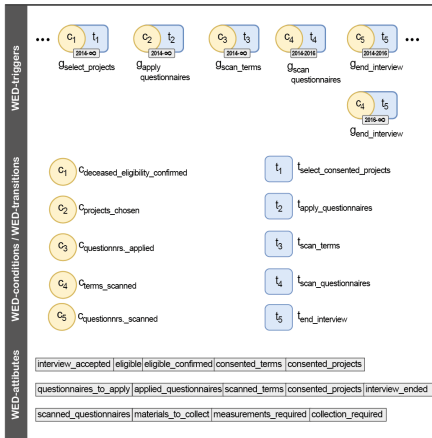
$\pi_{WS,2014}$



¹[Roberto, 2017] [Roberto et al., 2019] [Ferreira et al., 2019]

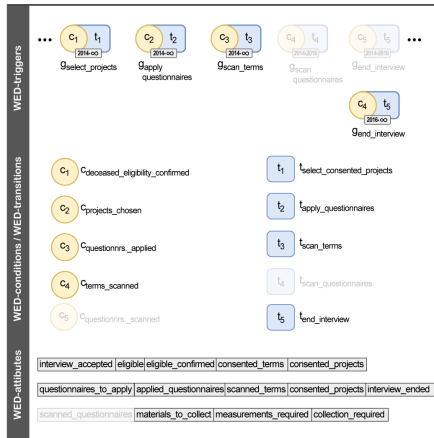
WED-flow projection¹

WED-flow



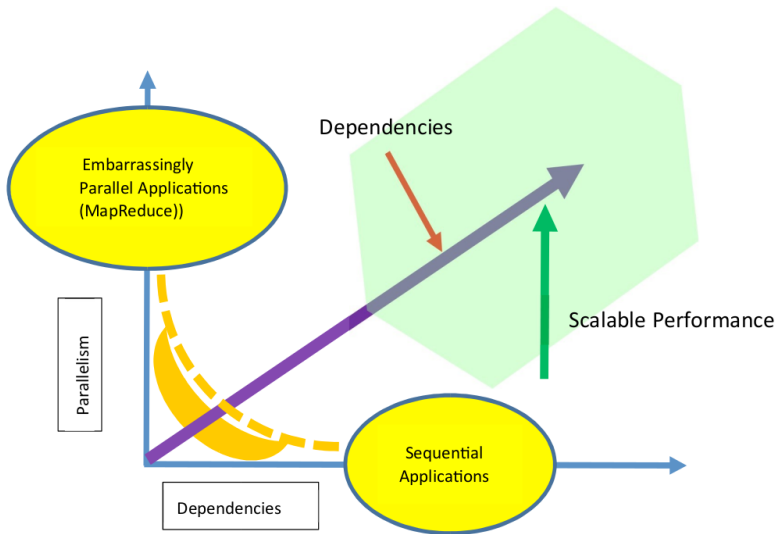
Superschema

$\pi_{WS,2016}$



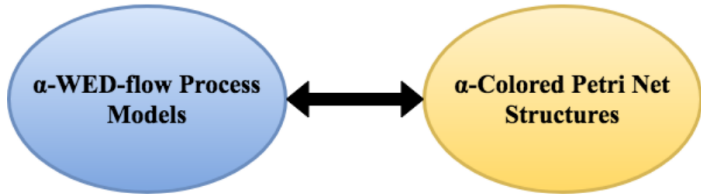
¹[Roberto, 2017] [Roberto et al., 2019] [Ferreira et al., 2019]

Dynamic Dependencies



Modeling time-critical process¹

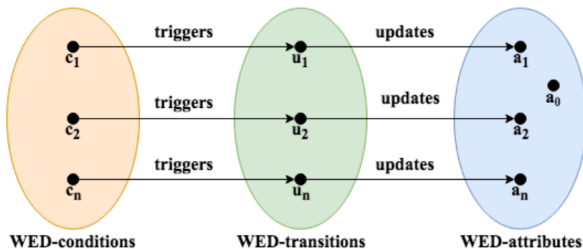
- α -WED-flow is a restriction over WED-flow
- α -CPN is a restriction over CPN



¹ Lima, 2018 - ICWS

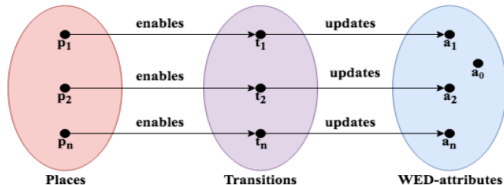
α -WED-flow: restriction

1. A WED-transition u_k updates the value of a single WED-attribute a_k initialized as null.
2. c_k is the only WED-condition associated with WED-transition u_k . Such a WED-condition c_k is a conjunction of one or more clauses testing the values of WED-attributes a_j , $0 \leq j \leq n$. Also, c_k necessarily tests the value of a_k for null.
3. $c_i: a_0 = \text{true}$ and $c_f: a_n = \text{true}$



α -CPN: restriction

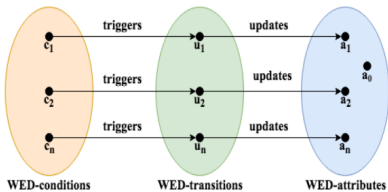
1. Place p_k is the single input place of transition t_k . Their connecting arc is annotated with a conjunction of one or more clauses testing the values of WED-attributes a_j , $0 \leq j \leq n$. Also, this logical predicate necessarily tests the value of a_k for null.
2. For all $1 \leq j \leq n$, $j \neq k$, a transition t_j is connected to place p_k if the logical predicate annotated on the arc connecting p_k to transition t_k tests the value of WED-attribute a_j .
3. **A valuation of a_j is attached to the tokens produced by the firing of t_j .** For example, if a_j is a Boolean WED-attribute, either $a_j = \text{true}$ or $a_j = \text{false}$ is attached to the tokens produced by the firing of t_j .



α -WED-flow $\iff$ α -CPN

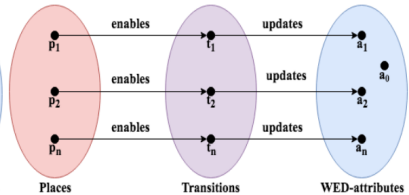
WED-flow constructs

- α -WED-flow restrictions



CPN constructs

- α -CPN restrictions



Equivalence between α -WED-flow process models and α -CPN structures

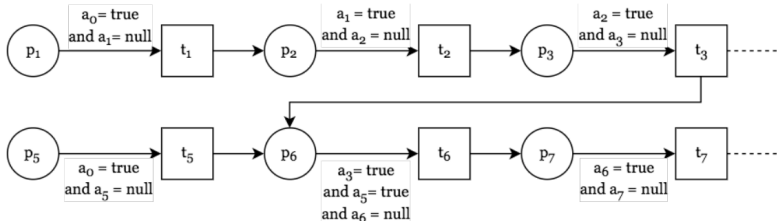
- By construction, there exists a correspondence between:

α -WED-flow	α -CPN
WED-transition u_k	Transition t_k
WED-condition c_k	Place p_k
WED-state (valuation of WED-attributes)	Values attached to tokens

- We also need to show that the execution semantics of corresponding α -WED-flow process models and α -CPN structures are equivalent.

Equivalence between α -WED-flow process models and α -CPN structures

WED-transition	Associate WED-condition
u_1 (Gather data and choose projects)	$c_1: a_0 = \text{true}$ and $a_1 = \text{null}$
u_2 (Consent with terms)	$c_2: a_1 = \text{true}$ and $a_2 = \text{null}$
u_3 (Select consented projects)	$c_3: a_2 = \text{true}$ and $a_3 = \text{null}$
u_4 (Fill out questionnaires)	$c_4: a_3 = \text{true}$ and $a_4 = \text{null}$
u_5 (Register body measurements)	$c_5: a_0 = \text{true}$ and $a_5 = \text{null}$
u_6 (Collect materials)	$c_6: a_3 = \text{true}$ and $a_5 = \text{true}$ and $a_6 = \text{null}$
u_7 (Check materials)	$c_7: a_6 = \text{true}$ and $a_7 = \text{null}$
u_8 (Process materials)	$c_8: a_7 = \text{true}$ and $a_8 = \text{null}$
u_9 (Scan paperwork)	$c_9: a_4 = \text{true}$ and $a_8 = \text{true}$ and $a_9 = \text{null}$



Desvio de Conceito (*Concept Drift*)¹

Modelos construídos para analisar dados, que evoluem intensamente ao longo do tempo e devem ser analisados quase em tempo real, tornam-se obsoletos rapidamente. Na área de aprendizado de máquina, esse fenômeno é chamado de Desvio de Conceito (“Concept Drift”).

¹[Sethi and Kantardzic, 2017, ACM-Journal]

Desvio de Conceito (*Concept Drift*)¹

Distribuição de probabilidade conjunta

Assumindo uma perspectiva probabilística, o Desvio de Conceito pode ser visto como uma mudança na distribuição de probabilidade conjunta das amostras de dados X e seus rótulos de classes correspondentes Y :

$$P(X, Y) = P(Y|X) \cdot P(X)$$

¹[Sethi and Kantardzic, 2017, ACM-Journal]

Definição Redes de Petri

A definição formal de uma rede de Petri é dada pela quintupla $RP = (P, T, F, W, M_0)$ em que:

$P = \{p_1, p_2, \dots, p_m\}$ é um conjunto finito de lugares;

$T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições;

$F \subseteq (P \times T) \cup (T \times P)$ é um conjunto de arcos;

$W : F \rightarrow \{1, 2, 3, \dots\}$ é a função de peso dos arcos;

$M_0 : P \rightarrow \{0, 1, 2, \dots\}$ é a marcação inicial da rede;

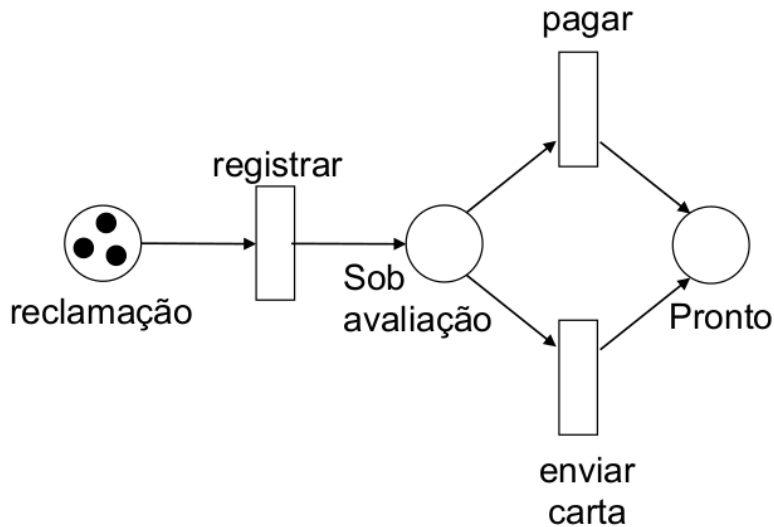
$P \cap T = \emptyset$ e $P \cup T \neq \emptyset$.

Uma estrutura de rede de Petri $N = (P, T, F, W)$ sem especificação da marcação inicial denota-se por N .

Workflow-net: a restriction of Petri net

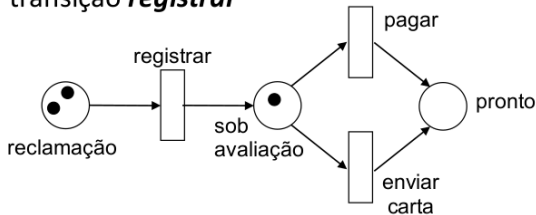
- **Workflow-net:** modela o fluxo de controle de um workflow:
 - Possui apenas um lugar fonte (início) e um absorção (fim)
 - Todo no do grafo está em um caminho do lugar fonte ao lugar absorção
- **Transições:** representam tarefas
- **Lugares:** representam condições que podem ser usadas como pré e/ ou pós-condições das tarefas
- **Arcos:** dependências causais

Exemplo – WF-net

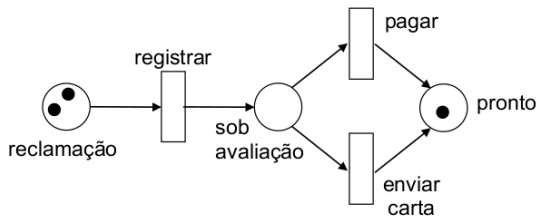


Disparo de transição

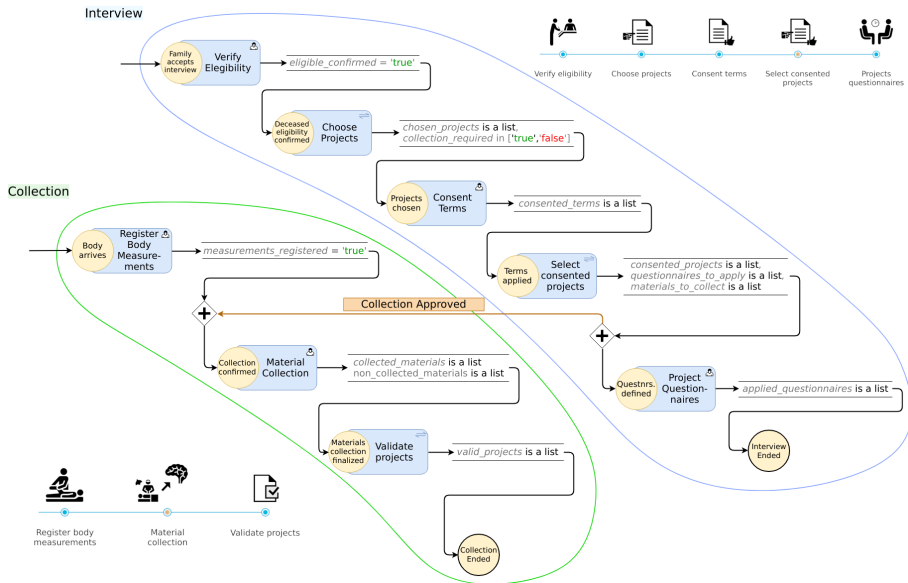
Disparo da transição **registrar**



Disparo da transição **pagar** ou **enviar carta**

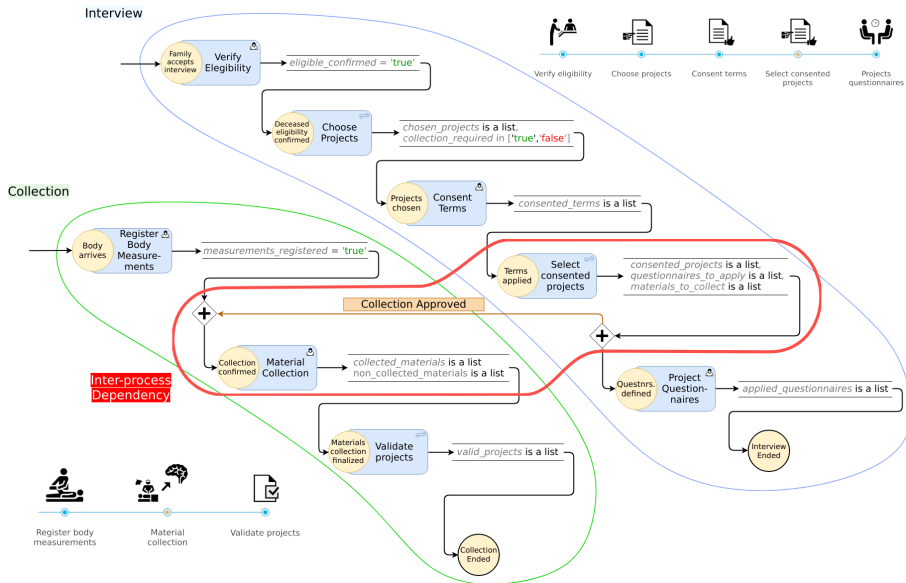


SISAUT usando diagrama WED-flow



Figura

SISAUT usando diagrama WED-flow



Figura

SISAUT usando diagrama WED-flow

