

Framework para Digital Twins de Processos

Fernando Silva* Jorge Pinho de Sousa† João Basto‡

Resumo—O acelerado e constante desenvolvimento tecnológico vem contribuindo cada vez mais para a criação de novas soluções no ramo da automação. Com o surgir do conceito da Indústria 4.0, as empresas procuram fazer parte da vanguarda desta revolução, no âmbito de assegurar qualquer vantagem competitiva. O *Digital Twin* (DT), como pilar da Indústria 4.0, revela-se crucial na implementação deste conceito e vem ganhando uma grande popularidade na indústria transformadora.

A presente dissertação apresenta uma *framework* para *Digital Twins* de processos que permite monitorizar em tempo real o chão de fábrica, fazer a simulação do futuro a partir de planos de produção e comunicar com serviços inteligentes externos. Efectivamente, a monitorização permite a visualização 3D dos eventos que estão a ocorrer na fábrica. O modo de simulação consegue simular o tráfego no chão de fábrica dado um plano de produção introduzido pelo utilizador. Por fim, a comunicação com serviços externos reforça a interoperabilidade da *framework*, sendo que neste projeto se recorre a um programa de otimização. Estas três componentes são a base da estrutura desenvolvida com que se procura potenciar as capacidades dos DTs de processos.

Index Terms—Digital Twin, Indústria 4.0, Simulação, Sistemas de produção.

I. INTRODUÇÃO

COM a introdução da era da indústria 4.0, deu-se vida a diversos novos conceitos que suportam a implementação da mesma. Um desses é o *Digital Twin*, que tal como o seu nome indica pode ser descrito como uma cópia digital de um objeto, de um processo ou até mesmo de um sistema. Apesar de a primeira definição de DT no ambiente industrial ser introduzida por Michael Grieves em 2002 [1], é em 2013 com o trabalho de Lee et al. [2] que o conceito DT se estende à representação virtual dos recursos de produção. Hoje em dia, no que toca à indústria transformadora, o DT é em geral definido como uma representação virtual de partes físicas ou componentes, produtos, processos ou até de sistemas complexos [3].

O grande objetivo desta dissertação consiste no planeamento e desenvolvimento de uma *framework* para *Digital Twins* de processos com visualização 3D de um *shop floor*, a ser usado por gestores de produção e de instalações para um melhor entendimento *real-time* do estado dos seus recursos operacionais e processos. Esta *framework* a permite sustentar a criação de DT de processos para diferentes tipos de produção, possibilitando também a simulação de planos de produção do utilizador e a comunicação com serviços inteligentes exteriores à *framework*.

O projeto foi dividido em três componentes - monitorização em tempo real, simulação e otimização. A primeira

responsabiliza-se pela criação de um modelo 3D do *shop floor* no *software FlexSim* que deve replicar em tempo real o estado e a posição das estações de trabalho e das peças a ser processadas. O sistema real é emulado por mensagens que são enviadas para uma aplicação programada em *Python*, que deve atualizar a base de dados (BD) e o *FlexSim* com os últimos eventos. Por seu turno, a componente de simulação, tal como o nome indica, permite fazer a simulação do futuro a partir de planos de produção introduzidos pelo utilizador. São criadas as peças deste plano que são processadas de forma isolada ou em paralelo com o último estado da tarefa de monitorização. Por fim, a componente de otimização procura demonstrar a interoperabilidade da *framework*, dado que se comunica com um serviço externo - neste caso um programa de otimização do plano de produção introduzido - capaz de atuar diretamente sobre o *Digital Twin*. A Figura 1 apresenta a arquitetura proposta para o projeto.

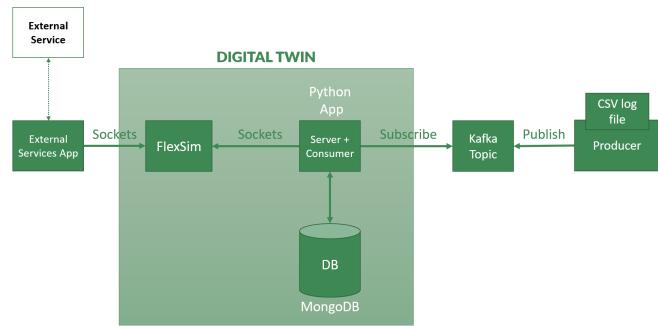


Figura 1. Arquitetura proposta

II. MONITORIZAÇÃO EM TEMPO REAL

A. Emulação da fábrica

Não existindo um sistema físico a replicar, surgiu a necessidade em criar uma simulação do que seria o comportamento num *shop floor*. Esta emulação deve ter em conta a entrada, o processamento e a saída de peças de estações de trabalho - ou *workstations* (WS) - de forma a replicar a movimentação de peças numa fábrica. A fábrica é constituída por *workstations*, sendo que cada uma contém um *buffer* de entrada onde aguardam as peças numa fila. Deste modo, é criado num ficheiro CSV a sequência de mensagens com os eventos a ocorrer, cujos parâmetros são: *Time Of Occurrence* (tempo de ocorrência), *Description* (tipo de evento - entrada, saída ou fim do processamento), *Associated Production Area* (estaçao de trabalho associada ao evento) e *Associated Agent* (identificador da peça). Estas mensagens são publicadas num tópico *Kafka*.

*up201604125@up.pt

†jsousa@fe.up.pt

‡joao.p.basto@inesctec.pt

B. Servidor Python

O servidor *Python* é responsável por três fluxos de informação: deve consumir as mensagens publicadas no tópico *Kafka*, manter a BD atualizada com as mudanças no *shop floor* e consultá-la sempre que necessário e reencaminhar para o *FlexSim* os eventos extraídos das mensagens.

C. Base de dados

O principal papel da BD passa por guardar uma *snapshot*, ou fotografia, do último estado do *shop floor*. É atualizada a sempre que ocorre um evento, sendo constituída por duas coleções - *objects* e *agents*. A coleção *object* é composta pelo identificador das estações de trabalho e *buffers* na fábrica, pelo estado das mesmas (a processar ou parado) e pelo identificador da peça que nelas se encontram, caso essa condição se verifique. Por outro lado, a coleção *agents* guarda o identificador das peças, a localização destas (referente à coleção *objects*) e o respetivo estado (a ser processada ou parada).

D. Processamento no *FlexSim*

É no *software FlexSim* que se cria o modelo 3D da fábrica. Aqui é feita a replicação visual do que ocorre na emulação criada. Esta componente é conseguida através de um fluxograma de processos, ferramenta do *FlexSim*, que permite fazer a lógica de movimentação no modelo. Quando uma mensagem é recebida pelo *FlexSim* é identificado o tipo de evento e conforme este, cria-se ou apaga-se uma peça na respetiva WS. Processada a mensagem, o fluxograma volta a ficar à escuta de mensagens do servidor.

III. SIMULAÇÃO

A. Localização de peças em processamento

A primeira parte da componente de simulação trata de localizar peças em processamento na tarefa de monitorização e de as mapear no fluxograma de simulação e no modelo visual (nas respetivas WS). Esta etapa parte da *snapshot* do último estado da fábrica, guardada na BD, a ser processada em paralelo com o plano de produção do utilizador. Caso a fábrica se encontre vazia, esta fase é ignorada.

B. Criação das peças a partir do plano de produção

Recriado o último estado da fábrica no modelo 3D, devem ser criadas as peças relativas ao plano de produção do utilizador. O plano deve ser introduzido numa tabela e a partir da leitura desta são criadas as peças numa fila inicial. Terminada a criação do plano de produção no modelo 3D, o fluxograma dá sinal para avançar com o processamento das peças.

C. Processamento das peças

Criadas as peças provenientes da monitorização e do plano de produção, resta fazer a processamento das peças, isto é, criar a lógica de movimentação pelas WS. Um fluxograma dedicado a esta etapa acede a uma tabela de processamento, que tem a sequência de processamento de cada tipo de peça, e a partir daí faz a movimentação das peças pelo *shop floor* dependendo dos seus tipos - as peças estão etiquetadas com o seu tipo. Terminada a sequência da peça, esta é retirada do modelo visual.

IV. OTIMIZAÇÃO

Por fim, a componente de otimização visa demonstrar a interoperabilidade da *framework* desenvolvida. É criada uma aplicação que faz a ponte entre o *FlexSim* e serviços exteriores à *framework*, sendo que nesta dissertação se recorreu a um programa de otimização de planos de produção. O plano de produção do utilizador é enviado do *FlexSim* para a aplicação e esta reencaminha o plano para o serviço externo, que depois devolve a sequência de produção otimizada à aplicação. Esta solução é enviada de volta para o *FlexSim* e é alterada a tabela com o plano melhorado.

V. CONCLUSÃO

O principal objetivo do projeto centrava-se no desenvolvimento de uma *framework* que explorasse as capacidades de *Digital Twins* de processos e que permitisse a sua aplicação a diferentes tipos de produção numa fábrica. Esta *framework* deveria ser capaz de implementar um DT dos processos de um *shop floor*, que permite a monitorização da fábrica em tempo real, a simulação de planos de produção e a comunicação com serviços inteligentes externos.

Em jeito de conclusão, os objetivos da dissertação foram cumpridos na sua totalidade, no que toca às três componentes desenvolvidas. Por sua vez, o resultado deste projeto poderá certamente ser uma base para trabalhos futuros no que diz respeito a *Digital Twins* de processos. A *framework* desenvolvida está aberta a ajustes e a novas funcionalidades, sendo um ponto de partida para a criação de uma estrutura funcional ao nível industrial.

Como trabalho futuro, seria interessante estender a arquitetura do *shop floor* a um protótipo constituído por sensores e atuadores. Este seria o ponto de partida para a integração da *framework* num sistema físico real mais tarde. Incluir distribuições de probabilidade no modelo, por exemplo para tempos de processamento, contribuiria para uma maior precisão na simulação. Finalmente, seria também relevante incorporar no motor de simulação recursos de transporte como empilhadores ou AGVs (*Automated Guided Vehicle*), encarregues de fazer o transporte das peças entre as WS.

AGRADECIMENTOS

Quero deixar o meu agradecimento ao meu orientador, Professor Doutor Jorge Pinho de Sousa, e ao meu co-orientador, Professor João Pedro Basto, pela oportunidade e por toda a ajuda. Agradeço também aos meus pais pela confiança em mim e à minha namorada pelo amor, pela amizade e por me acompanhar desde o início ao fim deste ciclo. Um obrigado a todos os que, de uma forma ou de outra, contribuíram para a concretização deste projeto.

REFERÊNCIAS

- [1] M. Grieves and J. Vickers, *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems*, 08 2017, pp. 85–113.
- [2] J. Lee, E. Lapira, B. Bagheri, and H.-A. Kao, “Recent advances and trends in predictive manufacturing systems in big data environment,” *Manufacturing Letters*, vol. 1, p. 38–41, 10 2013.
- [3] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, “Digital twin in manufacturing: A categorical literature review and classification,” *IFAC-PapersOnLine*, vol. 51, pp. 1016–1022, 01 2018.