

Model-Driven Development in Robotics Domain

a Systematic Literature Review

Tiago Heineck*, Enyo Gonçalves†, Aêda Sousa, Marcos Oliveira†, Jaelson Castro

Universidade Federal de Pernambuco (UFPE)

* Instituto Federal Catarinense (IFC)

† Universidade Federal do Ceará (UFC)

{th, ejtg, amcsb, jbc} @ cin.ufpe.br, {marcos.oliveira} @ ufc.br

Abstract—Robots are complex agents composed of various sensors and actuators that work together with software to meet specific requirements. The subset of robots that has the ability to interact among them and even with people, through gestures or speaking, is known as Social Robots. Model-Driven Development is a promising paradigm because it promotes the reuse of components and quick code generation with quality. Model-Driven Development has been widely used in the context of Robotics in order to reduce complexity, reduce development effort and reuse of software. Due to these facts, it becomes pertinent the development of a systematic literature review to compile these results. In this paper we investigate how MDD techniques have helped the field of Robotics, therefore a systematic literature review was conducted seeking to identify approaches and their main technical features, as well as the types of specific requirements, behavioral and social issues. We came to conclusion that the existing approaches provide many interesting capabilities, typically by using the component-based development paradigm seeking a higher level of software reuse and facilitating the implementation of systems.

Keywords—*model-driven development, robotics, software engineering, requirements engineering, systematic literature review*

I. INTRODUÇÃO

A robótica tem sido aplicada em diversos domínios nos últimos anos, como o industrial [1], saúde [2] e educação [3]. Além disso, empresas de todo o mundo estão fabricando robôs sociáveis para diversas finalidades, como: segurança, entretenimento e cuidados com a saúde [4].

O desenvolvimento de software na robótica tem sido um desafio, uma vez que os sistemas robóticos estão cada vez mais complexos, exigindo múltiplos sensores distribuídos no corpo do robô e assim incorporando tarefas mais complexas [5], como no caso de robôs sociais, que atuam em ambientes e contextos que exigem adaptação do sistema em tempo de execução. Também devem possuir características humanas como: expressar e perceber emoções, comunicação em alto nível de diálogo, aprender modelo de outros agentes, estabelecer relações sociais, exibir personalidade e caráter e aprender competências sociais [6] como cumprir normas.

Neste sentido, abordagens baseadas no desenvolvimento orientado a modelos (Model Driven Development - MDD) vem sendo propostas no contexto de robótica e apresentam-se como uma alternativa para mitigar a complexidade no desenvolvimento. Para Brambilla et al. [7], MDD é um

paradigma de desenvolvimento que usa modelos como artefatos primários do processo desenvolvimento, onde o código é gerado (semi) automaticamente a partir dos modelos.

No desenvolvimento de software é necessário que os requisitos de software sejam identificados e analisados para que possa ser especificada uma solução sem ambiguidade [8]. Uma das formas de se especificar requisitos é por meio de modelos [9], que são utilizados como ponto de partida para o MDD em diversas áreas [9] [10], inclusive na área de robótica [11] [12].

Devido ao grande número de propostas baseadas em MDD no domínio de robótica, identificou-se a importância de um melhor entendimento de como estas técnicas tem auxiliado o desenvolvimento de software neste domínio e de caracterizar como modelos são utilizados e os tipos de requisitos especificados, além de questões comportamentais e sociais que são usados nestas técnicas.

Assim, optamos pela realização de uma Revisão Sistemática da Literatura (RSL) com o objetivo de identificar como técnicas de MDD têm apoiado o domínio de robótica, verificando características que as técnicas possuem no apoio ao processo de desenvolvimento de software. Outras características de interesse são observadas, como os tipos de requisitos abordados classificando em funcionais e não-funcionais. Também analisamos a questão comportamental (reativo/deliberativo) auxiliando na adoção de um padrão e a existência de técnicas que atendam questões sociais para o apoio ao desenvolvimento de projetos envolvendo robôs sociais e robótica afetiva.

Diante dos dados levantados na RSL elaboramos um catálogo on-line “*Model-Driven Robot Repository*” [13] onde os estudos encontrados estão identificados de acordo com cada classificação realizada. Por outro lado, os resultados aqui sumarizados auxiliam no embasamento a respeito do assunto e tomada de decisão para o desenvolvimento de novas atividades de pesquisa, tendo como público interessados tanto no desenvolvimento quanto na aplicação de técnicas de MDD em robótica.

As próximas seções estão organizadas da seguinte maneira: na Seção II é apresentado o referencial teórico, a Seção III exhibe as questões metodológicas, os resultados são apresentados e discutidos na Seção IV, na Seção V constam as ameaças a validade do estudo, os trabalhos relacionados são

apresentados na Seção VI e finalmente na Seção VII as conclusões e trabalhos futuros.

II. REFERENCIAL TEÓRICO

A. Desenvolvimento Orientado a Modelos

Para Völter et. al. [14] os modelos são considerados artefatos de primeira ordem, utilizados como base para a geração automatizada de código, aumentando a velocidade de desenvolvimento, separação de conceitos e qualidade, entre outros fatores. MDD tem sido utilizado para diversas atividades, como linha de produtos de software (SPL) [15] [16] [17], desenvolvimento de jogos [18], sistemas embarcados [19], entre outras.

Modelos podem ser transformados em texto ou em outros modelos. Uma das técnicas mais utilizadas em MDD trata-se da transformação de modelos em texto, também conhecida como M2T (model-to-text), utilizada para implementação de código [20]. Biehl [21] explica que o M2T transforma elementos de um modelo em fragmentos de texto, caso o texto produzido seja código-fonte também é chamado de *model-to-code*. Já o tipo de transformação M2M (Model-to-model) cria elementos em um modelo alvo, mapeando elementos de um modelo-fonte.

B. Robótica e Robótica Social

Um robô pode ser uma máquina que executa tarefas repetitivas, sejam guiadas, predefinidas ou de maneira inteligente, capaz de perceber o ambiente e realizar a tomada de decisão [22]. De acordo com a Federação Internacional de Robótica [23] os robôs são classificados como: 1) Industrial: manipulador de múltiplo propósito, reprogramável, controlado automaticamente, programável em três ou mais eixos, ou 2) Serviço: mecanismos programáveis em dois ou mais eixos com um grau de autonomia, movendo no ambiente e realizando tarefas úteis para seres humanos, para uso pessoal ou profissional.

Um robô social é um dispositivo (semi) autônomo que interage e se comunica com humanos seguindo regras sociais ligadas ao seu papel [24]. Os robôs sociais ou socialmente interativos variam em forma e função, cada um com uma proposta, respeitando algumas normas sociais para aumento do desempenho no ambiente humano [6]. Kirby et al. [25] reforçam que para uma interação humano-robô seja mais suave é importante a adesão dos robôs em normas sociais.

III. METODOLOGIA

Esta RSL é baseada no guia de [26], muito utilizado por pesquisadores da engenharia de Software. Seguindo o roteiro, é recomendada a construção de um protocolo bem definido de pesquisa. Em [27] uma prévia foi apresentada, refinando o mesmo após sugestões indicadas por pesquisadores da área. A seguir apresentamos a caracterização do protocolo da RSL por meio das questões de pesquisa, critérios de busca, critérios de inclusão e exclusão, etapas de seleção de estudos e extração de dados.

A. Questões de Pesquisa

Esta RSL visa identificar **como técnicas de desenvolvimento orientado a modelos têm auxiliado o domínio da robótica**, buscando características que as mesmas possuem. Sendo assim está apoiada sob as questões de pesquisa definidas a seguir.

RQ1: Quais as principais características das técnicas de MDD que são utilizadas no apoio ao domínio de robótica?

Devido a amplitude das questões as características foram definidas previamente, identificadas em um conjunto inicial de 8 estudos em uma pesquisa *ad hoc* que motivaram a realização desta RSL, sendo elas:

- **Técnicas envolvidas:** as técnicas de engenharia de software envolvidas (CBD, OO, SOA, entre outras).
- **Benefícios propostos:** benefícios propostos ou buscados pelos autores com o uso da técnica.
- **Tipo de modelos:** classificação dos estudos em modelos comportamentais, estruturais e funcionais, como realizado em [28].
- **Contribuições relacionadas a MDD:** geração de código, meta-modelo, transformação de modelos, extensão de linguagem, entre outros.
- **Artefatos disponíveis:** diagramas, modelos, linguagem utilizados pelas abordagens.
- **Artefatos gerados:** classificação dos itens que são gerados pelas técnicas, como código-fonte (anotando a linguagem), modelos, arquivos específicos, entre outros.
- **Tipo de geração de código:** completa, parcial ou não mencionado.
- **Frameworks:** quando a abordagem mencionar frameworks específicos envolvidos o nome é anotado.

RQ2: Quais tipos de requisitos estão envolvidos nas abordagens?

Os requisitos de software são divididos em duas categorias: funcionais (Functional Requirements - FR) capturando a natureza da interação entre o componente e o ambiente, já os não-funcionais (Non-Functional Requirements - NFR) abordam os fatores adicionais para o atendimento de requisitos funcionais, como segurança e desempenho [29], sendo os estudos classificados desta forma.

RQ3: Quais os paradigmas/técnicas utilizados para questão comportamental do robô?

Em robótica comportamentos de robô podem ser considerados reativo e deliberativo [30], sendo atendidas por arquiteturas classificadas em **reativo** (sensor-atuador) quando o robô reage diretamente ao ambiente, **deliberativa ou hierárquica** (sensor-plano-atuador) quando o robô recebe entrada de sensores e passa por um plano correspondente a um modelo de mundo antes da saída ser dada por atuadores e **híbrido** (plano, sensor-atuador) que combina as características das duas anteriores [31].

RQ4: Quais e como são atendidas questões sociais?

Questões sociais estão relacionadas a percepção e interação do robô com as pessoas e ambientes sociais, a observação deste ponto é feita com base no *survey* de [32], onde menciona comportamentos sociais como coletividade (enxame), vida artificial, interação homem máquina, além de apresentar diversas outras como emoções, engajamento social, afetividade, entre outros. Neste sentido é extraída a questão social identificada e os artefatos utilizados para tal.

B. Critérios de Busca

Os critérios foram definidos optando por estudo publicados a partir de 2005 até a data da pesquisa (fevereiro de 2016), no idioma inglês e acessíveis pela rede do Centro de Informática da Universidade Federal de Pernambuco. Excluindo artigos com menos de 5 páginas. O ano de 2005 foi escolhido visando levantar os estudos mais recentes, concentrando nos esforços dos últimos 10 anos de pesquisas.

Partindo das questões foram identificadas palavras-chave para compor a *string* de busca apresentada na Tabela 1 e adaptada de acordo com cada base utilizada. Foram realizados pilotos de calibração a fim de cobrir resultados mais representativos para as perguntas do estudo. Como todas as questões e análises são feitas sob a interseção de robótica e MDD somente termos das duas áreas foram considerados e vinculados com o operador lógico “AND”.

TABELA 1. STRING DE BUSCA

Robô	Operador	MDD
robot*	AND	("model driven" OR "domain specific language" OR "domain specific modeling" OR metamodel OR "meta-model" OR "modeling language" OR "code generation" OR "generative programming" OR MDD OR MDA OR MDE OR DSL OR DSML)

As fontes de busca utilizadas foram ScienceDirect, IEEE Xplore, ACM Digital Library, Engineering Village e Web of Science, aplicando filtros para busca em título, resumo e palavras-chave.

C. Critérios de Inclusão e Exclusão

Os critérios de inclusão e exclusão foram definidos considerando as características já apresentadas na seção II e discutidos entre os pesquisadores, sendo os apresentados a seguir:

- 1) Estudos primários publicados a partir de 2005 e disponíveis de forma integral na rede da universidade, escritos em inglês e legíveis foram incluídos;
- 2) Artigos sobre técnicas de MDD específicas para robôs foram incluídos, abordagens para propósitos mais amplos que usam robô apenas como exemplo foram excluídos;

- 3) Textos que apresentam o uso de modelos no processo, mas não são artefatos primários/chaves foram excluídos;
- 4) Artigos sobre técnicas que consideram modelos como artefatos principais e tenham relação com geração de código-fonte, transformações ou interpretação de modelos (Models@RunTime) foram incluídos.

D. Etapas de seleção dos estudos

A figura 1 apresenta o processo de seleção dos estudos, sendo que na primeira etapa foram realizadas as buscas nas bases científicas, retornando 2140 estudos, seguida da eliminação automática de duplicados por meio da ferramenta StArt [33] e conferência manual comparando título, resumo, local de publicação e ano, restando 1039 estudos.

Na terceira etapa foi realizada a leitura de título, resumo e palavras-chave, selecionando 244 artigos, tendo as duplas analisado a relevância do conteúdo destes itens em relação a resposta das questões de pesquisa, bem como o enquadramento aos critérios de inclusão e exclusão. Como última etapa os mesmos foram lidos de forma integral, classificando 71 estudos que compõem a extração de dados desta revisão sistemática da literatura.

Em todas as etapas havendo dúvidas a respeito da inclusão a mesma foi considerada para uma análise mais profunda na próxima etapa.

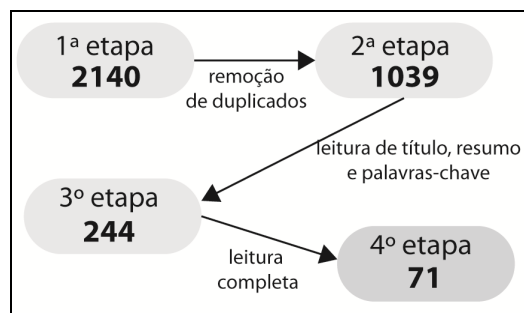


Fig. 1. Etapas da pesquisa

E. Extração de dados

Os dados foram extraídos em formulários pré-definidos contendo os itens referentes a publicação (ano, título, autores, local de publicação), os objetivos da pesquisa, local para as referências de possíveis estudos relacionados com as perguntas, além dos itens que compõem a formulação das respostas das questões de pesquisa, sendo eles: contribuição e benefícios, elementos, diagramas e artefatos, tipos de requisitos, tipos de questões sociais atendidas, paradigmas de engenharia de software envolvidos, classificação e questões em aberto. Todos os campos do formulário foram acompanhados de uma explicação e exemplos de preenchimento ainda na fase de planejamento.

F. Análise de Qualidade dos Estudos

Durante a fase de extração uma análise de qualidade adaptada de [34] foi realizada, não para fim de exclusão,

apenas para classificação. As questões foram pontuadas seguindo a escala de Likert [35], sendo “0” para não atende, “0,5” para atende parcialmente e “1” para “atende”, sendo elas:

- Existe uma motivação para a realização do estudo?
- Os objetivos da pesquisa estão claramente descritos?
- A abordagem ou técnica está claramente descrita?
- Apresenta meta-modelo ou descreve a linguagem?
- Os resultados são apresentados ou discutidos?
- É criada uma ferramenta/software ou algoritmo?
- A publicação apresenta exemplo ou caso de uso?
- As limitações são discutidas?
- Apresenta questões em aberto?

Na tabela 2 é apresentada a quantidade de estudos de cada pontuação, que reflete na soma do resultado das questões de qualidade, os demais são omitidos por questão de espaço, estando disponíveis em [13].

TABELA 2. QUANTIDADE DE ESTUDOS E PONTUAÇÕES

Pontuação	Quantidade de estudos
8,5 e 9	5
7, 7,5 e 8	19
6 e 6,5	21
5 e 5,5	14
4 e 4,5	10
Menor que 4	2

IV. RESULTADOS

Os resultados foram extraídos de 71 artigos selecionados para a etapa final, por legibilidade não são mencionados aqui, os mesmos estão disponíveis no repositório apresentado no item G desta seção, neste trabalho são apresentados de maneira sintetizada para cada questão de pesquisa, as evidências encontradas foram resumidas agrupando características e transformando em valores quantitativos, proporcionando uma visão geral dos resultados obtidos.

A. Visão Geral

Dos estudos analisados, 38 não restringem o tipo de robô, 13 tem o foco mais voltado para robôs de serviço, 11 em robôs industriais e 6 específicos para robôs móveis, 2 estudos envolvem futebol de robôs e 1 específicos para robôs modulares. Onde 52% dos estudos apresentam caso de uso com robô real dos mais variados tipos (ex. Lego, Pioneer 3DX, Robotino), 20% apresentam um exemplo, 15% não apresentam caso de uso ou exemplo e 13% apresentam exemplo em ambiente de simulação.

A Figura 2 apresenta a distribuição temporal dos estudos, sendo um dado importante para mensurar a evolução da quantidade de trabalhos com o passar do tempo. Identificamos que o ano de 2013 é o ano mais significativo com 16 estudos, seguido pelos anos de 2012 e 2014 com 11 estudos. Vale ressaltar que o ano de 2016 considera somente os meses de janeiro e fevereiro.

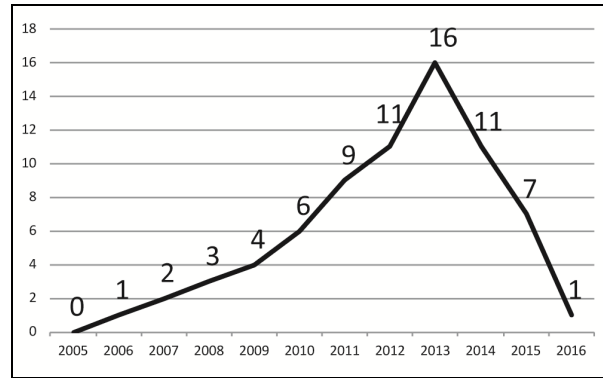


Fig. 2. Distribuição Temporal dos Estudos

A Tabela 3 apresenta uma lista das 4 fontes onde mais estudos foram extraídos. Este indicador mostra que mais artigos foram identificados no International Conference on Simulation, Modeling, and Programming for Autonomous Robot, seguido de IEEE International Conference on Intelligent Robots and Systems, MORSE/VAO Workshop on Model-Driven Robot Software Engineering and View-based Software-Engineering e International Conference on Intelligent Robots and Systems.

Os estudos mais citados são apresentados na Tabela 4, o número de citações foi extraído utilizando os dados do Google Scholar [36].

TABELA 3. LOCAL DE PUBLICAÇÃO

Local	Qtd de Estudos
International Conference on Simulation, Modeling, and Programming for Autonomous Robots	8
IEEE International Conference on Intelligent Robots and Systems	7
MORSE/VAO Workshop on Model-Driven Robot Software Engineering and View-based Software-Engineering	5
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)	3

TABELA 4. TRABALHOS MAIS CITADOS

Trabalho	Qtd de Citações
Robotic software systems: From code-driven to model-driven designs [37]	55
The BRICS component model: A model-based development paradigm for complex robotics software systems [5]	53
A new skill based robot programming language using UML/P Statecharts [38]	44
RobotML, a domain-specific language to design, simulate and deploy robotic applications [39]	44
Design abstraction and processes in robotics: From code-driven to model-driven engineering [40]	34

B. RQ1: Quais as principais características das técnicas?

Esta questão vai de encontro ao objetivo deste trabalho, verificando como as técnicas de MDD tem contribuído para a robótica.

Analisando as **técnicas/áreas** relacionadas nos estudos foram identificados 29 estudos que mencionam desenvolvimento baseado em componentes (CBD ou CBSE), seguido de 8 que mencionam orientação a objetos, 5 orientado a agentes e 3 menções de arquiteturas orientadas a serviços e padrões de projeto (design patterns), outras técnicas que aparecem com menor expressividade são por exemplo: programação imperativa, ontologia, engenharia baseada em conhecimento, rastreabilidade de requisitos, software orientado a eventos e linha de produto de software.

O desenvolvimento baseado em componentes tem sido buscado pela sua capacidade em promover o reuso de software em projetos de robótica, tendo algumas abordagens utilizado do conceito de componentes de prateleira, combinando os mesmos para implantação em diferentes plataformas robóticas. Este item vai ao encontro da análise que envolve os **benefícios propostos** pelos autores, onde 51% dos estudos mencionam reuso de software, enquanto 28% buscam redução de complexidade, 23% agilidade, 21% separação de conceitos ou de papéis, 13% qualidade, 6% manutenibilidade, além de outros conceitos que aparecem em menos de 4% dos estudos.

Seguindo a análise os próximos itens envolvem mais diretamente características das abordagens, o primeiro deles refere-se aos **tipos de modelos**, na Figura 3 são apresentados os resultados, com predominância de modelos estruturais (ex. diagrama de componentes e de classes) e de comportamento (ex. máquinas de estado, diagramas de atividade e de casos de uso), os modelos funcionais (ex. data flow, blocos funcionais) aparecem menos.

Relacionado ao tipo de modelos estão os artefatos analisados, onde na Figura 4 são apresentados os **artefatos disponíveis** que foram encontrados nos estudos, ficando claro que diagramas que representam estado são bastante utilizados pelas abordagens, seguido por diagrama de componentes, diagrama de atividades e linguagens específicas de domínio, os itens modelos específicos, linguagens específicas, outros diagramas e outros artefatos representam a somatória de itens específicos de com no máximo 2 registros cada. Uma prática encontrada é a utilização de artefatos mais específicos, que são criados como linguagens específicas ou extensões de linguagens para atendimento específico a características do domínio de robótica.

Quanto os **artefatos gerados**, a Figura 5 apresenta o resultado da análise, os itens identificados como “outros” correspondem a somatória dos artefatos com no máximo de 2 registros. Seguindo a análise é possível observar que a maioria dos artefatos gerados são em código C++, aparecendo 18 vezes, seguido pelas linguagens Java, C, linguagens específicas de domínio, Python e Ada. A grande quantidade de artefatos indicados como outros apresenta um comportamento de que normalmente as técnicas de MDD são utilizadas para atender necessidades mais específicas dos autores.

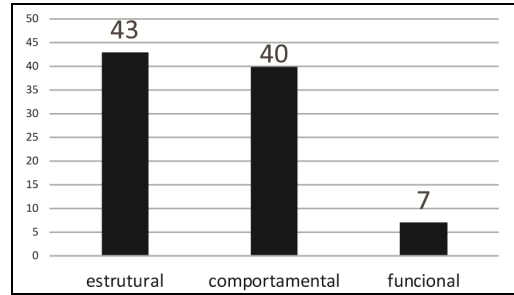


Fig. 3. Tipos de modelos

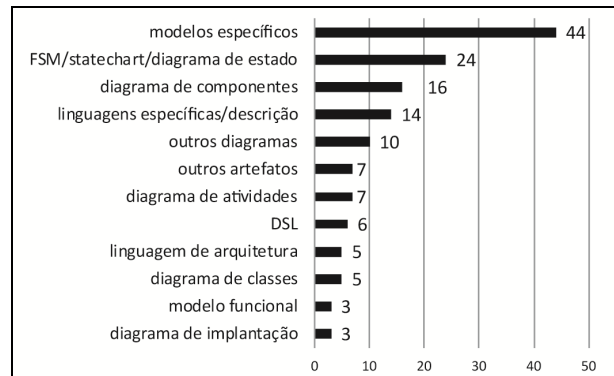


Fig. 4. Artefatos Disponíveis

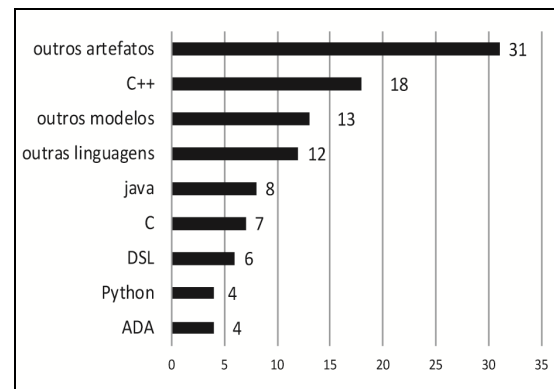


Fig. 5 Artefatos Gerados

Além dos artefatos gerados, algumas técnicas têm relação com **frameworks** específicos para o domínio de robótica, onde 49% dos estudos mencionam algum tipo, sendo 20% ROS [41], 14% Orocos [42], 7% SmartSOFT [43] e 8% outros que representaram menos de 2 registros cada. Indicando que além das técnicas de MDD os pesquisadores utilizam outras formas que auxiliam o processo de desenvolvimento, como a utilização de frameworks específicos para o domínio, sendo que algumas abordagens possuem middlewares e frameworks próprios, como o caso dos demais itens identificados.

Quanto a **Geração de código**, 15 estudos informam a geração total do código-fonte, o mesmo número menciona a geração parcial de código, neste caso são gerados esqueletos de código para preenchimento manual, os demais estudos não

deixam claro o nível de código gerado. Mesmo considerando a geração parcial de código a utilização de MDD fornece uma padronização que em nossa avaliação aumenta o entendimento das pessoas envolvidas no projeto.

Na Tabela 5 são apresentadas as **contribuições relacionadas a MDD**, observando a geração de código (model-to-text) predominante, seguido por meta-modelos, transformação de modelos (model-to-model), extensão de linguagem, nova linguagem, interpretação de modelos (modelos executáveis).

Os estudos que mencionam extensões de UML e não apresentam um meta-modelo, não foram contabilizados neste item.

TABELA 5. CONTRIBUIÇÕES RELACIONADAS A MDD

Tipo de contribuição	Quantidade
Geração de código	62
Meta-modelo	32
Transformação de modelos	24
Extensão de linguagem	21
Nova linguagem	11
Interpretação de modelos	5
Transformação de texto para modelos	2

C. RQ2: Quais tipos de requisitos estão envolvidos nas abordagens?

O objetivo desta questão é classificar os requisitos de software em funcionais, não-funcionais ou ambos, neste sentido consideramos requisitos funcionais quando a técnica atende características relacionadas as partes de software envolvidas com as finalidades do sistema e não-funcionais quanto atendem fatores adicionais tais como segurança e desempenho.

As técnicas em sua grande maioria atendem apenas requisitos funcionais, como apresentado na Tabela 6, somando 75% dos resultados, porém aproximadamente 22% atendem ambos e menos de 3% somente requisitos não-funcionais. Nas técnicas que atendem ambos os tipos, os requisitos não-funcionais são normalmente acrescentados por meio de atributos nas ferramentas, chamados de propriedades não-funcionais, relacionados a desempenho ou qualidade.

TABELA 6. TIPOS DE REQUISITOS

Tipo	Porcentagem
Funcionais	75%
Não-Funcionais	3%
Ambos	22%

Das abordagens específicas para requisitos não-funcionais foram identificadas, em [44] os autores usam uma abordagem guiada a modelos para construir e operar sistemas de auto otimização, focando no comportamento não-funcional da aplicação utilizando contratos de QoS (Quality of Service), segue o paradigma Models@run.time, que estende a aplicabilidade dos modelos produzidos para o ambiente de execução [45]. Em [11] os autores introduzem rastreabilidade

de requisitos de segurança (*Safety*) em aplicações robóticas por meio de MDD, com a finalidade de identificar, gerenciar e mitigar riscos de maneira automática.

D. RQ3: Quais os paradigmas/técnicas utilizados para questão comportamental do robô?

Neste item são identificadas as técnicas utilizadas para reorganização do robô, auxiliando a desenvolvedores de novas linguagens seguirem um caminho por uma delas ou optando pela neutralidade. Como questão comportamental são observadas técnicas utilizadas para organização da arquitetura interna do sistema, os resultados são apresentados na Tabela 7.

TABELA 7. TÉCNICAS PARA QUESTÃO COMPORTAMENTAL DE ROBÔ

Paradigma/Técnica	Porcentagem
Deliberativo/Hierárquico	5%
Híbrido	8%
Não menciona	64%
Reativo	15%
Teleo-reativo	1%

A maioria, 64% dos estudos não mencionam, não informam ou não atendem questões comportamentais ou reorganização de tarefas por meio de uma técnica específica. Como no caso de [46] um dos requisitos é que exista neutralidade, ou seja, não deve ser imposto um estilo arquitetural (reativo, deliberativo, híbrido). Este valor indica que a questão comportamental não necessariamente é uma preocupação das abordagens, sendo que tais padrões podem variar pelo uso de frameworks ou até mesmo pela opção de neutralidade, uma análise mais aprofundada neste ponto para um melhor entendimento é indicada.

Nos 46% dos estudos identificados que de alguma forma atendem este quesito, predomina abordagens reativas, seguidas de híbrido, deliberativo. Em [47] o paradigma teleo-reativo é utilizado, programas teleo-reativos são um conjunto de regras reativas que continuamente monitoram o ambiente e disparam ações para satisfação de objetivos.

E. RQ4: Quais e como são atendidas questões sociais?

O objetivo aqui é identificar aspectos sociais atendidos pelas técnicas de MDD para guiar no desenvolvimento de projetos e estudos de robôs sociais móveis. Contudo, poucos estudos foram identificados, indicando a possibilidade de uma lacuna a ser preenchida por futuras pesquisas. Alguns estudos apresentam características relacionadas a aspectos sociais, como coletividade, engajamento, enxame e regras de trabalho.

Em [48] os autores utilizam uma combinação de MDD com Modelagem Orientada a Objetos em tempo real e Algoritmos de Formação Triangular (TFA) para simular o comportamento social de enxame (swarm), um comportamento observado em insetos.

No estudo [49] são utilizadas máquinas de estado para geração de estratégias de jogo em times de robô, verificando posicionamento de outros competidores e do ambiente do campo. Na mesma linha de futebol de robôs, [50] apresenta

uma combinação de abordagens, baseada em Engenharia de Software Orientada a Agentes e MDE, mencionando um modelo de Protocolo de Interface de Agente que inclui regras de engajamento.

Em [51] diagramas de atividades são estendidos para modelagem de regras de trabalho relacionadas ao carregamento de peças em um ambiente de produção com múltiplos robôs cooperativos.

No estudo [52] MDD apoia o desenvolvimento de um sistema de visão computacional que auxilia a movimentação autônoma de cadeiras de pessoas em reabilitação.

F. Discussões

Por meio dos resultados alcançados com a RSL, verificamos que o domínio de robótica tem características próprias, relacionado com diversos campos de estudo, como inteligência artificial, sistemas embarcados, sistemas cyber-físicos, visão computacional, engenharia mecânica e elétrica, entre outras.

O desenvolvimento de robôs de serviços e sociais realmente envolvem muitas características que aumentam a complexidade para programadores e engenheiros, pois lidam com variabilidades e mudanças de requisitos em tempo de execução, não sendo uma tarefa fácil. Sendo assim, muitos dos estudos buscam MDD visando facilidade no desenvolvimento, deixando mais simples para não especialistas realizarem modificações.

Atendendo o objetivo principal deste trabalho, foi identificado que o desenvolvimento orientado a modelos tem auxiliado o domínio de robótica predominantemente com geração de código, utilizando em mesma proporção modelos estruturais e comportamentais, sendo diagramas de componentes e diagramas de estado os mais utilizados respectivamente em cada tipo. Além do reuso de software, componentes são uma alternativa interessante a redução de complexidade, pois permitem encapsular detalhes de implementação para usuários não especialistas do domínio de robótica.

Quanto a geração de código existe um equilíbrio entre técnicas que geram códigos completos ou parciais, mesmo gerando esqueletos de código entendemos que apoia uma padronização que pode facilitar o entendimento do código gerado. Sendo que o principal artefato gerado é código-fonte na linguagem C++.

No contexto de especificação de requisitos, poucos estudos apoiam requisitos não-funcionais, acreditamos que estes sendo mais explorados podem garantir uma melhor execução do robô no atendimento de suas tarefas.

Referente a questão comportamental do robô, uma análise aprofundada deve ser realizada nos estudos onde não são mencionados, isto deve ser feito entrando em contato com os autores. Mesmo assim os indicadores apontam que o paradigma reativo é mais abordado pelas técnicas, porém estamos convencidos de que a neutralidade destes estilos arquiteturais é o caminho mais interessante para uma abordagem de MDD em robótica.

Muitas questões sociais, como normas, afetividade e emoções, entre outras, não são tratadas especificamente pelas abordagens encontradas neste estudo. Portanto, um estudo empírico poderia ser realizado para verificar se uma das abordagens atenderia estes pontos sem adaptações ou então identificar as adaptações necessárias, podendo caracterizar uma questão em aberto.

Este trabalho serve como base para a realização de novos estudos, agrupando trabalhos relevantes relacionados a aplicação de técnicas de MDD no domínio de robótica, facilitando assim para pesquisadores interessados em novas atividades de pesquisa em encontrar trabalhos relacionados ao tema, bem como resumindo dados que podem ser utilizados para tomada de decisão no sentido de escolha de modelos a serem utilizados, tipos de artefatos, decisão sobre padrões arquiteturais ou até mesmo na escolha de uma abordagem para aplicação em projetos de robótica.

Quanto a metodologia utilizada, a revisão sistemática prova ser um método exaustivo, porém é uma excelente ferramenta para identificação e sumarização de dados sobre determinado tema.

G. Catálogo On-line

Os resultados com mais detalhes e identificação dos estudos selecionados, além dos artefatos utilizados no processo estão disponíveis no nosso catálogo on-line em <http://mdr.videira.ifc.edu.br>.

Uma ferramenta web foi desenvolvida para incrementar o catálogo facilmente, as principais funcionalidades são: (1) usuários podem cadastrar novos estudos, (2) novas perguntas ou análises podem ser adicionados e (3) qualquer usuário pode reportar inconsistências ou sugerir melhorias nas informações de cada estudo.

O objetivo principal do catálogo é reunir em um local os estudos existentes, facilitando para pesquisadores ou usuários novatos que tenham interesse no assunto, da mesma forma os usuários podem contribuir adicionando informações sobre atualizações ou melhorias nas técnicas apresentadas.

V. AMEAÇAS A VALIDADE

Uma das ameaças em uma revisão sistemática da literatura é a possibilidade de viés de pesquisa, neste sentido, todas as etapas foram realizadas em duplas, havendo incompatibilidade de resultados um terceiro pesquisador deu seu voto de desempate e em alguns casos foram realizadas discussões.

Alguns estudos podem ter sido excluídos da pesquisa por questões subjetivas, outro fato que pode ter ocasionado a não inclusão de alguns estudos foi a montagem da *string* de busca, alguns termos relacionados a MDD poderiam ter sido melhor explorados, como *model-oriented*, *model-based*, *transformation* e *interpretation*. Sendo assim, toda a documentação foi realizada tomando o cuidado de garantir a possibilidade de extensão deste estudo a partir dos mesmos procedimentos e recursos utilizados, sendo de interesse dos autores a continuidade e ampliação do catálogo on-line.

Para minimizar riscos os pesquisadores mais experientes do grupo revisaram todo o protocolo apontando melhorias. Em

todas as etapas foram realizados pilotos verificando a consistência das etapas seguintes, além de um piloto geral do protocolo, havendo necessidade de ajustes foram realizados de imediato, os objetivos do estudo e as questões de busca permaneceram inalterados durante toda a condução do processo.

VI. TRABALHOS RELACIONADOS

Trabalhos que envolvem MDD e robótica já foram realizados por outros pesquisadores, tendo um foco diferente deste.

Pons et al. [53] realizaram uma revisão sistemática buscando responder se técnicas de MDD, CBD (Desenvolvimento baseado em componentes) e SOA (Arquitetura orientada a serviço) estavam sendo aplicadas e qual a tendência, além de verificar a interseção entre as mesmas e quais técnicas de MDE tem sido aplicadas e seu nível de automação. A busca incluiu artigos entre 1999 até o início de 2011 considerando as bases IEEE e ACM, sendo 22 artigos relacionados como MDD tendo o ano de 2010 como o maior com 7 registros, identificando 12 estudos que também envolvem CBD e apenas 2 também envolvendo SOA, corroborando com nossos resultados, onde CBD também é identificado como a área mais relacionada, com 29 estudos e SOA mantendo uma menor expressividade, com 3 menções. Os autores também apontam que diagramas UML são utilizados na maioria, apontando a menção em 64% dos estudos, a soma deste tipo de artefato encontrado em nosso estudo reforça a importância, tendo menção em 69% dos estudos. Os autores indicam também que 65% utilizam ferramentas já existentes, sendo que os outros 35% apontam para a criação de ferramenta, porém o levantamento deste item não é o foco de nosso estudo.

Na dissertação de Cattivera et al. [54] um mapeamento sistemático da literatura é realizado com foco em engenharia orientada a modelos (MDE) e robôs móveis, as tendências de publicação são apresentadas considerando o período entre 2004 até março de 2015, além disso os estudos são classificados por tipo de robô móvel (aéreo, aquático, terrestre), a maneira como apoiam (DSL, transformação ou geradores), aspectos modelados (navegação, movimento, otimização, validação) e tipo de ferramenta (baseado ou não em eclipse). Os autores recomendam a realização de uma revisão sistemática da literatura para aumento do entendimento das técnicas e outros itens que não são identificados, a análise aponta que 94,20% dos estudos mencionam algum tipo de transformação, sendo mais geral que nosso estudo, onde apontamos o tipo de transformação realizada.

Ambos os estudos apresentam um ponto de vista com foco em detalhes mais relacionados a MDD, nosso estudo difere de ambos, com foco no apoio ao desenvolvimento de software robótico, analisando os tipos de artefatos disponíveis (diagramas e modelos), tipos de artefatos gerados (linguagem de programação, arquivos de configuração), tipos de modelo, especificação de software e uma análise em aspectos comportamentais e sociais. Mesmo havendo diferenças nos

itens analisados e a maneira a qual foram conduzidos, todos contribuem para um melhor embasamento a respeito do uso de técnicas de MDD em robótica.

Revisões sistemáticas envolvendo MDD em áreas correlatas também foram realizadas. Em [19] os autores investigam a relação com sistemas embarcados, em [55] são apresentadas aplicações em conceitos de segurança, em [28] os autores identificam técnicas de engenharia de requisitos utilizados em MDD. No entanto o foco principal destas não se trata de robótica.

VII. CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho uma revisão sistemática da literatura foi realizada buscando evidências na literatura sobre técnicas de desenvolvimento orientado a modelos que tem auxiliado o domínio da robótica, foram selecionados 71 artigos de 2140 iniciais, sendo avaliados sob o ponto de vista de desenvolvimento de software em robótica com o uso de MDD. Este trabalho apontou para dados extraídos dos estudos selecionados, buscando sumarizar os esforços das pesquisas realizadas nos últimos 10 anos.

Os resultados apontam que técnicas de MDD tem sido propostas para o domínio de robótica, sendo as principais características encontradas as seguintes: reuso de software por meio do desenvolvimento baseado em componentes, agilidade e qualidade de código apoiados pelos mecanismos de geração de código-fonte e os artefatos disponíveis apoiam na maioria dos casos a especificação de requisitos funcionais, tendo apenas 2 abordagens que apoiam especificamente requisitos não-funcionais em qualidade e *safety* e quando atendem ambos os tipos os requisitos não-funcionais são apoiados por parâmetros em ferramentas e considerados propriedades não-funcionais.

Portanto, é possível concluir que existe um grande esforço por parte dos pesquisadores de MDD no desenvolvimento e aplicação de técnicas desta área no domínio de robótica, mesmo assim existem campos a serem explorados, pelos artefatos identificados é possível observar uma ligação mais próxima a implementação, existindo espaço para níveis mais altos de abstração, bem como o apoio a requisitos não-funcionais podem ser mais explorados, inclusive para lidar com requisitos emocionais que acreditamos ser uma alternativa no apoio a questões de afetividade em robôs sociais.

Neste sentido, poucos estudos apresentam o atendimento a questões sociais, porém acreditamos que as técnicas podem ser utilizadas em projetos envolvendo robôs sociais, podendo fazer uso justamente de componentes para o tratamento de muitas delas. Este ponto abre possibilidade de novos estudos buscando identificar empiricamente a utilização de MDD em robôs socialmente interativos.

Quanto a questão comportamental do robô a maioria dos estudos não menciona ou não restringe a um paradigma ou técnica, quando apresentado, predomina o estilo arquitetural reativo. Contudo, durante a realização desta análise e em nossas discussões concluímos que a neutralidade destes estilos seja uma estratégia a ser considerada, deixando aberta para as necessidades de cada projeto.

Futuramente é importante a realização de *snowballing* com a finalidade de complementar nosso catálogo, seguindo o guia de [56] e utilizando os estudos selecionados para etapa de extração como base. Algumas informações que não estão explícitas nos estudos serão melhores investigadas entrando em contato com os autores, outras análises ainda podem ser realizadas utilizando o conjunto de artigos catalogados, agregando novas informações e conclusões.

Da mesma forma, é indicada a realização de estudos empíricos envolvendo algumas das abordagens aplicando no contexto de robôs sociais, isto permitirá uma verificação mais completa do apoio no desenvolvimento de robôs sociais ou da necessidade de adaptações para o atendimento aos mesmos. Outros estudos são igualmente importantes para verificar o quanto as abordagens impactam realmente no tempo de desenvolvimento, redução de custo, facilidade de uso e reuso de software, assim uma avaliação empírica baseada em critérios específicos podem concluir a efetividade da utilização de MDD neste domínio.

Um ponto fundamental é a divulgação do catálogo on-line tanto para a comunidade de MDD quanto para a comunidade de robótica, assim sendo, tem sido divulgado por e-mail para pesquisadores interessados no assunto.

AGRADECIMENTOS

Agradecemos ao Instituto Federal Catarinense pela hospedagem do catálogo on-line.

Esta pesquisa foi apoiada pela FACEPE (Fundação de Amparo à Ciência e Tecnologia do Estado do Pernambuco) e pelo CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico).

REFERÊNCIAS

- [1] T. Brogårdh, "Present and future robot control development-An industrial perspective," *Annu. Rev. Control*, vol. 31, no. 1, pp. 69–79, 2007.
- [2] R. Bogue, "Robots in healthcare," *Ind. Robot An Int. J.*, vol. 38, pp. 218–223, 2011.
- [3] J. H. J. Han, M. J. M. Jo, S. P. S. Park, and S. K. S. Kim, "The educational use of home robots for children," *Rom. 2005. IEEE Int. Work. Robot Hum. Interact. Commun. 2005.*, pp. 378–383, 2005.
- [4] C. D. Kidd, W. Taggart, and S. Turkle, "A sociable robot to encourage social interaction among the elderly," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2006, no. May, pp. 3972–3976, 2006.
- [5] H. Bruyninckx, M. Klotzbücher, N. Hochgeschwender, G. Kraetzschmar, L. Gherardi, and D. Brugali, "The BRICS component model: a model-based development paradigm for complex robotics software systems," *Proc. 28th Annu. ACM Symp. Appl. Comput.*, pp. 1758–1764, 2013.
- [6] K. Fong, T., Nourbakhsh, I., & Dautenhahn, "A Survey of Socially Interactive Robots : Concepts , Design , and Applications Terrence Fong , Illah Nourbakhsh , and Kerstin Dautenhahn," *Rob. Auton. Syst.*, vol. 42, no. 3–4, pp. 143–166, 2003.
- [7] M. Brambilla, J. Cabot, and M. Wimmer, "Model-Driven Software Engineering in Practice." Morgan & Claypool Publishers, 21-Sep-2012.
- [8] R. S. Pressman, *Engenharia de Software*, 7th ed. Porto Alegre: Bookman, 2011.
- [9] M. Dos Santos Soares and J. Vrancken, "Requirements specification and modeling through SysML," *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, no. October, pp. 1735–1740, 2007.
- [10] D. Ameller, "Considering Non-Functional Requirements in Model-Driven Engineering," *2010 18th IEEE Int. Requir. Eng. Conf.*, pp. 189–198, 2009.
- [11] P. Sánchez, D. Alonso, F. Rosique, B. Álvarez, and J. A. Pastor, "Introducing safety requirements traceability support in model-driven development of robotic applications," *IEEE Trans. Comput.*, vol. 60, no. 8, pp. 1059–1071, 2011.
- [12] N. Yakymets, S. Dhouib, H. Jaber, and A. Lanusse, "Model-driven safety assessment of robotic systems," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 1137–1142, 2013.
- [13] T. Heineck, E. J. T. Gonçalves, A. Sousa, M. Oliveira, and J. Castro, "Model Driven Robots Repository," 2016. [Online]. Available: <http://mdrr.videira.ifc.edu.br/>.
- [14] S. Völter, Markus and Stahl, Thomas and Bettin, Jorn and Haase, Arno and Helsen, *Model-driven software development: technology, engineering, management*. John Wiley & Sons, 2013.
- [15] A. L. Medeiros, E. Cavalcante, T. Batista, and E. Silva, "ArchSPL-MDD: An ADL-Based Model-Driven Strategy for Automatic Variability Management," in *2015 IX Brazilian Symposium on Components, Architectures and Reuse Software*, 2015, pp. 120–129.
- [16] A. P. Magalhaes, J. M. N. David, R. S. P. Maciel, B. C. Silva, and F. A. da Silva, "Modden: An Integrated Approach for Model Driven Development and Software Product Line Processes," in *2011 Fifth Brazilian Symposium on Software Components, Architectures and Reuse*, 2011, pp. 21–30.
- [17] A. S. Nascimento, C. M. F. Rubira, R. Burrows, and F. Castor, "A Model-Driven Infrastructure for Developing Product Line Architectures Using CVL," in *2013 VII Brazilian Symposium on Software Components, Architectures and Reuse*, 2013, pp. 119–128.
- [18] E. F. do Prado and D. Lucredio, "A Flexible Model-Driven Game Development Approach," in *2015 IX Brazilian Symposium on Components, Architectures and Reuse Software*, 2015, pp. 130–139.
- [19] P. G. G. Queiroz and R. T. V. Braga, "Development of critical embedded systems using model-driven and product lines techniques: A systematic review," *Proc. - 2014 8th Brazilian Symp. Softw. Components, Archit. Reuse, SBCARS 2014*, pp. 74–83, 2014.
- [20] L. M. Rose, N. Matragkas, D. S. Kolovos, and R. F. Paige, "A feature model for model-to-text transformation languages," *Model. Softw. Eng. (MISE), 2012 ICSE Work.*, pp. 57–63, 2012.
- [21] M. Biehl, "Literature study on model transformations," *R. Inst. Technol. Tech. Rep. ISRN/KTH/MMK*, no. July, pp. 1–28, 2010.
- [22] R. A. F. Romero, E. Prestes, F. S. Osório, and D. Wolf, *Robótica Móvel*, 1st ed. Rio de Janeiro: LTC, 2014.
- [23] I. F. of Robotics, "Service Robots," 2016. [Online]. Available: <http://www.ifr.org/service-robots/>. [Accessed: 02-Apr-2016].
- [24] J. Kahn, P.H., N. G. Freier, B. Friedman, R. L. Severson, and E. N.

- Feldman, "Social and moral relationships with robotic others?," *RO-MAN 2004. 13th IEEE Int. Work. Robot Hum. Interact. Commun.*, pp. 545–550, 2004.
- [25] R. Kirby, J. Forlizzi, and R. Simmons, "Affective social robots," *Rob. Auton. Syst.*, vol. 58, no. 3, pp. 322–332, 2010.
- [26] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," *Engineering*, 2007.
- [27] T. Heineck, J. Castro, and E. J. T. Gonçalves, "Técnicas de Desenvolvimento Orientado a Modelos no Domínio de Robótica e Robôs Sociais," in *WTDSI - SBSI 02016*, 2016.
- [28] G. Loniewski, E. Insfran, and S. Abrahão, "A Systematic Review of the Use of Requirements Engineering Techniques in Model-Driven Development," *MoDELS*, 2010.
- [29] G. C. Roman, "Taxonomy of Current Issues in Requirements Engineering," *Computer (Long Beach, Calif.)*, vol. 18, no. 4, pp. 14 – 23, 1985.
- [30] R. G. Simmons, "Structured control for autonomous robots," *IEEE Trans. Robot. Autom.*, vol. 10, no. 1, pp. 34–43, 1994.
- [31] R. R. Murphy, *Introduction to AI robotics*. MIT press, 2000.
- [32] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Rob. Auton. Syst.*, vol. 42, no. 3–4, pp. 143–166, 2003.
- [33] E. Hernandez, "Using GQM and TAM to evaluate StArt-a tool that supports Systematic Review," *Clei Electron. J.*, vol. 15, no. 1, p. 13, 2012.
- [34] D. Dermeval, J. Vilela, I. I. Bittencourt, J. Castro, S. Isotani, P. Brito, and A. Silva, "Applications of ontologies in requirements engineering: a systematic review of the literature," *Requir. Eng.*, 2015.
- [35] J. McIver and E. G. Carmines, *Unidimensional scaling*, no. 24. Sage, 1981.
- [36] Google, "Google Scholar," 2016. [Online]. Available: scholar.google.com.br. [Accessed: 06-Jul-2016].
- [37] C. Schlegel, A. Steck, and A. Lotz, "Robotic Software Systems: From Code-Driven to Model-Driven Software Development," 2009.
- [38] U. Thomas, G. Hirzinger, B. Rumpe, C. Schulze, and A. Wortmann, "A new skill based robot programming language using UML/P Statecharts," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 461–466, 2013.
- [39] S. Dhoui, S. Kchir, S. Stinckwich, T. Ziadi, M. Ziane, and M. Z. Robotml, "RobotML, a Domain-Specific Language to Design, Simulate and Deploy Robotic Applications," *Int. Conf. Simulation, Model. Program. Auton. Robot.*, 2012.
- [40] C. Schlegel, A. Steck, D. Brugali, and A. Knoll, "Design abstraction and processes in robotics: From code-driven to model-driven engineering," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6472 LNAI, pp. 324–335, 2010.
- [41] O. S. R. Foundation, "ROS," 2016. [Online]. Available: <http://www.ros.org/>.
- [42] "OROCOS." [Online]. Available: <http://www.oroocos.org/>.
- [43] University of Applied Sciences, "SmartSoft." [Online]. Available: <http://smart-robotics.sourceforge.net/>.
- [44] C. Piechnick, R. Schöne, S. Götz, and U. Assmann, "Model-driven Multi-Quality Auto-Tuning of Robotic Applications," *Proc. 2015 Jt. MORSE/VAO Work. Model. Robot Softw. Eng. View-based Software-Engineering*, 2015.
- [45] G. Blair, N. Bencomo, and R. B. France, "Models@ run.time," *Computer (Long Beach, Calif.)*, vol. 42, no. 10, pp. 22–27, 2009.
- [46] S. Dhoui, S. Kchir, S. Stinckwich, T. Ziadi, M. Ziane, and M. Z. Robotml, "RobotML, a Domain-Specific Language to Design, Simulate and Deploy Robotic Applications," *Simulation, Model. Program. Auton. Robot. Third Int. Conf.*, 2014.
- [47] P. Sánchez, D. Alonso, J. M. Morales, and P. J. Navarro, "From Teleo-Reactive specifications to architectural components: A model-driven approach," *J. Syst. Softw.*, vol. 85, no. 11, pp. 2504–2518, 2012.
- [48] D. Ouellet, S. N. Givigi, and A. J. Beaulieu, "Control of swarms of autonomous robots using Model Driven Development - A state-based approach," *Syst. Conf. (SysCon), 2011 IEEE Int.*, pp. 512–519, 2011.
- [49] R. a. C. Rabelo, H. T. Macedo, E. O. Freire, and R. M. Oliveira, "Automatic code generation of SIMUROSOT game strategies," *ACM SIGSOFT Softw. Eng. Notes*, vol. 38, no. 2, p. 1, 2013.
- [50] A. Paraschos, N. I. Spanoudakis, and M. G. Lagoudakis, "Model-driven behavior specification for robotic teams," *Proc. 11th Int. Conf. Auton. Agents Multiagent Syst. (AAMAS 2012)*, vol. 1, pp. 171–178, 2012.
- [51] E. Koycheva, A. Braune, and S. Hennig, "Optimization of Production in an autonomous Robot served Manufacturing Facility," *2010 5th IEEE Int. Conf. Intell. Syst.*, 2010.
- [52] S. M. Grigorescu, O. Prenzel, and A. Gräser, "Model driven developed machine vision system for service robotics," *Proc. Int. Conf. Optim. Electr. Electron. Equipment, OPTIM*, pp. 877–883, 2010.
- [53] C. Pons, R. Giandini, and G. Arévalo, "A systematic review of applying modern software engineering techniques to developing robotic systems," *Ing. e Investig.*, vol. 32, no. 1, pp. 58–63, 2012.
- [54] G. Cattivera and G. L. Casalaro, "Model-Driven Engineering for Mobile Robot Systems: A Systematic Mapping Study," Malardalen University, 2015.
- [55] P. H. Nguyen, J. Klein, Y. Le Traon, and M. E. Kramer, "A systematic review of model-driven security," *Proc. - Asia-Pacific Softw. Eng. Conf. APSEC*, vol. 1, pp. 432–441, 2013.
- [56] C. Wohlin, "Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering," *18th Int. Conf. Eval. Assess. Softw. Eng. (EASE 2014)*, pp. 1–10, 2014.