

Relato de experiência da linguagem de modelagem SysADL para documentação de sistemas embarcados com Arduino ^{*}

Sara Guimarães, Marcia Lucena e
Roberta Coelho

Departamento de Informática e Matemática Aplicada (DIMAP)
Universidade Federal do Rio Grande do Norte, Natal, Rio Grande do Norte, Brasil
sguimaraaes@gmail.com, {marciaj, roberta}@dimap.ufrn.br

Resumo O presente artigo discute como foi realizada a documentação de um sistema embarcado com a placa Arduino. Os sistemas embarcados são caracterizados por realizarem a automação de atividades, que muitas vezes se utilizam de elementos do tipo sensor e atuador (e.g., irrigação, monitoramento e exploração espacial). Diante da relevância desses sistemas, a utilização técnicas adequadas de documentação torna-se premente. Neste artigo apresentamos um sistema embarcado utilizado em uma base de lançamento de foguete PET para automatizar o acionamento do lançamento e a inserção de combustível no foguete. Neste artigo apresentamos como este sistema foi modelado utilizando a visão estrutural, a visão comportamental e o diagrama de requisitos da SysADL, e discutimos sobre vantagens e desvantagens observadas com a modelagem adotada.

Palavras-chaves Sistemas embarcados · requisitos · Arduino · SysADL.

1 Introdução

Sistemas embarcados são caracterizados por terem capacidade de processamento de dados e serem de dedicação exclusiva para uma tarefa. Essas tarefas podem variar desde o controle do estado de uma luz (ligada ou apagada) com base na luminosidade do ambiente, um sistema de alerta de temperatura de um fábrica ou um sistema robótico para cirurgias médicas. A alta variedade de componentes associado às atividades rotineiras que podem ser automatizadas resulta em aplicabilidade cada vez maior para os sistemas de automação. Para garantir que esses sistemas sejam de fácil compreensão e manutenibilidade é necessário aplicar técnicas de documentação de requisitos que se adéquem às suas características.

Neste artigo é discutido como a linguagem de modelagem SysADL pode ser utilizada para documentar um sistema embarcado com Arduino e ele está organizado de modo a discutir os trabalhos relacionados na seção 2, seguindo

^{*} Apoiado pela Universidade Federal do Rio Grande do Norte (UFRN).

na seção 3 com as características do sistema escolhido para a documentação. Na seção 4 a documentação com o SysADL para o sistema da seção 3. Já na seção 5 são apresentadas as discussões sobre as normas do SysADL para o estudo de caso. Por fim, na seção 6 são apresentadas as considerações finais do trabalho.

2 Trabalhos relacionados

Em [6] é feita uma revisão das técnicas de elicitação e de especificação de requisitos para sistemas embarcados. Na elicitação apresentam o guia GERSE composto por duas fases para a elaboração de sete artefatos de modo a conduzir para elaboração de requisitos técnicos do sistema. Na elicitação [6] é apresentado o *template* TERASE para requisitos não funcionais. Esse *template* trabalha em aspectos do ambiente físico como temperatura, pressão e velocidade, além de aspectos dos dispositivos físicos como sensores, atuadores e componentes eletrônicos de prototipagem. Entretanto, sistemas embarcados são compostos tanto de requisitos não funcionais como de requisitos funcionais. No presente trabalho mostramos como a linguagem de modelagem SysADL pode ser utilizada para documentar requisitos funcionais e não funcionais de um sistema de automação com a placa Arduino.

[1] trás estatísticas sobre o uso de UML e *Model-Driven* para sistemas embarcados e mostra que aspectos significativos são melhorados em produtividade e qualidade de acordo com os participantes da pesquisa.

Em [3] é descrito um estudo de caso utilizando a linguagem de modelagem UML para o cenário de sistemas embarcados. O grande diferencial apresentado neste artigo consiste na representação de diagramas de classes que também representam componentes de testes. Todavia não é apresentado como os requisitos funcionais e não funcionais de um sistema embarcado estão relacionados com as classes (como conseguimos expressar utilizando a linguagem SysADL no presente trabalho).

Em [5] é discutido um sistema de IoT de um *Smart Place* utilizando o computador de placa única *Raspberry Pi* capaz de controlar automaticamente a temperatura de uma sala de modo a contribuir para a economia de energia e para o conforto das pessoas. Neste trabalho foi realizada uma avaliação experimental com cinco alunos de graduação. Os autores identificaram que a neutralidade dos participantes para a facilidade em utilizar a ferramenta pode ser relacionado com a relativa complexidade na compreensão dos elementos, relações e regras do estilo, porém, os participantes consideraram a arquitetura com impacto positivo para IoT. Além disso, utilizar o SysADL ainda foi motivado por não haver nenhum trabalho em [7] relatando a utilização desta linguagem com a placa Arduino.

3 Base de lançamento de foguete PET automatizada

Sistemas de embarcados possuem um componente base que é o microcontrolador responsável por controlar todos os demais componentes que serão conec-

tados a ele via prototipagem eletrônica. Esses sistemas devem focar na interação – através de sensores e atuadores – com o meio ambiente ao qual o sistema é inserido e essa interação é feita de modo exclusivo. Além disso, devem ser confiáveis, eficientes, dedicados à aplicação, atuar com restrições de tempo real e ser composto de componentes analógico e/ou digitais [2].

Neste trabalho será utilizado como estudo de caso um sistema embarcado utilizando a placa de prototipagem Arduino UNO que possui o microcontrolador Atmega 328p. Esse sistema é dedicado para uma base de lançamento de foguete PET com controle para o lançamento do foguete e o preenchimento do reservatório de combustível [4].

No artigo de [4] foram utilizadas as representações da Figura 1 para o sistema desenvolvido. Na Figura 1 (a) temos a montagem eletrônica do sistema mostrando como a pinagem utilizada para interligar os componentes na visão esquemática; e na Figura 1 (b) temos a visão em fluxograma descrevendo o comportamento do sistema com base em blocos de decisões e de ações.

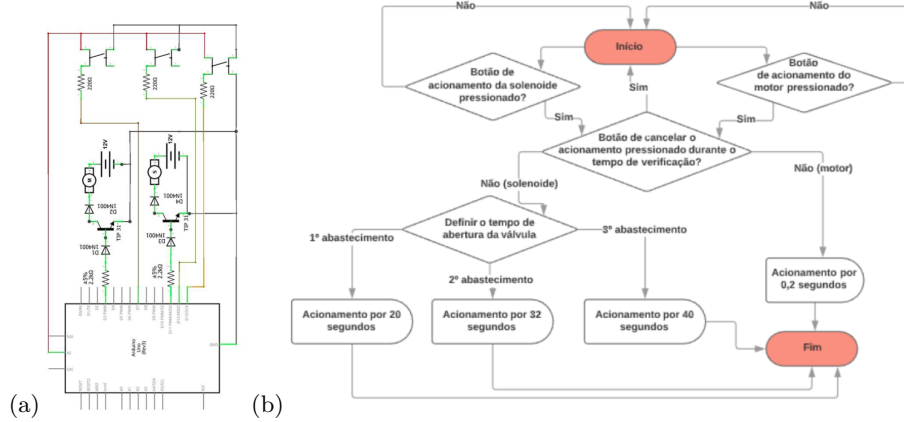


Fig. 1. (a) visão esquemática do circuito; (b) fluxograma da lógica de funcionamento

4 SysADL para documentação do estudo de caso

Segundo [7], a SysADL consiste em uma linguagem de modelagem, com foco na arquitetura de *software*, para documentação de sistemas embarcados. Neste trabalho foram elaboradas a visão estrutural, a visão comportamental, e o diagrama de requisitos seguindo a SysADL.

A visão estrutural vai utilizar pacotes para componentes, portas, conectores e tipos de dados. A configuração desses pacotes através do componente central SistemaBaseTorque é apresentada na Figura 2 e define a visão estrutural do SysADL, chamada de Diagrama de Definição de Bloco Interno (do inglês *Internal Block Definition* (IBD)). Cada um dos elementos é descrito a seguir:

- **Válvula solenoide:** componente para liberar a água e abastecer o foguete da base de lançamento. Possui uma porta, *ValvulaStatusIPT*, de entrada para especificar por meio de um comando o status (*open* ou *close*) da solenoide.

- **Motor:** componente para liberar a trava do foguete e permitir o seu lançamento. Possui uma porta, *MotorStatusIPT*, de entrada para especificar o status (*on* ou *off*) do motor.
- **Botão acionar válvula:** componente para permitir que o usuário acione a abertura da solenoide. Possui uma porta, *BotaoValvulaOPT*, para detectar o status (*on* ou *off*) do botão.
- **Botão acionar motor:** componente para permitir que o usuário acione a rotação do motor. Possui uma porta, *BotaoMotorOPT*, para detectar o status (*on* ou *off*) do botão.
- **Botão cancelar acionamento:** componente para permitir que o usuário cancela acionamentos solicitados na solenoide/motor. Possui uma porta, *BotaoCancelarOPT*, para detectar o status (*on* ou *off*) do botão.
- **Controlador acionamento:** componente para controlar os acionamentos com base na interação com o usuário. Possui portas, *BotaoMotorIPT* e *BotaoValvulaIPT*, para detectar qual componente foi acionado, uma porta, *BotaoCancelarIPT*, para detectar se o cancelamento do acionamento foi solicitado, duas, *ValvulaStatusOPT* e *MotorStatusOPT*, para especificar o status dos componentes a serem acionados e mais quatro portas, *AberturaValvulaOPT*, *AberturaValvulaIPT*, *AtivarMotorOPT* e *AtivarMotorIPT*, para especificar o tempo de acionamento dos componentes de válvula e solenoide.

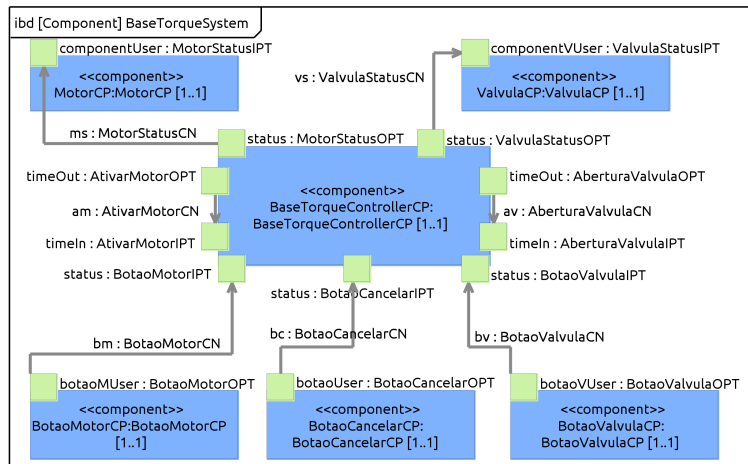


Fig. 2. Diagrama IBD dos componentes do sistema embarcado

A visão comportamental demonstra como o sistema consegue exercer funcionalidades por meio de atividades e interações. As atividades são definidas com a especificação das ações, dos parâmetros de dados aceitos e de como os elementos podem se relacionar. Este é chamado de diagrama de definição de bloco (do inglês *Block Definition Diagram* (BDD)) ilustrado na Figura 3.

Na Figura 3 é possível observar três atividades: (i) a atividade de controle de acionamento que é composta pelas ações de acionar o motor e de acionar a válvula, em ambas ações existem equações que formulam a lógica de programação para o controle correto de decidir qual componente será acionado e

verificar possíveis solicitações para cancelar o acionamento; e (ii) as atividades de abertura da válvula e de acionamento do motor que são semelhantes e possuem, cada uma, uma ação para realizar de o acionamento com base na lógica de controle especificada na equação que analisa o tempo de ativação adequado para cada componente. Os diagramas paramétricos e de atividade da visão comportamental, além do detalhamento dos pacotes da visão estrutural, são disponibilizados em um repositório do GitHub¹.

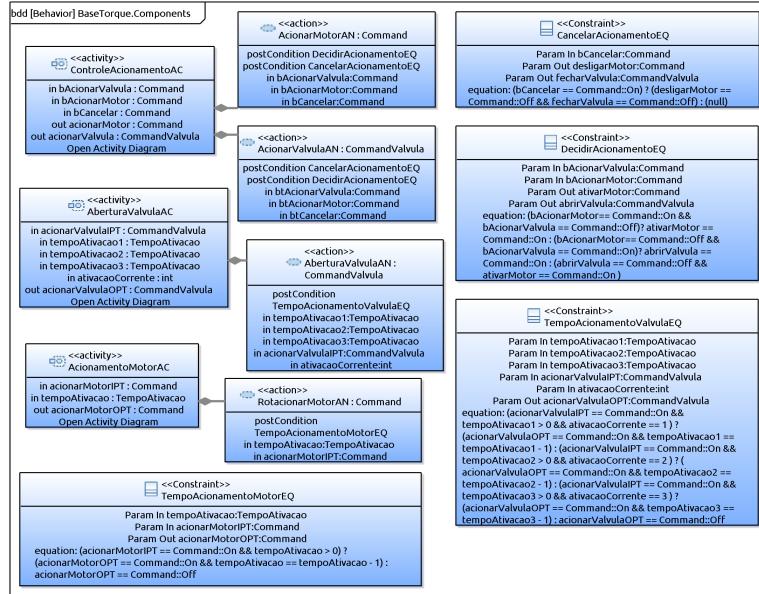


Fig. 3. Diagrama BDD do comportamento dos componentes do sistema embarcado

Por fim, o último artefato é apresentado na Figura 4 e organiza os requisitos funcionais (FR) e não funcionais (NFR). O requisito funcionais de id=1 é o requisito geral que especifica o controle de acionamento, este é composto por outros requisitos responsáveis pela verificação dos botões. Os botões referentes aos componentes de válvula e motor possuem requisitos funcionais derivados que especificam o controle do componente. Além disso, cada requisito também é associado a um componente da visão estrutural por um relacionamento de *satisfies*.

Os requisitos não funcionais possuem o requisito geral de qualidade com id=2 que é composto por outros requisitos não funcionais que descrevem características como modificabilidade, escalabilidade, disponibilidade e acurácia.

5 Discussão

Nesta seção será descrito um comparativo entre o a documentação original do sistema apresentada no artigo [4] com a modelagem seguindo o SysADL. Em

¹ Disponível em: github.com/guimaraaes/eng_computacao_ufersa/tree/master/base-lancamento-com-reducao-torque-para-foguete

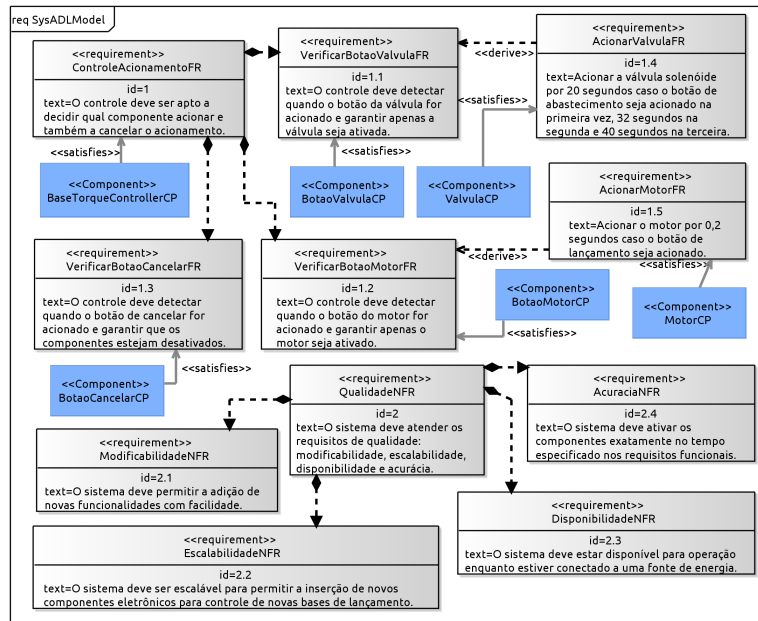


Fig. 4. Diagrama dos requisitos funcionais e não funcionais

relação à modelagem estrutural as duas documentações são complementares. A modelagem utilizada inicialmente apresenta detalhes da montagem eletrônica do circuito necessário para que os componentes possam se comunicar com o microcontrolador. Já a SysADL apresenta a conexão detalhada dos componentes em termos de variáveis de *output* e *input*. Porém com a SysADL foi possível representar as relações entre os requisitos e a quais componentes do sistema eles estão associados (ver Figura 4).

Em relação a visão comportamental o SysADL possibilitou um maior detalhamento de como o sistema se comporta com a divisão das atividades em blocos. Além de serem definidas equações lógicas nos elementos de *Constraint* para compreensão das condições que levam determinada ação a ser aceita com base nos tipos de dados assumidos pelos componentes.

Por outro lado, a modelagem de comportamento descrita no artigo [4] (e comumente utilizada para modelagem de sistemas embarcados) apresenta na Figura 1(b) apresenta uma organização em único fluxograma capaz de sintetizar o funcionamento do sistema sem a necessidade de estudar cada bloco individualmente como a SysADL. Além disso, nenhuma das duas representações apresentam a visão do sistema sob diagramas de Máquinas de Estados Finitos.

Em pesquisa com desenvolvedores de SE com Arduino constatou-se que o SysADL se mostrou atrativo por possuir similaridade com UML e por ser uma proposta para documentação de aspectos de *hardware* também. Apesar de necessitar de um estudo prévio para compreensão do que será apresentado nos diagramas, o SysADL foi considerado apto para documentar de modo complementar projetos com Arduino.

6 Conclusão

Nestes trabalhos apresentamos como a linguagem de modelagem SysADL pode ser utilizada para modelar a estrutura, o comportamento e os requisitos de um sistema embarcado com Arduino.

Diversas técnicas de especificação de requisitos buscam garantir a documentação adequada de um sistema. Esta documentação permite que os desenvolvedores entendam o funcionamento desse sistema mesmo que não tenham participado da sua elaboração e ainda possibilita o mapeamento de novas mudanças na documentação. Neste cenário, a SysADL apresentou vantagens em relação aos diagramas comumente usados para modelagem deste tipo de sistema. Através da SysADL foi possível representar os requisitos funcionais e não funcionais e representar como os requisitos funcionais estão mapeados com os componentes do sistema. Porém sentimos a necessidade de representações complementares que pudessem tornar claro como ocorre a conexão dos componentes sob a perspectiva de circuito eletrônico.

Como trabalho futuro pretendemos realizar a modelagem do mesmo sistema embarcado em outras linguagens e definir critérios de comparação entre estes modelos e o modelo baseado na SysADL. Também pretendemos investigar potenciais usos da documentação baseada na SysADL na elaboração de casos testes para o sistema embarcado.

Referências

1. Agner, L.T.W., Soares, I.W., Stadzisz, P.C., Simão, J.M.: A brazilian survey on uml and model-driven practices for embedded software development. *Journal of Systems and Software* **86**(4), 997–1005 (2013). <https://doi.org/10.1016/j.jss.2012.11.023>, <https://www.sciencedirect.com/science/article/pii/S0164121212003160>, SI : Software Engineering in Brazil: Retrospective and Prospective Views
2. Damke, D.E., de Moraes, P.F., de Oliveira Melo, C.: Avaliação do processo de gerenciamento e engenharia de requisitos em mpes de sistemas embarcados: um estudo de caso (2008)
3. Filho, C.A.M.H., da Cunha, A.M., Ramos, D.B.: Especificação de requisitos, modelagem e implementação de um sistema de software embarcado e de tempo real para os sub-sistemas de controle de voo, potência e combustível de um veículo aéreo não tripulado. (2007)
4. Negreiros, S.G., de Oliveira, G.F.B.: Automação para a base de lançamento de um foguete pet com redução de torque (2020)
5. Santos, L., Silva, E., Batista, T., Cavalcante, E., Leite, J., Oquendo, F.: An architectural style for internet of things systems. In: *Proceedings of the 35th Annual ACM Symposium on applied computing*. pp. 1488–1497. SAC '20, ACM (2020)
6. da Silva, R.A., Castro, J.F.B.: Elicitação e Especificação de Requisitos para Sistemas Embarcados. (2017), monografia (Mestrado em Ciência da Computação), UFPE (Universidade Federal de Pernambuco), Recife, Brazil
7. UFRN: sysadl (2017), <https://sysadl.imd.ufrn.br/>, acesso em: 01 de dezembro de 2021