Sistema para gestão da rede elétrica residencial

Bernard Ambrosio Georges
Mestrado em Engenharia Informática
Braga, Portugal
pg53698@alunos.uminho.pt

João Paulo Peixoto Castro
Mestrado em Engenharia Informática
Braga, Portugal
pg53929@alunos.uminho.pt

Universide do Minho - Campus de Gualtar, Braga, Portugal

1 Introdução

A energia é um recurso que está presente no dia a dia da população, com especial destaque na sua utilização para fins residenciais. Este recurso é indispensável para alimentação dos nossos eletrodomésticos, iluminação dos nossos ambientes e o funcionamento de uma infinidade de dispositivos que utilizámos diariamente. No entanto, a maioria das fontes de energia atualmente usadas não são de fontes renováveis. Este aspeto é fonte de grande preocupação, pois representa um desafio significativo em termos da sustentabilidade da sua produção. Porém, o uso de fontes renováveis também possui um conjunto de dificuldades.

Deste modo e em seguimento da pesquisa previamente feita, o grupo apresenta a criação de um sistema multiagente que gere uma rede elétrica de um grupo de casas que através da coordenação dos vários membros tenta reduzir o efeito da variabilidade das suas fontes.

Ao utilizar uma abordagem multiagente, cada residência pode ser considerada um agente, sendo assim capaz de tomar decisões independentes em relação a seu consumo. No entanto, esses agentes também são capazes de interagir e cooperar entre si, compartilhando informações e recursos de forma a otimizar a utilização das fontes renováveis disponíveis.

Essa coordenação entre os membros da rede permite uma melhor adaptação para as flutuações na geração de energia, contribuindo para o balanceamento da oferta e demanda. Reduzindo, por isso, os efeitos negativos da variabilidade das fontes renováveis.

2 Arquitetura

Esta secção tem como objetivo expor a arquitetura criada no projeto de uma forma geral, apresentando as várias etapas e comportamentos tomados pelos agentes desde a produção da energia até a sua entrega às casas. Sendo assim, primeiramente, é importante caracterizar o tipo da arquitetura do sistema criado. Visto que o sistema contém agentes reativos e um agente deliberativo, podemos considerar a arquitetura deste como uma arquitetura híbrida.

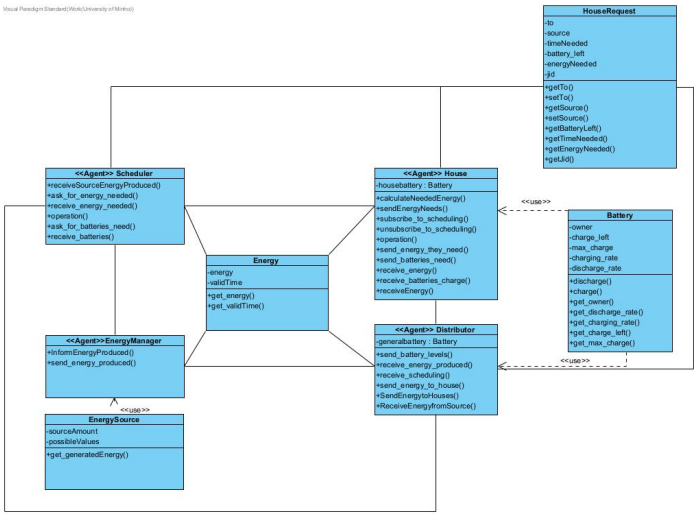

Fig. 1. Diagrama de classes

2.1 Agentes

É fácil perceber que as classes mais importantes numa arquitetura são os agentes, uma vez que estes são os pilares usados para a construção do sistema. Por isso é importante perceber qual a utilidade que cada agente possui e quais são os comportamentos usados de modo que cada um concretize a sua função.

Fonte Energética Este agente é responsável por medir e encaminhar a energia produzida durante a execução do sistema. Sendo assim, este agente informa periodicamente o agente responsável pelo agendamento das casas a energia a ser produzida no período de abastecimento seguinte. Simultaneamente, este reencaminha a energia produzida pelas fontes para o sistema de distribuição.

Infelizmente, devido ao curto tempo não foi possível implementar um modelo realista da produção de energia, visto que seria necessário utilizar os valores da API produzidos pelas fontes e treinar um sistema inteligente nesses dados. Deste modo, para exemplificar foi criado uma classe que usando os dados recolhidos da API *weathermap* relativamente a velocidade do vento, temperatura e humidade calcula a energia produzida, permitindo assim uma representação mais realística.

Distribuição Como apresentado no diagrama 1, o sistema de distribuição é constituído por duas classes. A principal é responsável pela transmissão de energia, sendo que esta possui ainda uma bateria. É importante ressaltar o elemento subsequente, visto que é um mecanismo de precaução e ajuste da rede caso haja insuficiência energética. Contudo, a adição deste componente apresenta alguma complexidade ao sistema de distribuição, dado que o distribuidor agora deve

regular a fonte a usar no envio da energia. Consequentemente, após a receção da energia e do escalonamento, o agente procede a enviar a energia necessária para as casas verificando a fonte para o pedido e casa que a deve receber. Este processo decorre de uma forma cíclica até o final do período válido do escalonamento enviado. Atualmente, este período é fixo, e é definido pelo utilizador na criação do sistema.

Agendamento O sistema de agendamento (scheduler) representa o cérebro do sistema, sendo responsável pela coordenação entre os agentes e a ordenação e satisfação dos pedidos vindos das casas consoante a produção possível. De modo a cumprir isso, o agente espera ser informado da energia a ser produzida no próximo ciclo de abastecimento para iniciar o escalonamento dos pedidos. Visto que o sistema já possui a energia a ser produzida, o agente agora deve recolher as necessidade de todas as casas e informá-las do início de um novo ciclo e, consequentemente, de uma nova agenda. Assim que este receber as necessidades das casas começa por ordenar as casas consoante a sua necessidade, tempo necessário de energia e bateria disponível. De seguida, caso reste energia disponível, o agente faz o escalonamento das baterias das várias casas usando como critério a quantidade de carga que a mesma contém, semelhante a um leilão. Por fim, o agente envia a agenda criada para o sistema de distribuição que, como explicado em cima, distribui a energia.

Com a motivação de permitir a escalabilidade do sistema, foi proposto e posteriormente implementado um sistema de subscrição permitindo que as casas possam se inscrever, entre escalonamentos, no agente, de modo a ser incluído na lista de casas a participar no escalonamento seguinte. É importante ressaltar que apesar desta arquitetura ser completamente assíncrona, a construção da agenda é um momento completamente síncrono, impossibilitando a receção de mensagem que não sejam esperadas naquela instância. Esta questão é proeminente no caso onde o agente de agendamento está no momento da criação do novo escalonamento e uma casa quer se registar, visto que a residência não conseguirá fazê-lo devido o bloqueio no qual o agente se encontra. Sendo assim, neste caso em específico, a casa deverá esperar o fim desta fase para se inscrever propriamente no sistema.

Casa Por fim, a casa representa o agente responsável pelo cliente do sistema, sendo que o seu comportamento envolverá representar as necessidades das casas nos seus pedidos. Assim, de modo a executar a sua função, o agente ao receber a informação do início de uma nova agenda este envia a necessidade da casa juntamente com o estado atual da sua bateria pessoal. Para além disso, este sistema deve estar atento a receção da energia alocada tanto a sua casa como a sua bateria pessoal. Sendo que em último caso, a casa utiliza a sua bateria para suportar as suas necessidades.

Analogamente ao que ocorre no agente da fonte não foi possível criar um modelo para determinar a energia necessária, sendo por isso implementado um sistema que utiliza o tamanho da casa juntamente com um fator aleatório para

determinar a necessidade que a casa possui. Outra particularidade deste agente é o facto deste ser o único agente a ser inicializado isoladamente, sendo que os restantes são criados na execução de uma *main* geral, enquanto que este possui a sua própria *main*. Esta alteração foi feita para permitir a criação de múltiplas casas sem a necessidade de alterar, reiniciar ou modificar o restante do sistema de qualquer modo.

3 Protocolo de comunicação

Para que haja a coordenação entre os múltiplos agentes é necessária a criação de um protocolo usado pelos mesmos para transmitir e receber a informação desejada. Deste modo, esta secção tem como objetivo apresentar o protocolo e explicar a nomenclatura e motivação por detrás da criação das mensagens. Para facilitar a visualização destas etiquetas foi criado um diagrama de sequência abaixo representado. Este diagrama apresenta tanto o funcionamento do sistema como as várias mensagens para produzir o efeito desejado.

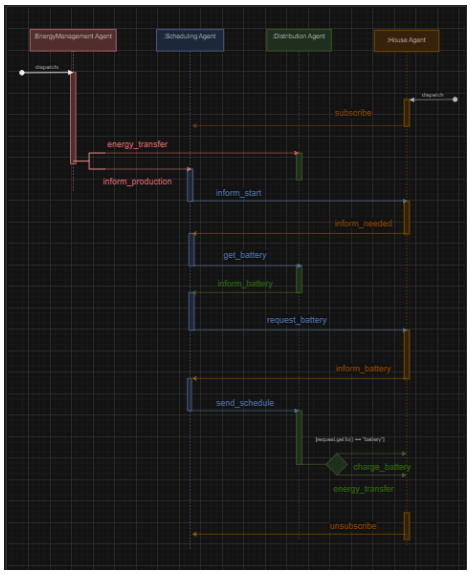

Fig. 2. diagrama de sequencia

Para complementar e para compreender a troca de mensagens entre os agentes foi feito um segundo diagrama. O diagrama ilustra as *performatives* trocadas entre cada classe de um modo mais simples e conciso.

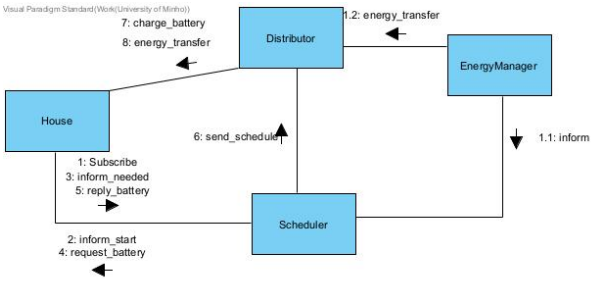

Fig. 3. diagrama de comunicação

3.1 Classes Auxiliares

Uma parte importante da criação da mensagem é a estruturação usada para transferir a informação desejada entre os agentes, sendo que esse efeito pode ser concretizado de múltiplos modos. Para facilitar a formatação da mensagem foram usadas classes que serviram como moldes para a codificação e descodificação, possibilitando a transferência segura e fiável da informação deseja.

Estes parágrafos têm o propósito de explicar o raciocínio por detrás da criação destas classes auxiliares responsáveis pela transferência de informação.

Energy A mais simples das duas classes, esta tem como objetivo representar a energia transferida entre os agentes. Esta classe é usada predominantemente no distribuidor, visto que toda a energia transferida no sistema passa pelo agente.

Estrutura da Classe:

- **energy** - usado para enviar a quantidade de energia transferida por hora.
- **validTime** - tempo (em horas) da transferência de energia.

House Request Está é a classe usada em todas as mensagens para o escalonamento das casas. Sendo assim esta tem uma papel fundamental para a comunicação entre a casa, agente de escalonamento e o distribuidor. Esta mensagem é usada para representar os pedidos das casas e o escalamento, sendo por isso uma mistura de elementos da casa e da distribuição.

Estrutura da Classe:

- **jid** - id do agente responsável pelo pedido.
- **energyNeeded** - energia necessária por hora.
- **timeNeeded** - tempo (em horas) do pedido de energia.
- **battery_left** - bateria restante na sua bateria pessoal da casa.
- **source** (fonte/bateria do distribuidor) - fonte a ser usada para fornecimento de energia.

– **to (casa/bateria)** - visto que o envio pode ser tanto para a casa como para a sua bateria, é necessário distinguir o mesmo na distribuição. sendo usado esta variável para representar o recetor.

3.2 Tipos de performatives

Como é possível verificar no diagrama 3, um agente recebe várias mensagens com múltiplas informações e objetivos. Sendo assim, devido o fato do sistema ser assíncrono, é necessário definir explicitamente cada ação de forma que o agente saiba o significado e objetivo que cada mensagem possui. Para isso foram definidas as seguintes *performatives*, que tem como objetivo rotular cada mensagem de forma a possibilitar o processamento das mensagens. Esta parte do relatório apresenta o propósito de cada *performative* criada.

subscribe Esta mensagem vem com o jid da casa no seu cabeçalho sendo que ao ser recebida pelo agente de agendamento este adiciona o identificado na sua lista de casas registadas da rede elétrica.

unsubscribe Este comportamento é o oposto do previamente apresentado e é lançado no final da vida do agente da casa, de modo a remover a casa da lista de subscritores do agente de agendamento. Representa o fim da execução daquele agente casa.

inform_production: Usada para informar o agente de agendamento da energia produzida pelo agente, esta mensagem marca o início do novo escalonamento energético, servindo como uma bandeira de partida para o agendamento.

energy_transfer: Este *performative* é, como mostra o diagrama 2, a segunda parte do comportamento do agente de fonte. Este tem como objetivo iniciar a transferência da energia produzida pelas fontes para o agente responsável por sua divisão.

inform_start: Como referido na secção do agente, assim que o agendamento receber a mensagem rotulada *inform_production* ele começa o seu processo enviando as mensagens de início da fase de escalonamento para todas as casas subscritas na sua rede através do uso desta mensagem.

request_battery: Como o nome indica, esta *performative* tem como propósito pedir às casas as suas baterias. Este rótulo pode ser usado em dois destinatários, podendo ser usado para requisitar a bateria das casas ou a do agente de distribuição. Consequentemente, esta mensagem é usada em duas instâncias pelo agente de agendamento, no caso de necessitar mais energia para além da produzida pela fonte, sendo enviada, nesse caso, para o distribuidor e no caso de haver sobre-produção e ser possível carregar pelo menos algumas baterias das casas.

send_schedule: Última mensagem na fase de agendamento, esta envia o escalonamento criado para o agente distribuidor usado uma lista de *HouseRequest* para representar os pedidos feitos pelas casas e as baterias a serem carregadas.

ack_start: Está é a mensagem usada para reconhecer a subscrição da casa na rede elétrica do agente de agendamento. Assim que a casa receber esta mensagem, o comportamento, periódico, do *subscribe* é finalizado, sendo também removido da lista de comportamentos a ser reproduzidos por seu agente.

ack_stop: Esta é a mensagem usada para reconhecer a remoção da casa da rede elétrica do agente de agendamento. Assim que a casa receber esta mensagem, ela finaliza a sua execução.

inform_needed Usada como continuação da troca de mensagens iniciais, como seguimento do recebimento da mensagem acima, o agente casa usa esta mensagem para enviar (informar) a casa das suas necessidades para esta próxima fase de agendamento.

inform_battery Feita como resposta a mensagem acima, esta utiliza a classe bateria no seu corpo para enviar os dados da mesma para o agente de agendamento.

inform_stop Usada na finalização do sistema principal. Informa a casa o fim da rede de modo que a casa se desligue também.

charge_battery Finalmente, ao receber o agendamento, o agente de distribuição pode finalmente enviar a energia para as casas corretas. Sendo assim, ele usa esta *performative* juntamente com a *energy_transfer*. Esta é usada no caso do agente estar a enviar para a casa a energia de carregamento da bateria, enquanto que o *energy_transfer* é usado para satisfazer os pedidos iniciais da casa. Esta divisão foi feita pelo facto de apesar de serem entidades diferentes, a bateria pertence à casa e este possui o mesmo jid do seu proprietário, sendo, por isso, necessário a separação de uma forma em relação às mensagens.

4 Fluxo de Atividades

De seguida, após a descrição das múltiplas *performatives* será apresentado o fluxo do sistema e será também explicado o fluxo do comportamento de cada agente de modo a facilitar a perceção do funcionamento individual de cada agente dentro do sistema global.

Assim, para oferecer uma visualização fácil do caminho que o sistema passa durante a execução do escalonamento foi criada o diagrama de atividades a seguir.

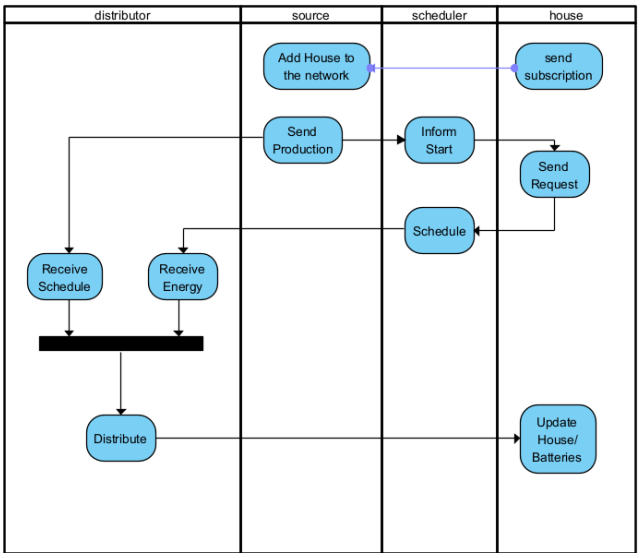

Fig. 4. Diagrama de atividades

4.1 Fonte

Comportamento periódico que envia a energia produzida pelas fontes para o agente de agendamento através do uso da *perfomative*: *inform_production* e envia a energia ao distribuidor através da *energy_transfer*. Este comportamento é o ponto de partida para os períodos de escalonamento. Por isso o período do comportamento será o mesmo que o tempo entre escalonamentos.

4.2 Escalonamento de energia

Comportamento cíclico responsável pelo agente de agendamento. Este está encarregue de responder e enviar todas as mensagens necessárias para manter e escalonar a rede.

Sendo assim, este agente possui o seguinte fluxo durante o seu período de escalonamento com as *performative* que recebe no início da fase e as que envia no final.

1. (inform_production) Recebe a energia produzida pelas fontes.
2. De seguida o agente envia às casas o início do período de escalonamento. (inform_start)
3. (inform_needed) Assim que o agente termina o envio, ele fica à espera da resposta de todas as casas em relação aos seus pedidos.
4. Ordenação dos pedidos consoante a quantidade, tempo e bateria restante da casa.
5. Pede ao agente de distribuição a sua bateria. (request_battery)

6. Caso sobre energia vinda da fonte ou da bateria do distribuidor, ele pede às casas a informação sobre as suas baterias. (request_battery).
7. (inform_battery) Recebe a bateria das casas.
8. Ordena as baterias a serem carregadas.
9. Envia todos os pedidos ordenados para o agente de distribuição (send_schedule).

É importante denotar que durante este processo o comportamento se encontra numa fase rígida onde só processa as mensagens esperadas, ou seja, caso receba qualquer outra *performative*, este descarta a mensagem e continua à espera da mensagem esperada.

Como é referenciado acima, para além desta fase, este comportamento pode receber as seguintes mensagens entre períodos:

- subscribe
- unsubscribe

4.3 Casa

Este comportamento, cíclico, recebe as seguintes *performative* respondendo os seguintes valores:

- inform_start - Após receber isto é feito o cálculo das necessidades da casa. Sendo respondida através da *performative inform_ needed*.
- request_battery - Recebida após a anterior. Esta *performative* não é sempre usada durante um período de escalonamento. Após a receção deste o agente, prepara a informação da sua bateria e envia-a através da *performative inform_ battery*
- charge_battery - De seguida o agente pode receber esta mensagem que não possui resposta, mas desencadeia o carregamento da bateria da casa.
- energy_transfer - Normalmente recebida antes da mensagem anterior, esta é usada para "fornecer" a energia necessária à casa.
- inform_stop - Quando esta mensagem é recebida, ela representa o fim da rede.
- ack_stop - semelhante ao de cima, esta mensagem é usada como resposta do comportamento de unsubscribe, simbolizando a sua remoção correta do sistema.
- ack_start - Esta é a resposta do comportamento periódico de subscribe, por isso, quando esta mensagem é recebida o comportamento é terminado e removido da lista de comportamentos da casa.

4.4 Distribuidor

O comportamento final, este é responsável pelo distribuidor, deve esperar a receção da energia vinda da fonte *energy_ transfer*) e o agendamento (send_schedule) vinda do escalonador para, finalmente, enviar a energia para as casas corretamente. Para além disso, este também pode receber pedidos quanto à sua bateria.

Para distribuir corretamente a energia o agente deve verificar a fonte para saber de onde pode retirar a energia e o destinatário do pedido de modo a corretamente reencaminhar a energia. O destinatário não é caracterizado apenas pelo seu jid, mas também pelo componente que alimenta, ou seja, se é suposto alimentar a casa ou é para carregar a bateria. Este componente é diferenciado pela casa através do tipo da *performative* de envio. Para carregar a bateria é usada a *performative charge_ battery* e para alimentar a casa é usada a *energy_ transfer*.

4.5 Comportamentos Pontuais

Enquanto que os comportamentos apresentados acima são contínuos, o que significa que ficam ativos na duração completa dos agentes, os comportamentos apresentados a seguir são pontuais, sendo usados apenas em momentos muito específicos da vida do sistema.

Subscribe Comportamento periódico lançado na criação (inicio) do agente da casa sendo responsável pela adição do mesmo na rede elétrica. Este envia mensagens do tipo *subscribe*.

Unsubscribe Comportamento periódico lançado na finalização (fim) do agente da casa, sendo responsável pela remoção do mesmo na rede elétrica. Este envia mensagens do tipo *unsubscribe*.

A representação destas ações são feitas sempre no início ou no fim dos diagramas, porém estas podem ocorrer entre escalonamentos sem a necessidade de reiniciar ou recriar o sistema.

Apesar destas ações serem únicas e ocorrerem apenas na iniciação do agente da Casa optou-se por fazer estes comportamentos periódicos, ao oposto de comportamentos one-shot, devido ao "bloqueio" que o scheduler possui na sua fase de agendamento. Assim, caso uma mensagem seja enviada durante a fase onde o agente do agendamento não está a receber outras mensagens, esta não será esquecida, sendo enviado outros pedidos até a receção do reconhecimento da sua ação.

Paragem do agente de agendamento O único comportamento one-shot no sistema é usado para informar as casas da paragem do sistema. Através do uso da *performative inform_ stop*, este informa o fim da execução de modo a finalizar as casas sem que seja necessária a sua paragem individual.

5 Resultados

Esta secção destina-se a demonstrar os resultados e as várias mensagens usadas para demonstrar o sucesso dos múltiplos passos de desenvolvimento do sistema.

5.1 Subscribe

As capturas de ecrã a seguir apresentam o sucesso da inscrição da casa na rede energética. Em 5 vemos uma subscrição bem sucedida da casa house1. De seguida, mostramos o estado do sistema antes e depois desta ação, respetivamente.

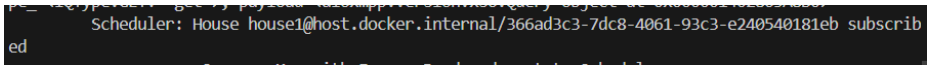

Fig. 5. House Subscription

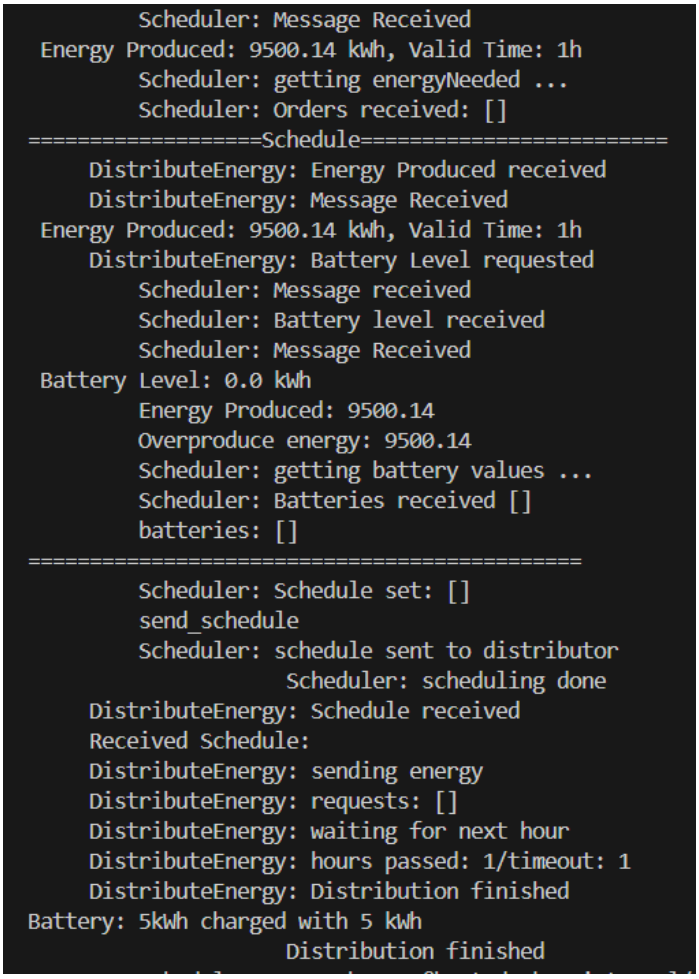

Fig. 6. Pré subscrição

Como é de se esperar ao iniciar o sistema, não existem casas subscritas.

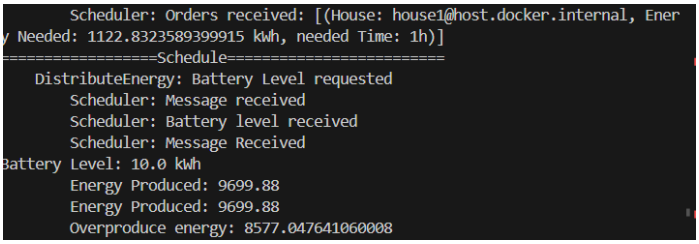

Fig. 7. Agendamento

5.2 Escalonamento

Como é possível notar na imagem acima há uma superprodução de 8577.048 kWh. Assim, o escalonador aproveita para carregar as baterias para além da casa, como é demonstrado abaixo.

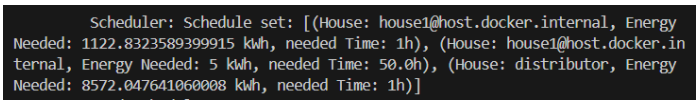

Fig. 8. Escalonamento

É de notar a prioridade que o escalonador toma. Sendo a casa seguida por sua bateria e somente no final é que é carregada a bateria do distribuidor. Para além disso, o carregamento do distribuidor é com toda a energia restante, portanto caso haja um valor muito grande ele deve carregar apenas o seu limite de carregamento.

5.3 Receção de energia

Por fim, é possível notar a receção da energia por parte da casa.

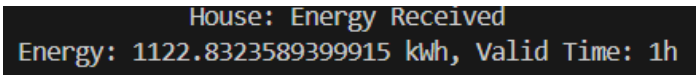

Fig. 9. Receção de energia

Tanto para as suas necessidades como para as da sua bateria.

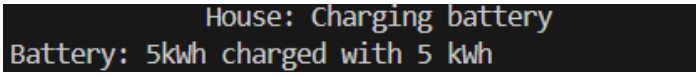

Fig. 10. Carregamento da bateria

5.4 Unsubscribe

A seguir podemos verificar que a casa ao sair do sistema, a execução continua.

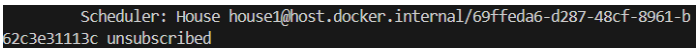

Fig. 11. Unsubscribe

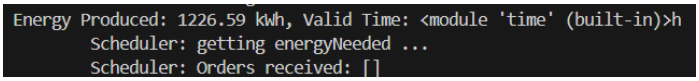

Fig. 12. Escalonamento depois da remoção da casa

5.5 Finalização do sistema principal

É também possível que o sistema principal seja retirado para manutenção. Como se mostra abaixo antes da sua finalização. O agente de agendamento envia corretamente a mensagem de término para todas as casas ainda subscritas na sua rede. Estas, por sua vez, terminam também a sua execução.

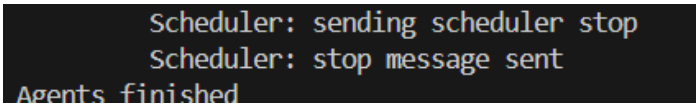

Fig. 13. mensagem de paragem

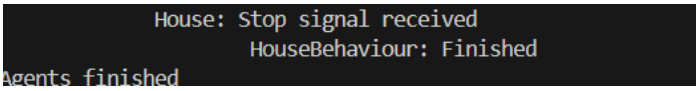

Fig. 14. receção e paragem da casa

6 Notas dos desenvolvedores

Apesar de estarmos satisfeitos com o resultado do nosso projeto há sempre espaço para melhorias, alterações e experiências novas, por causa disso, decidimos deixar neste capitulo final as nossas ideias e opiniões quanto a melhorias e avanços que podem ser feito quanto ao sistema desenvolvido.

Primeiramente, é importante ressaltar a melhoria mais óbvia e previamente mencionada que seria a criação de modelos ML para a previsão e autonomia do sistema. Infelizmente não foi possível testar o sistema num ambiente real ou mesmo virtual que simulasse a conduta do agente.

Para além disso, há experimentos que podem ser feitos à arquitetura do sistema que podem vir a ser vantajosas para a eficiência do sistema.

Um exemplo disto pode ser a diminuição da quantidade de baterias. Um exemplo seria por cada quatro casas haver uma bateria pertencente a uma casa que partilhasse a sua energia com as restantes casas. Isso iria necessitar de uma cooperação muito mais profunda que a atualmente apresentada.

Por fim, para além destas pequenas alterações ao sistema, queríamos oferecer remodelações possíveis do modelo que idealizamos e notamos durante a criação deste que não foram a favor da nossa ideia inicial, mas que poderia vir a ser útil em outros casos.

Em primeiro, seria possível adicionar o fator de custo elétrico ao sistema transformando, por isso, o sistema num híbrido. Seria possível simular o comportamento de um carro elétrico apresentado no artigo [1] onde cada casa deve quando possível carregar as suas baterias em momentos de baixo custo e descarregar quando necessita da mesma ou em momento de alto custo elétricos. Para isso, poderia ser removido o agente de agendamento e seria possível fazer um escalonamento *first come first serve*.

A nossa última ideia foi remover a fonte por completo ou, no máximo, usá-la apenas como um *failsafe* para rede. Neste caso a produção viria das casas que neste caso teriam paneis solares instalados no seu teto. Isto concebido após uma pequena pesquisa apresentou que uma casa com painéis solares pode se auto-sustentar durante os dias na maioria das estações sendo a única exceção o inverno. Nesta pesquisa o autor possuía um carro elétrico, o que aumentou o seu consumo exponencialmente. Sendo assim, o único desafio seria fornecer a energia para a parte da noite. Isto poderia ser feito com o agente de distribuição que durante o dia recolhe a carga restante das casas e durante a noite redistribui o mesmo para as casas de forma a resolver esta discrepância energética.

7 Conclusão

Em conclusão, este projeto proporcionou a oportunidade de criar um sistema que utiliza as potencialidades de um sistema multiagente de modo a otimizar o uso de fontes renováveis de energia.

Por fim, os requisitos do sistema foram cumpridos e dentro do prazo estipulado. O sistema apresenta um conjunto amplo de comportamentos e mensagens

que tomam uso destas para criar um ambiente cooperativo. Para além disso, este trabalho permitiu a utilização prática dos conhecimentos adquiridos no projeto de investigação que serviu como a fundação e inspiração para a criação deste projeto.

References

1. M. Tan, Z. Zhang, Y. Ren, I. Richard and Y. Zhang, "Multi-Agent System for Electric Vehicle Charging Scheduling in Parking Lots," in Complex System Modeling and Simulation, vol. 3, no. 2, pp. 129-142, June 2023, doi: 10.23919/CSMS.2023.0005.