

Criando o seu componente Lucc-ME

Um guia de desenvolvedor

Versão 3.0 | Dezembro 2016

The logo consists of the text 'LuccME' in a bold, sans-serif font. The 'L' is blue, the 'u' is light blue, the 'c' is orange, the second 'c' is yellow-orange, the 'M' is green, and the 'E' is light green. The letters are closely spaced and have a slight shadow effect.

Criando o seu componente LuccME

Um guia de desenvolvedor

Versão 3.0 | Dezembro 2015



Autores:

Ana Paula Dutra de Aguar

Diego Melo

Elói Lennon Dalla-Nora

Talita Assis

Francisco Gilney Silva Bezerra

LuccME

Conteúdo

1	Introdução ao arcabouço Lucc-ME	1
1.1	O que é LuccME.....	1
1.2	A filosofia do LuccME	1
2	Construindo um componente para Lucc-ME	3
2.1	Introdução	3
2.2	Componente de Demanda	5
2.3	Componente de Potencial	5
2.4	Componente de Alocação	5

1 Introdução ao arcabouço LUCC-ME

1.1 O que é LuccME

LuccME é um arcabouço de modelagem espacialmente explícita de código aberto para mudanças de uso e cobertura da terra (LUCC em inglês) desenvolvido pelo Centro de Ciência do Sistema Terrestre (CCST) e seus colaboradores, como uma extensão do Terra-ME (Carneiro et al., 2013). Usando o LuccME o usuário pode criar facilmente modelos espacialmente explícitos de desmatamento, expansão agrícola, degradação florestal, expansão urbana e outras mudanças de uso da terra, em diferentes escalas e áreas de estudo, combinando componentes existentes e/ou criando os seus próprios.

1.2 A filosofia do LuccME

Existem diferentes tipos de modelos LUCC, os quais podem ser classificados de acordo com a finalidade, escala, abordagem ou teoria subjacente. Apesar da diversidade de modelos, uma estrutura funcional comum pode ser identificada em diversos modelos espacialmente explícitos (Verburg et al., 2006; Eastman et al., 2005), conforme ilustrado na Figura 1. Em geral, três componentes principais podem ser identificados: *Demanda* – responsável pelo cálculo da magnitude ou quantidade de mudança; *Potencial* – responsável pelo cálculo da adequação ou suscetibilidade de mudança de cada célula; *Alocação* – responsável pela distribuição espacial das mudanças com base na demanda por terra e potencial de mudança de cada célula. Estes componentes são organizados de maneira *top-down* onde a demanda por terra é alocada espacialmente de acordo com a adequação de cada célula. Diversos modelos LUCC bem conhecidos seguem esta estrutura, incluindo a família de modelos CLUE (Veldkamp e Fresco, 1996; Verburg et al., 1999; Verburg et al., 2001), Dinamica EGO (Soares-Filho et al., 2002) e GEOMOD (Pontius et al., 2001), empregando diferentes técnicas e abordagens para os três componentes (Eastman et al., 2009; Lesschen et al., 2007). Entretanto, estes modelos são implementados em diferentes plataformas computacionais, geralmente não possuem código aberto, portanto, não podem ser facilmente modificados ou combinados.

Neste sentido, o LuccME permite a construção de novos modelos, combinando componentes de *Demanda*, *Potencial* e *Alocação*, desenvolvidos de acordo com as ideias originais dos principais modelos disponíveis na literatura. Entretanto, uma característica distinta do LuccME é sua organização modular. Isso permite ajustar a estrutura do modelo de acordo com as necessidades do usuário. No LuccME, os componentes podem ser selecionados e modificados com base nas necessidades de uma determinada aplicação ou escala de análise e facilmente parametrizados a partir de uma interface simples. O LuccME é construído sobre o TerraME, um ambiente de programação genérico para modelagem espacial dinâmica, desenvolvido para suportar modelos de diversos domínios, incluído, hidrologia, biodiversidade, uso da terra e muitos outros. Sua linguagem de programação tem funções embutidas que tornam mais fácil o desenvolvimento de modelos multi-escala e multi-paradigma para aplicações ambientais. O TerraME oferece interface com a biblioteca geográfica TerraLib (Camara et al., 2008), o que permite aos modelos acesso direto ao banco de dados geográfico. O LuccME, TerraME e TerraLib são produtos tecnológicos desenvolvidos pelo INPE e seus parceiros.

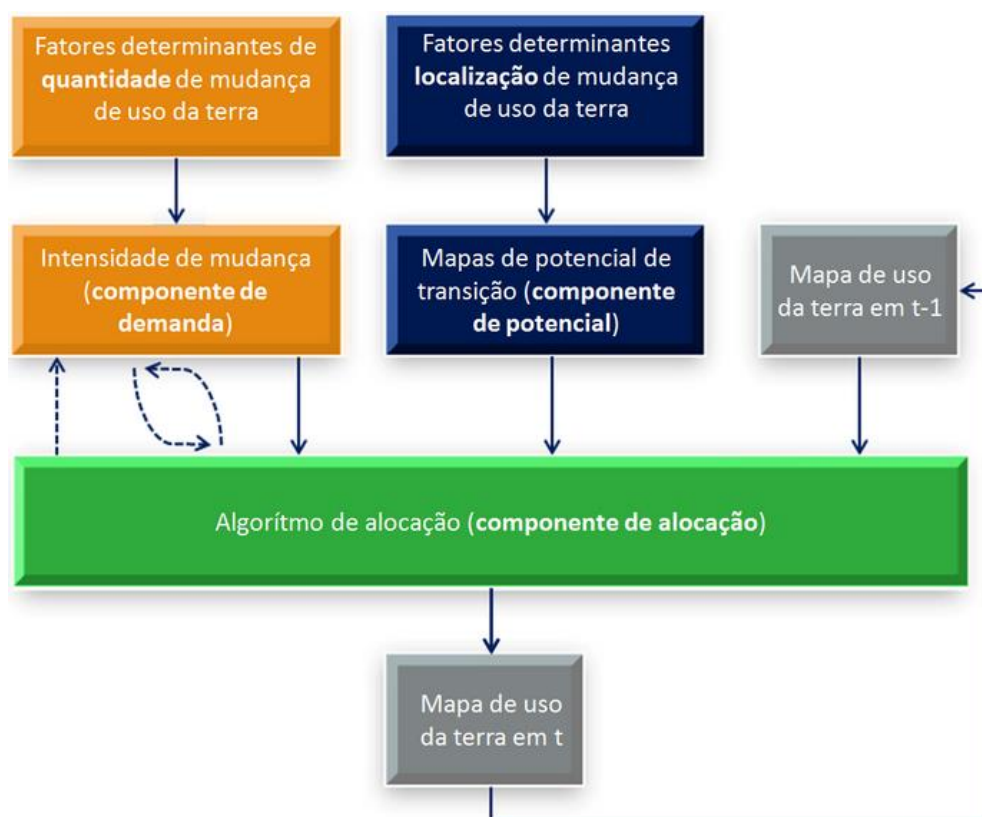


Figura 1: Estrutura geral dos principais modelos LUCCE espacialmente explícitos disponíveis na literatura (adaptados de Verburg et al., 2006).

Além de proporcionar uma maneira fácil de usar e criar novos modelos, o LuccME foi desenvolvido para suportar o desenvolvimento de modelos que podem ser integrados a modelos ambientais multi-escala mais complexos. Neste sentido, os modelos criados utilizando o LuccME são encapsulados em “ambientes”, conceito proposto por Carneiro (2006) e um conceito central no TerraME, para que eles possam ser facilmente acoplados a outros modelos, por exemplo, modelos do sistema terrestre ou modelos de uso da terra multi-escala.

2 Construindo um componente para Lucc-ME

2.1 Introdução

Para construir um componente para o Lucc-ME o desenvolvedor deve conhecer a sua estrutura e seguir as interfaces definidas (*dados que devem estar presentes na saída do componente*).

A Figura 2 mostra a arquitetura básica do Lucc-ME:

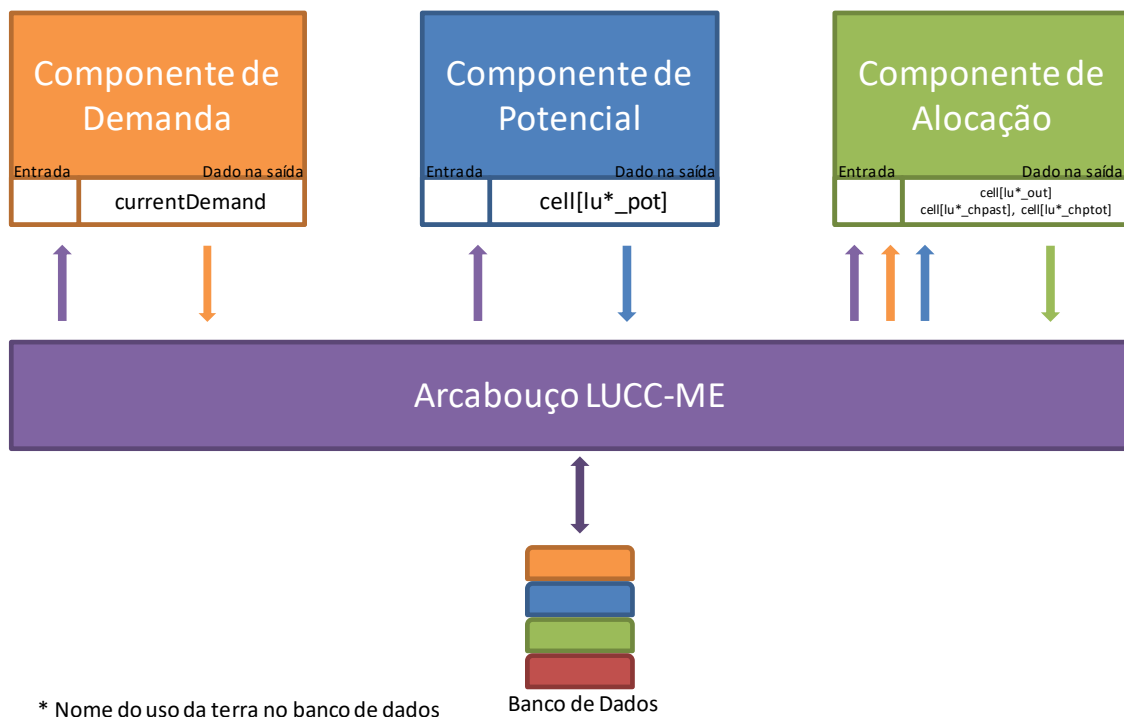


Figure 2: Arquitetura básica do Lucc-ME.

Todo componente deve possuir pelo menos dois métodos, *verify* e *execute*, mas o desenvolvedor pode criar quantos métodos forem necessários para sua aplicação.

O Arcabouço chama primeiramente o método *verify* e posteriormente o *execute*.

É função do método *verify* checar todas as entradas do componente, tais como: se todos os parâmetros de entrada foram declarados no arquivo de configuração (*se não possuir parâmetros opcionais*); a validação desses parâmetros com o banco de dados ou faixa de valores.

Abaixo encontra-se um código de um componente vazio:

```
-- Empty component example
-- @arg component An instance of the component itself (an object, with all the data).
-- @arg component.execute Handles with the execution method of an EmptyComponent component.
-- @arg component.verify Handles with the verify method of an EmptyComponent component.
-- @return The modified component.
-- @usage emptyComponent = EmptyComponent {}
function EmptyComponent(component)
    -- Handles with the rules of the component execution.
    -- @arg self A EmptyComponent component.
    -- @arg event A representation of a time instant when the simulation engine must execute.
    -- @arg luccMeModel A container that encapsulates space, time, behavior, and other
    -- environments.
    -- @usage self.emptyComponent:execute(event, model)
    component.execute = function(self, event, luccMEModel)
        end

    -- Handles with the parameters verification.
    -- @arg self A EmptyComponent component.
    -- @arg event A representation of a time instant when the simulation engine must execute.
    -- @arg luccMeModel A container that encapsulates space, time, behavior, and other
    -- environments.
    -- @usage self.emptyComponent:verify(event, model)
    component.verify = function(self, event, luccMEModel)
        end

    return component
end
```

Apesar do retorno de um componente ser sempre o próprio componente modificado, cada tipo de componente (demanda, potencial ou alocação) possui sua peculiaridade, um conjunto de dados que devem estar presentes no componente modificado, o que será explicado nas próximas sessões.

2.2 Componente de Demanda

O componente de demanda é o primeiro a ser chamado pelo arcabouço, ele é responsável pela demanda anual de cada uso da terra durante o período da simulação.

A entrada do componente de demanda usualmente é a demanda de cada uso da terra para cada ano da simulação (*anual demand*).

A saída de um componente de demanda deve conter uma tabela de demandas, *currentDemand*, que contém a demanda de cada uso da terra para cada ano de simulação, baseada nas regras implementadas no método *execute* do componente.

2.3 Componente de Potencial

O componente de potencial é o segundo chamado pelo arcabouço, ele é responsável pelo potencial anual de cada uso da terra para o período de simulação.

A entrada usual de um componente de potencial são os parâmetros relevantes para o cálculo do potencial, não havendo uma regra específica, uma vez que depende da necessidade do componente.

A saída de um componente de potencial deve conter o potencial anual de cada uso da terra para cada célula durante o período de simulação, seguindo o padrão: `cell[<landUseTypeName>_pot]`, aonde `<landUseTypeName>` é o nome do uso da terra usado na aplicação.

Por exemplo:

Utilizando os usos da terra: floresta, desmatamento e outros; o componente de potencial deve ter em sua saída um `cell[floresta_pot]`, `cell[desmatamento_pot]` and `cell[outras_pot]`.

2.4 Componente de Alocação

O componente de alocação é o terceiro e último chamado pelo arcabouço, ele é responsável pela alocação anual de uso da terra durante o período de simulação.

A entrada usual de um componente de alocação são os parâmetros relevantes para o cálculo da alocação são o potencial de cada célula para cada uso e a demanda de cada uso, mas isso não é uma via de regra, pois depende da necessidade do componente.

A saída de um componente de alocação deve conter o valor anual de cada uso da terra para cada célula durante o período de simulação, seguindo o padrão: `cell[<landUseTypeName>_out]`, aonde `<landUseTypeName>` é o nome do uso da terra usado na aplicação, bem como a diferença entre

os valores de cada célula do início da simulação e o ano a ser alocado, seguindo o padrão: cell[<landUseTypeName>_chtot], e por fim, a diferença entre os valores de cada célula do ano anterior da simulação e o ano a ser alocado, seguindo o padrão: cell[<landUseTypeName>_chpast].

Por exemplo:

Utilizando os usos da terra: floresta, desmatamento e outros; o componente de alocação deve ter em sua saída um cell[floresta_out], cell[desmatamento_out], cell[outros_outt], cell[floresta_chtot], cell[desmatamento_chtot], cell[outros_chtot], cell[floresta_chpast], cell[desmatamento_chpast] e cell[outros_chpast].