Explicit Composition Constructs in DSLs

The case of the epidemiological language Kendrick

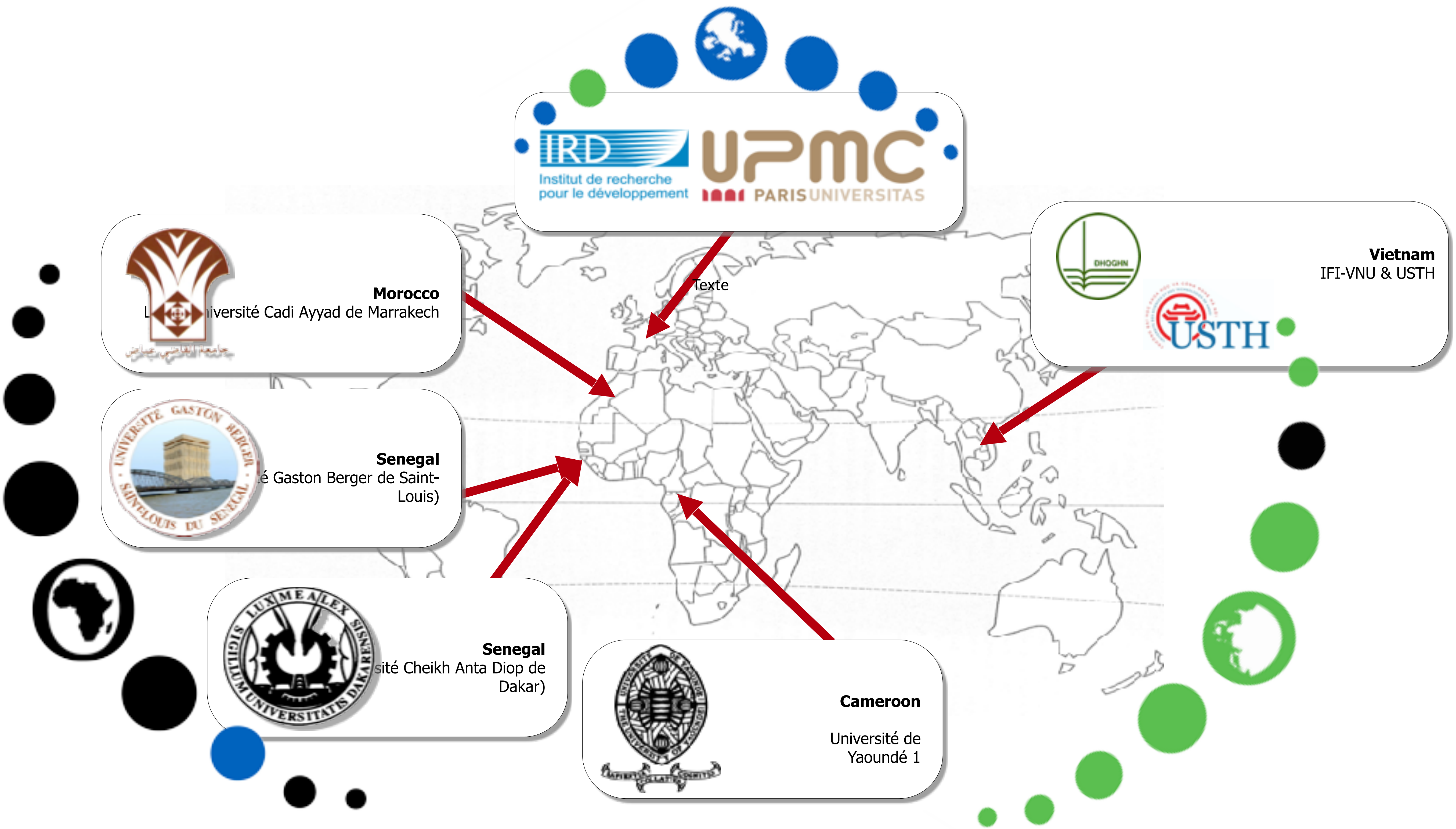
Bùi Thị Mai Anh, Nick Papoulias, Mikal Ziane
Serge Stinckwich

IRD Institut de recherche pour le développement — UMMISCO Unite Mixte Internationale — IFI Institut de la Francophonie pour l'Informatique — LIP6

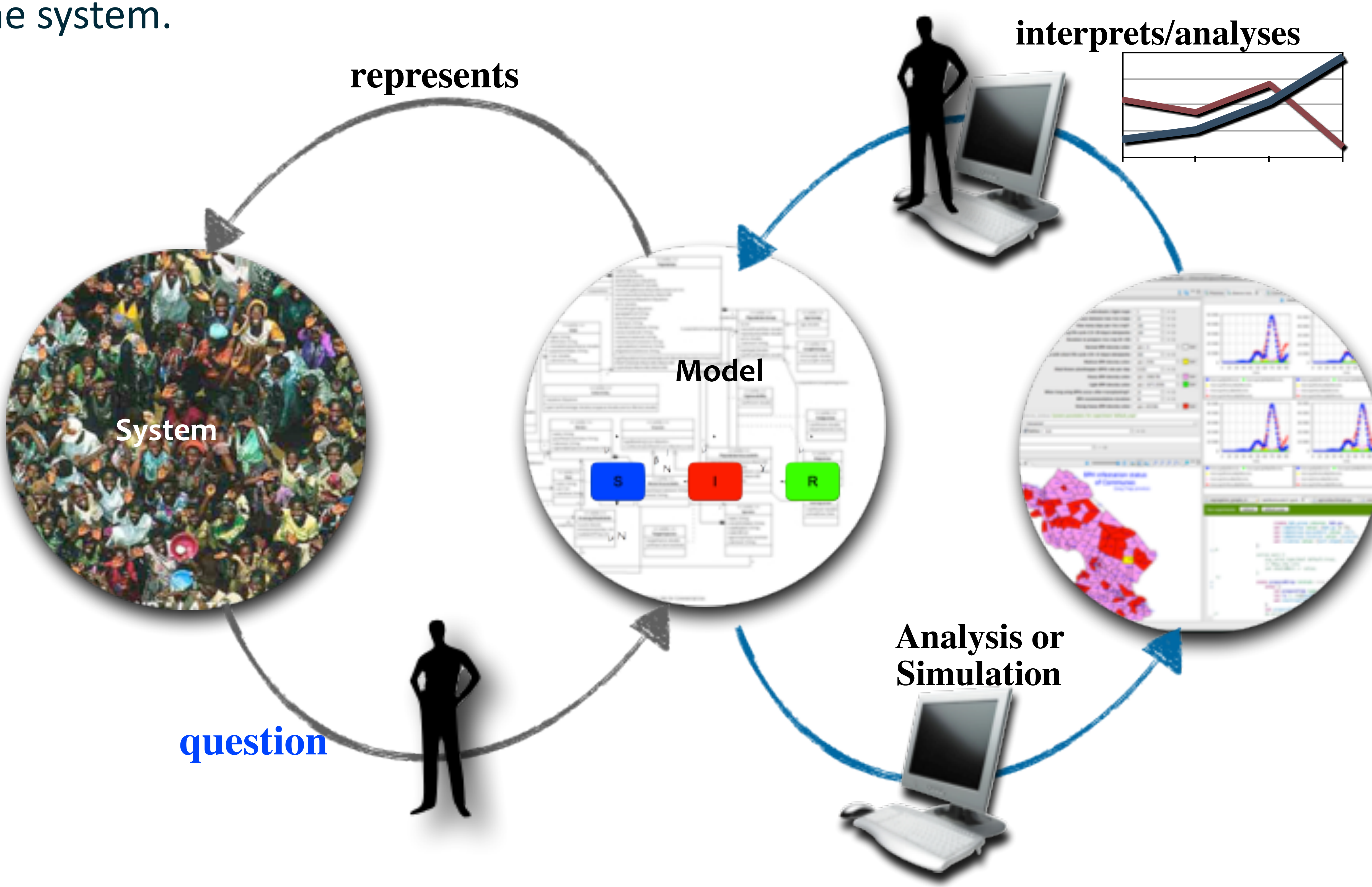
IWST 2016, Prague

International Research Unit UMMISCO Mathematical and Computational Modeling of **Complex Systems Laboratory**

65 members + Phd Students: Professors, Researchers, Associates (26 HDR),1 research engineer, 3 admin staff, 3 post-docts and 45 PhD students



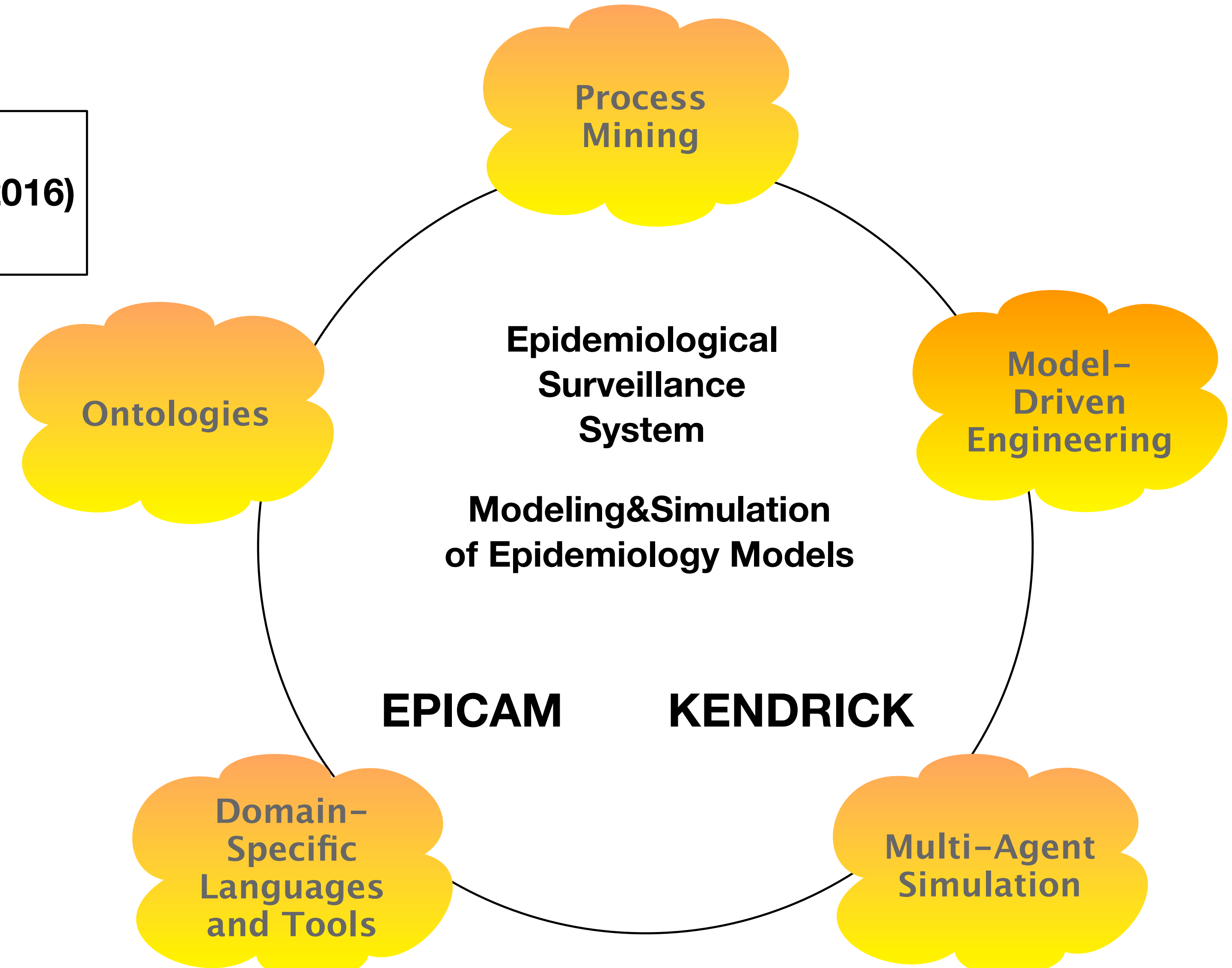
UMMISCO's approach is based on the building of models that are abstract (simplified) representation of a system which supports answering **questions** about the system.



3 Key application domains :(i) Emerging diseases (ii) Climate change and natural hazards.(iii) Ecosystems and natural resources.

UMMISCO East-Central Africa Unit (Yaoundé)
Research Activities on Model-Driven Epidemiology

1 Professor (Maurice Tchuenté)
2 researchers from UMMISCO/Bondy (3 months in 2016)
6 PhD students



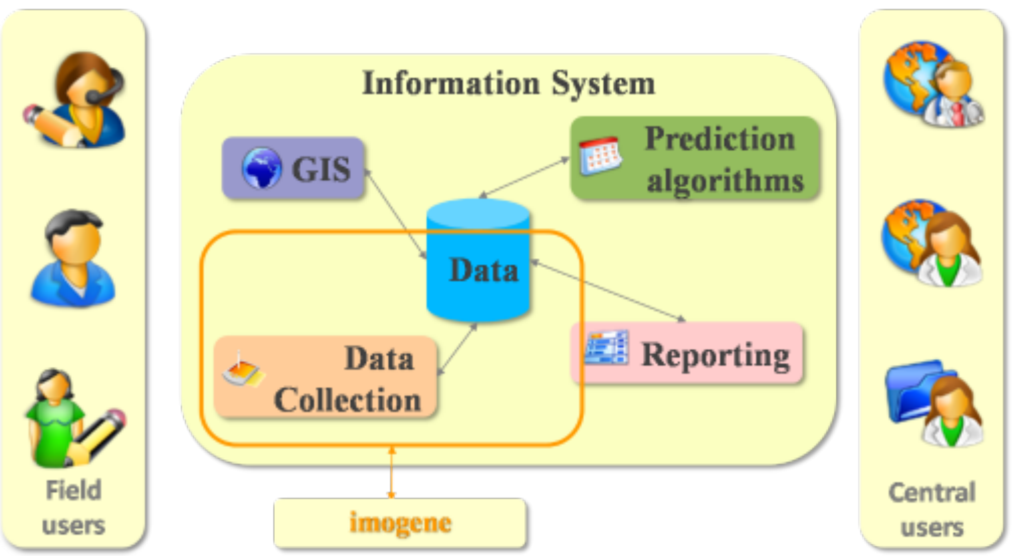
UMMISCO Unite Mixte Internationale

IRD Institut de recherche pour le développement

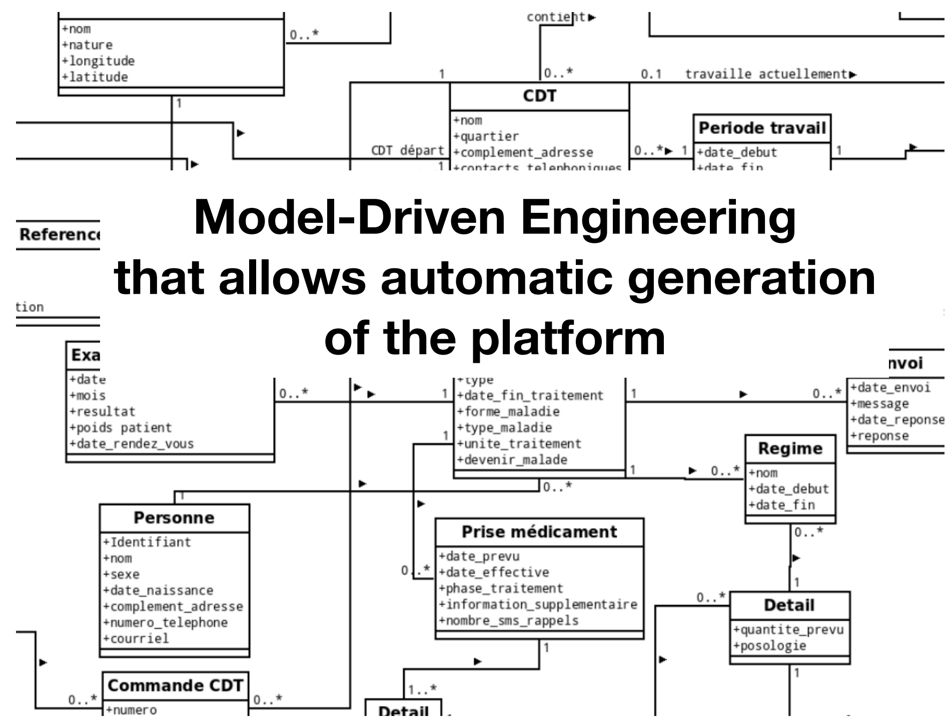
UPMC SORBONNE UNIVERSITÉS

EPICAM is a Model-Driven Engineering Platform for Epidemiological Surveillance System

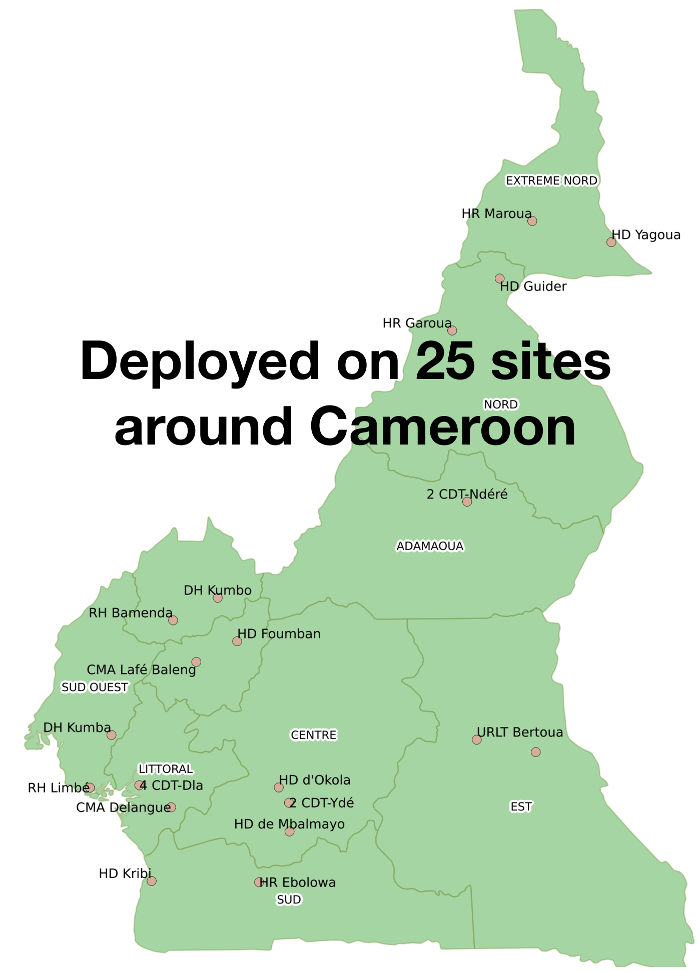
Applied to Tuberculosis but adaptable to other diseases



3 PhD students from Yaoundé 1 University involved



Model-Driven Engineering that allows automatic generation of the platform



Deployed on 25 sites around Cameroon

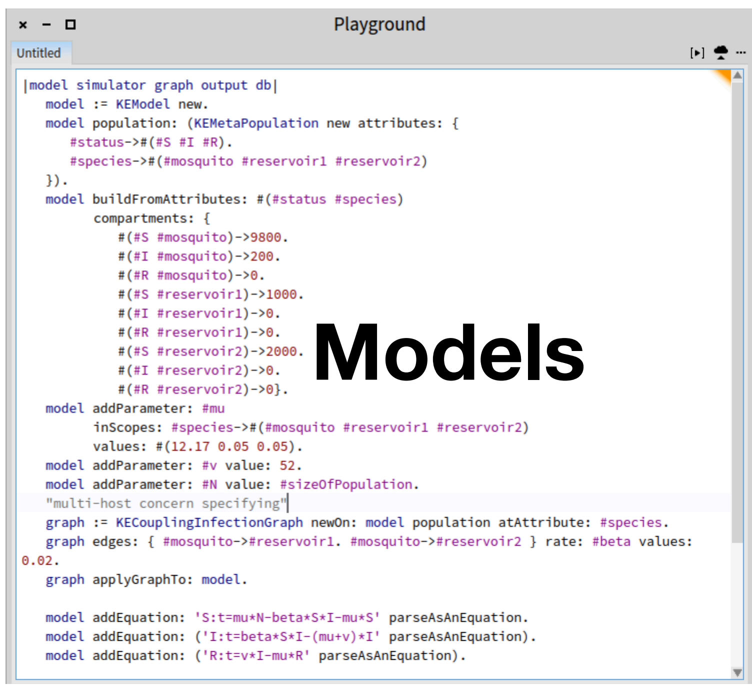
https://github.com/UMMISCO/EPICAM

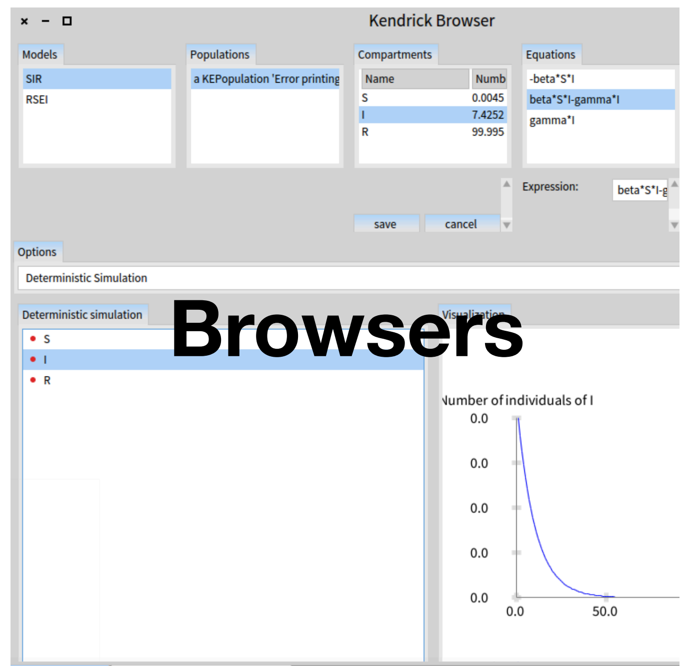
CENTRE PASTEUR DU CAMEROUN
PNLT
Medes Institut de Médecine et de Physiologie Spatiales
UNIVERSITÉ DE YAOUNDÉ I · THE UNIVERSITY OF YAOUNDÉ I · SAPIENTIA COLLATIVA COGNITIO
UMMISCO Unité Mixte Internationale
IRD Institut de recherche pour le développement

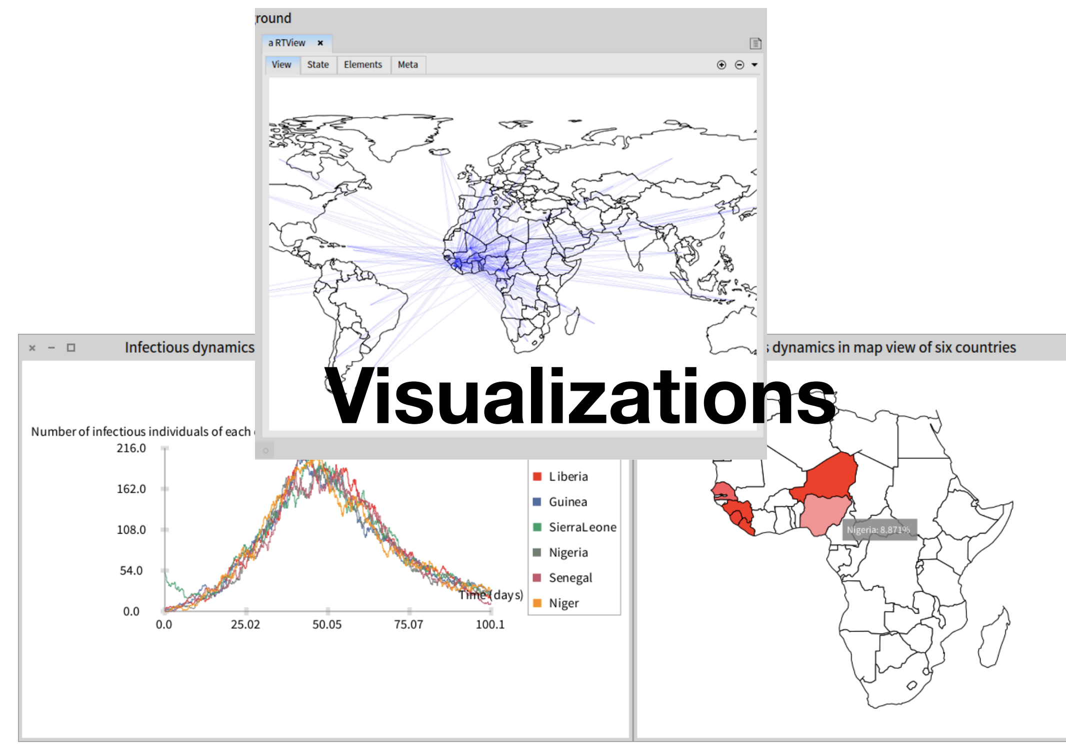
Kendrick is a platform for epidemiological modeling and analysis

It helps epidemiologists craft custom analyses cheaply

2 PhD students from Yaoundé 1 University involved







http://ummisco.github.io/kendrick/

VIỆN QUỐC TẾ PHÁP NGỮ
Institut Francophone International

UMMISCO
Unité Mixte Internationale

IRD
Institut de recherche pour le développement

EUROPEAN SMALLTALK USER GROUP
esug
www.esug.org

OBJECT PROFILE

Google Summer of Code

What is Epidemiology Modeling ?

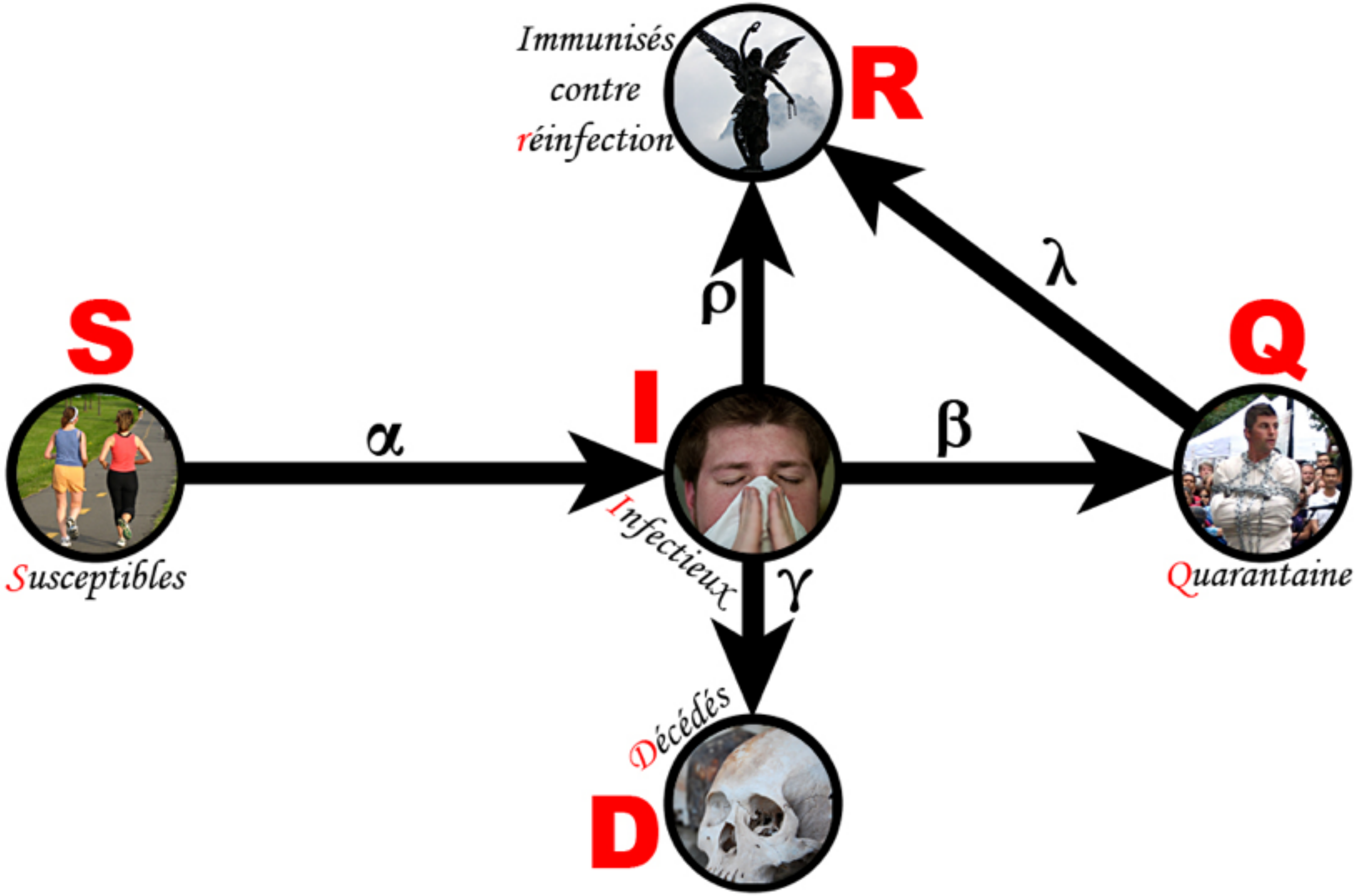
- Building mathematical models to study speed of a disease in a population
- Compartmental models

Anderson Gray McKendrick was born in 1876 in Edinburgh, the last of five children. He studied medicine at the University of Glasgow where his father was a professor of physiology. In 1900 he joined the Indian Medical Service. Before going to India, he accompanied Ronald Ross on a mission to fight malaria in Sierra Leone. He then served in the army for 18 months in Sudan. At his arrival in India, he was appointed as medical doctor in a prison in Bengal where he tried to control dysentery. In 1905 he joined the new Central Institute for Medical Research in Kasauli (in the North of India). He worked on rabies but also studied mathematics. In 1920, having been infected by a tropical disease, he returned to Edinburgh and became the superintendent of the Royal College of Physicians Laboratory.

Fig. 16.1 McKendrick (1876–1943) and Kermack (1898-1970)

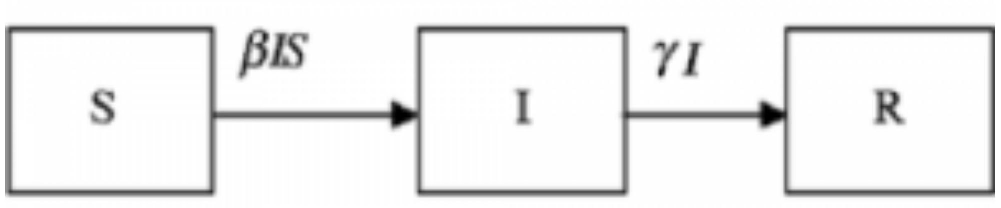
In 1926 McKendrick published an article on the "Applications of mathematics to medical problems", which contained several new ideas. He introduced in particular a continuous-time mathematical model for epidemics that took into account the stochastic aspect of infection and recovery.

N. Bacaër, *A Short History of Mathematical Population Dynamics*, 89
DOI 10.1007/978-0-85729-115-8_16, © Springer-Verlag London Limited 2011



Immunisés
contre
réinfection
R
ρ
λ
S
Q
I
α
β
Susceptibles
Infectieux
γ
Quarantaine
Décédés
D

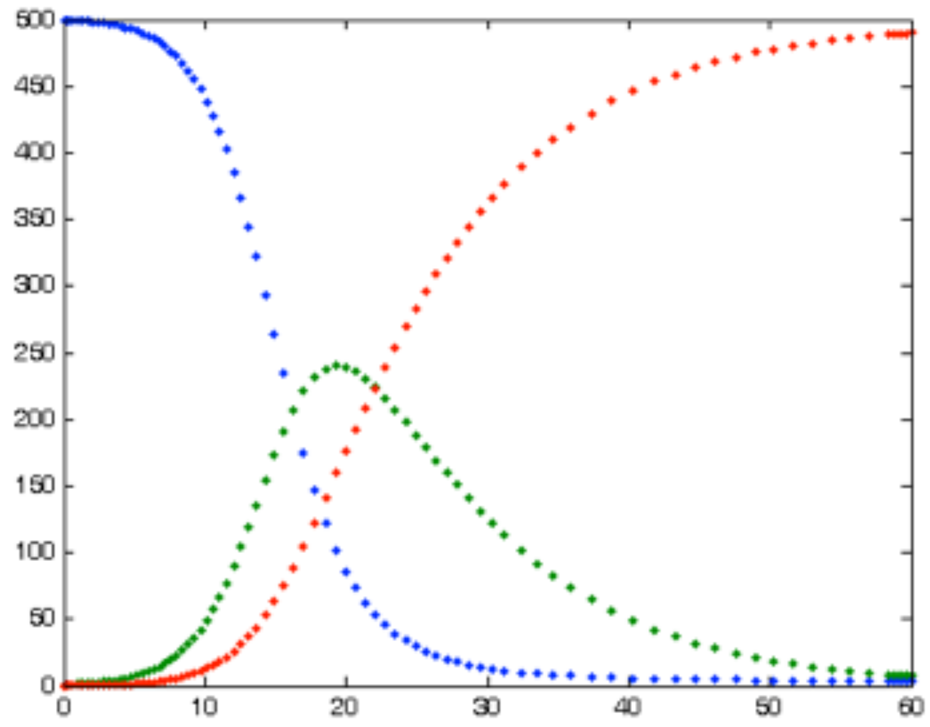
SIR Model



$$\frac{dS}{dt} = -\beta I S$$

$$\frac{dI}{dt} = \beta I S - \nu I$$

$$\frac{dR}{dt} = \nu I$$



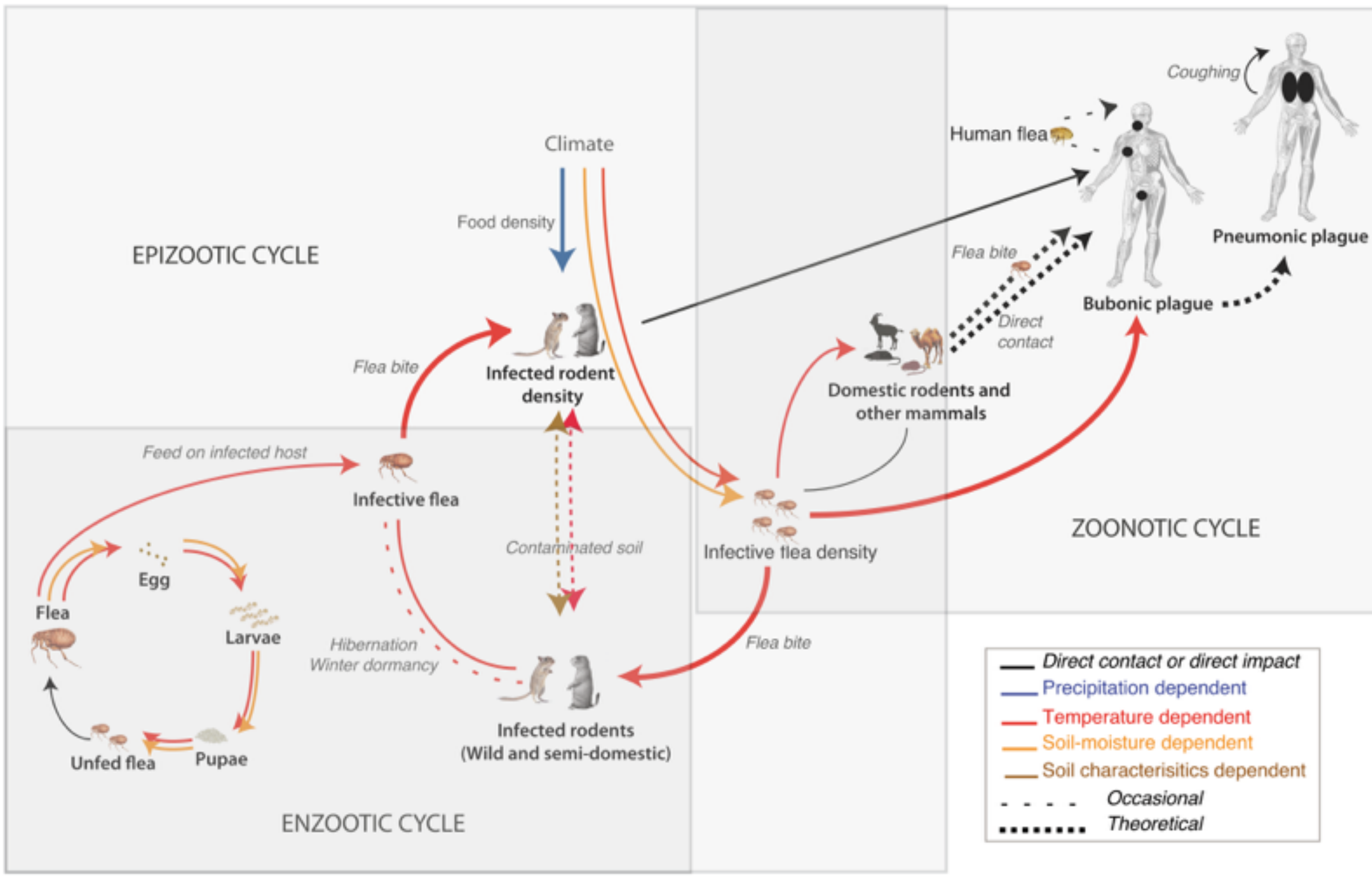
Models Simulations

- 3 ways to do simulations:
  - Population-level: deterministic simulation (ODEs solver)
  - Individual-Level: stochastic simulation (Gillepsie simulation)
  - Agent-based level

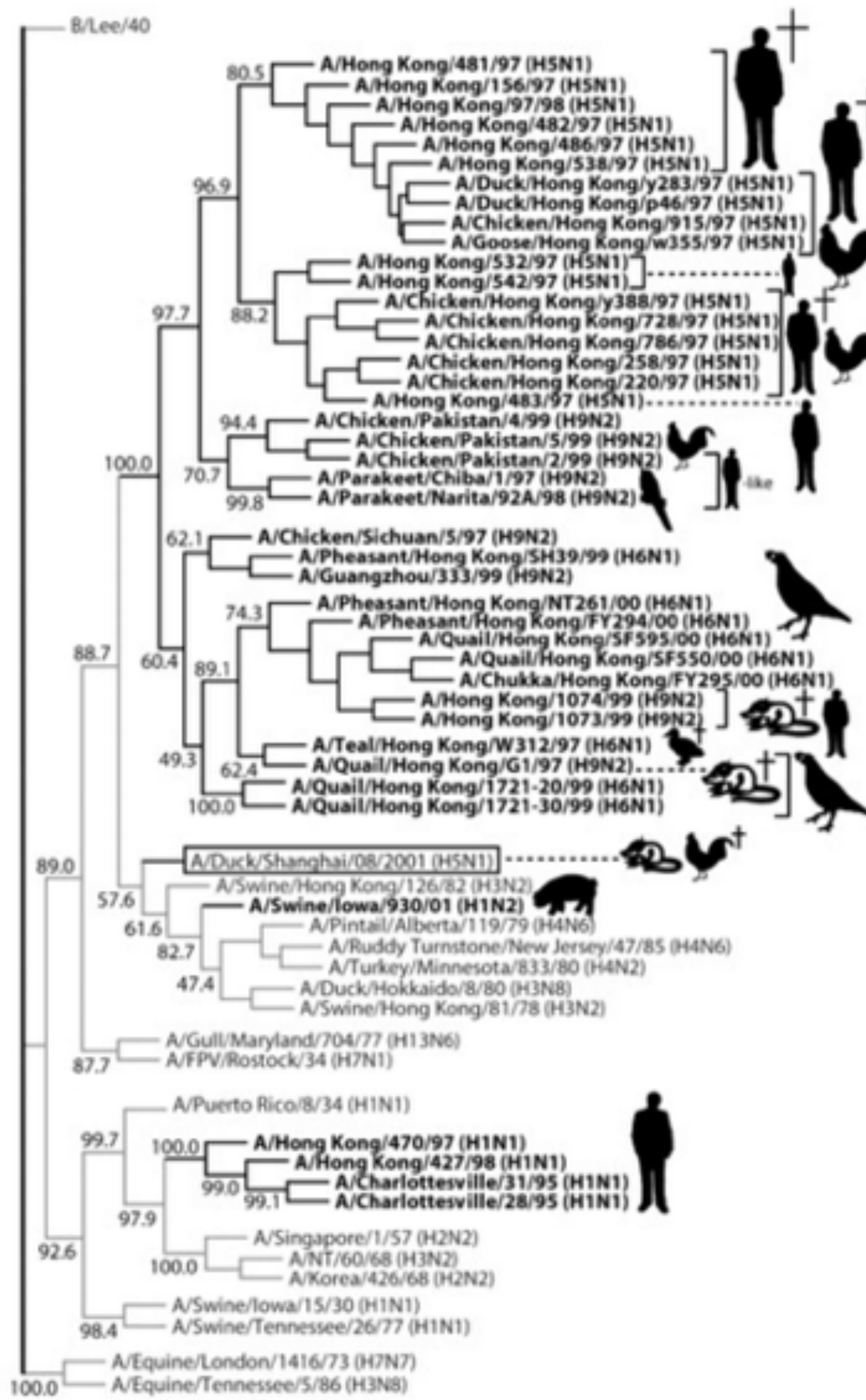
Multi-concerns Models of Epidemiology

- Seasonality
- Multi-hosts
- Multi-strains
- Age/Risk structure
- Spatial aspect
- Control strategies

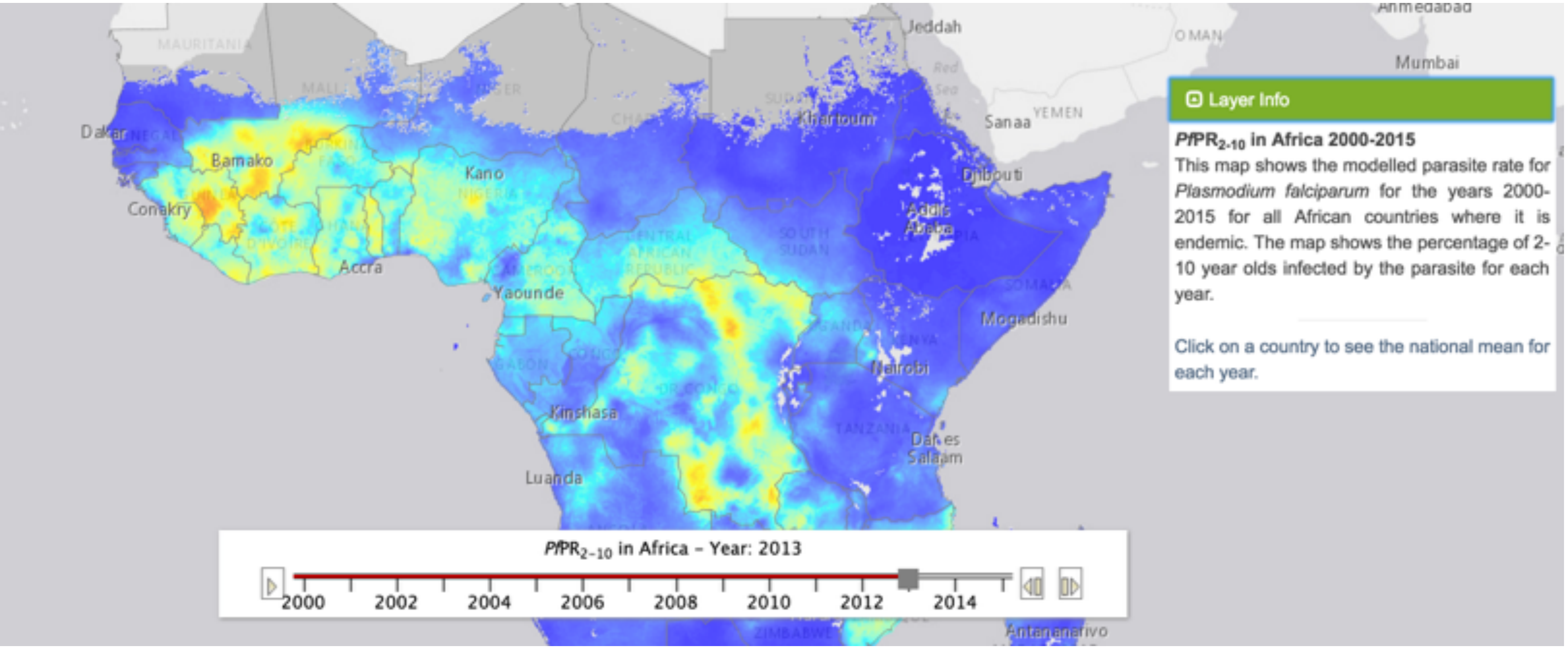
Multi-hosts concerns



Multi-strains concerns



Spatial concerns



Control Strategies concerns

How epidemiologists are building their models ?

SEIRS model

$$\begin{cases} \frac{dS}{dt} & = & \mu N + \nu R - \lambda S - \mu S \\ \frac{dE}{dt} & = & \lambda S - \sigma E - \mu E \\ \frac{dI}{dt} & = & \sigma E - \gamma I - \mu I \\ \frac{dR}{dt} & = & \gamma I - \mu R - \nu R \end{cases}$$

```
I=zeros(5,2);I(1,2)=10;
E=zeros(5,2);R=E; %...
S = reshape(S,[1 ns*np]); E = reshape(E,[1 ns*np]);
I = reshape(I,[1 ns*np]);R = reshape(R,[1 ns*np]);
[T,Y]=ode45(@rightSideAIModel,[0 tMax],[S E I R],options); %...
function res=rightSideAIModel(t, pop)
I=reshape(pop(2*np*ns+1:3*np*ns),[np ns]); %...
lambda=zeros(np,ns);
for p=1:np
    for s=1:ns
        lambda(p,s)=sum(beta(s,:,p).*I(p,:)./N(p,:));
    end
end %...
deltaI = zeros(np,ns); deltaR = zeros(np,ns);%...
for s=1:ns
   deltaI(:,s)=rho(:,:,s)*I(:,s)-sum(rho(:,:,s))'.*I(:,s); %...
end
dSdt = mu.*N + nu.*R - lambda.*S - mu.*S + deltaS;
dEdt = lambda.*S - sigma.*E - mu.*E + deltaE;
dIdt = sigma.*E - gamma.*I - mu.*I + deltaI;
dRdt = gamma.*I - mu.*R - nu.*R + deltaR;%...
```

$$
\begin{cases}
\frac{dS_{ps}}{dt} & = & \mu_{ps}N_{ps} + \nu_{ps}R_{ps} - \lambda_{ps}S_{ps} - \mu_{ps}S_{ps} \\
& & + \sum_{q=1}^{n} \rho_{pqs}S_{qs} - \sum_{q=1}^{n} \rho_{qps}S_{ps} \\
\frac{dE_{ps}}{dt} & = & \lambda_{ps}S_{ps} - \sigma_{ps}E_{ps} - \mu_{ps}E_{ps} \\
& & + \sum_{q=1}^{n} \rho_{pqs}E_{qs} - \sum_{q=1}^{n} \rho_{qps}E_{ps} \\
\frac{dI_{ps}}{dt} & = & \sigma_{ps}E_{ps} - \gamma_{ps}I_{ps} - \mu_{ps}I_{ps} \\
& & + \sum_{q=1}^{n} \rho_{pqs}I_{qs} - \sum_{q=1}^{n} \rho_{qps}I_{ps} \\
\frac{dR_{ps}}{dt} & = & \gamma_{ps}I_{ps} - \mu_{ps}R_{ps} - \nu_{ps}R_{ps} \\
& & + \sum_{q=1}^{n} \rho_{pqs}R_{qs} - \sum_{q=1}^{n} \rho_{qps}R_{ps} \\
\lambda_{ps} & = & \sum_{i}^{humans,birds} \beta_{isp}I_{pi}/N_{pi}
\end{cases}
$$

```
I1=zeros(5,2);I2=zeros(5,2);I1(1,2)=10;I2(1,2)=10
E=zeros(5,2);R=E; %...
S = reshape(S,[1 ns*np]); E = reshape(E,[1 ns*np]);
I1 = reshape(I1,[1 ns*np]);
I2 = reshape(I2,[1 ns*np]);
R = reshape(R,[1 ns*np]);
[T,Y]=ode45(@rightSideAIModel,[0 tMax],[S E I1 I2 R],options); %...
function res=rightSideAIModel(t, pop)
I1=reshape(pop(2*np*ns+1:3*np*ns),[np ns]);
I2=reshape(pop(3*np*ns+1:4*np*ns),[np ns]); %...
lambda=zeros(np,ns);
for p=1:np
    for s=1:ns
        lambda(p,s)=sum( ((beta1(s,:,p).*I1(p,:))
        +(beta2(s,:,p).*I2(p,:))) ./N(p,:));
    end
end %...
deltaI1 = zeros(np,ns);
deltaI2 = zeros(np,ns);
deltaR = zeros(np,ns); %...
for s=1:ns
    deltaI1(:,s)=rho(:,:,s)*I1(:,s) - sum(rho(:,:,s))'.*I1(:,s);
    deltaI2(:,s)=rho(:,:,s)*I2(:,s) - sum(rho(:,:,s))'.*I2(:,s);
end
dSdt = mu.*N + nu.*R - lambda.*S - mu.*S + deltaS;
dEdt = lambda.*S - sigma1.*E - sigma2.*E - mu.*E + deltaE;
dI1dt = sigma1.*E - mu.*I1 - gamma1.*I1 + deltaI1;
dI2dt = sigma2.*E - gamma2.*I2 - mu.*I2 + deltaI2;
dRdt = gamma1.*I1 + gamma2.*I2 - mu.*R - nu.*R + deltaR;%...
```

Concerns are scattered

```
I1=zeros(5,2);I2=zeros(5,2);I1(1,2)=10;I2(1,2)=10
E=zeros(5,2);R=E; %...
S = reshape(S,[1 ns*np]); E = reshape(E,[1 ns*np]);
I1 = reshape(I1,[1 ns*np]);
I2 = reshape(I2,[1 ns*np]);
R = reshape(R,[1 ns*np]);
[T,Y]=ode45(@rightSideOfModel,[0 [illegible]], [S E I1 I2 R] [illegible]); %
function [illegible]=rightSideOfModel(t,po[illegible]
I1=reshape(pop(2*np*ns+1:3*np*ns),[np ns]);
I2=reshape(pop(3*np*ns+1:4*np*ns),[np ns]); %...
lambda=zeros(np,ns);
for p=1:np
    for s=1:ns
        lambda(p,s)=sum( ((beta1(s,:,p).*I1(p,:))
        +(beta2(s,:,p).*I2(p,:))) ./N(p,:));
    end
end %...
deltaI1 = zeros(np,ns);
deltaI2 = zeros(np,ns);
deltaR = zeros(np,ns); %...
for s=1:ns
    deltaI1(:,s)=rho(:,:,s)*I1(:,s) - sum(rho(:,:,s))'.*I1(:,s);
    deltaI2(:,s)=rho(:,:,s)*I2(:,s) - sum(rho(:,:,s))'.*I2(:,s);
end
dSdt = mu.*N + nu.*R — lambda.*S — mu.*S + deltaS;
dEdt = lambda.*S — sigma1.*E - sigma2.*E — mu.*E + deltaE;
dI1dt = sigma1.*E - mu.*I1 - gamma1.*I1 + deltaI1;
dI2dt = sigma2.*E - gamma2.*I2 - mu.*I2 + deltaI2;
dRdt = gamma1.*I1 + gamma2.*I2 — mu.*R — nu.*R + deltaR;%...
```

```
I1=zeros(5,2);I2=zeros(5,2);I1(1,2)=10;I2(1,2)=10
E=zeros(5,2);R=E; %...
S = reshape(S,[1 ns*np]); E = reshape(E,[1 ns*np]);
I1 = reshape(I1,[1 ns*np]);
I2 = reshape(I2,[1 ns*np]);
R = reshape(R,[1 ns*np]);
[illegible]
[illegible]
```

Concerns are scattered

```
I1=reshape(pop(2*np*ns+1:3*np*ns),[np ns]);
I2=reshape(pop(3*np*ns+1:4*np*ns),[np ns]); %...
lambda=zeros(np,ns);
for p=1:np
    for s=1:ns
        [illegible]
        +(beta2(s,:,p).*I2(p,:))) ./N(p,:));
    end
end %...
deltaI1 = zeros(np,ns);
deltaI2 = zeros(np,ns);
deltaR = zeros(np,ns); %...
for s=1:ns
    deltaI1(:,s)=rho(:,:,s)*I1(:,s) - sum(rho(:,:,s))'.*I1(:,s);
    deltaI2(:,s)=rho(:,:,s)*I2(:,s) - sum(rho(:,:,s))'.*I2(:,s);
end
dSdt = mu.*N + nu.*R — lambda.*S — mu.*S + deltaS;
dEdt = lambda.*S — sigma1.*E - sigma2.*E — mu.*E + deltaE;
dI1dt = sigma1.*E - mu.*I1 - gamma1.*I1 + deltaI1;
dI2dt = sigma2.*E - gamma2.*I2 - mu.*I2 + deltaI2;
dRdt = gamma1.*I1 + gamma2.*I2 — mu.*R — nu.*R + deltaR;%...
```

Concerns are tangled

```
I1=zeros(5,2);I2=zeros(5,2);I1(1,2)=10;I2(1,2)=10
E=zeros(5,2);R=E; %...
S = reshape(S,[1 ns*np]); E = reshape(E,[1 ns*np]);
I1 = reshape(I1,[1 ns*np]);
I2 = reshape(I2,[1 ns*np]);
R = reshape(R,[1 ns*np]);
[T,Y]=[illegible]Model[illegible], [illegible]
function [illegible]rightSideModel(t,pop[illegible]
I1=reshape(pop(2*np*ns+1:3*np*ns),[np ns]);
I2=reshape(pop(3*np*ns+1:4*np*ns),[np ns]); %...
lambda=zeros(np,ns);
for p=1:np
    for s=1:ns
        lambda(p,s)=[illegible]
        +(beta2(s,:,p).*I2(p,:))) ./N(p,:));
    end
end %...
deltaI1 = zeros(np,ns);
deltaI2 = zeros(np,ns);
[illegible] %...
for s=1:ns
    deltaI1(:,s)=rho(:,:,s)*I1(:,s) - sum(rho(:,:,s))'.*I1(:,s);
    deltaI2(:,s)=rho(:,:,s)*I2(:,s) - sum(rho(:,:,s))'.*I2(:,s);
end
dSdt = mu.*N + nu.*R - lambda.*S - mu.*S + deltaS;
dEdt = lambda.*S - sigma1.*E - sigma2.*E - mu.*E + deltaE;
dI1dt = sigma1.*E - mu.*I1 - gamma1.*I1 + deltaI1;
dI2dt = sigma2.*E - gamma2.*I2 - mu.*I2 + deltaI2;
dRdt = gamma1.*I1 + gamma2.*I2 - mu.*R - nu.*R + deltaR;%...
```

Concerns are scattered

Concerns are tangled

Not modular, not reusable

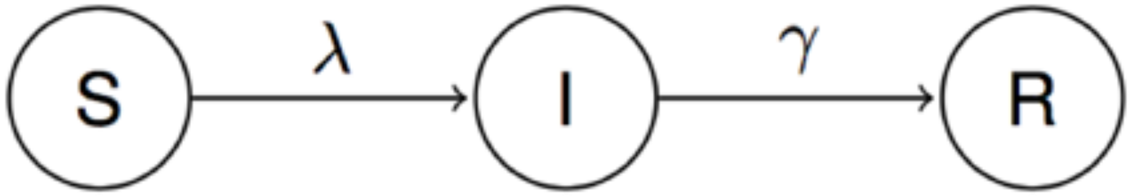
Separation of Concerns in Epidemiology Modeling

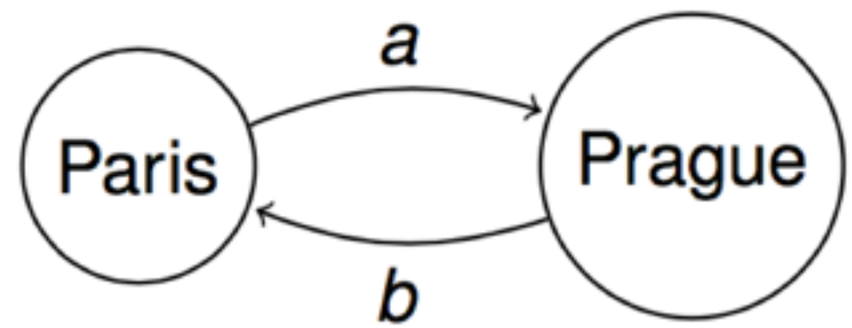
- Decompose highly-coupled monolithic models into modular concerns
  - define concerns with as few dependencies as possible
  - Combine concerns as freely as possible

Solution

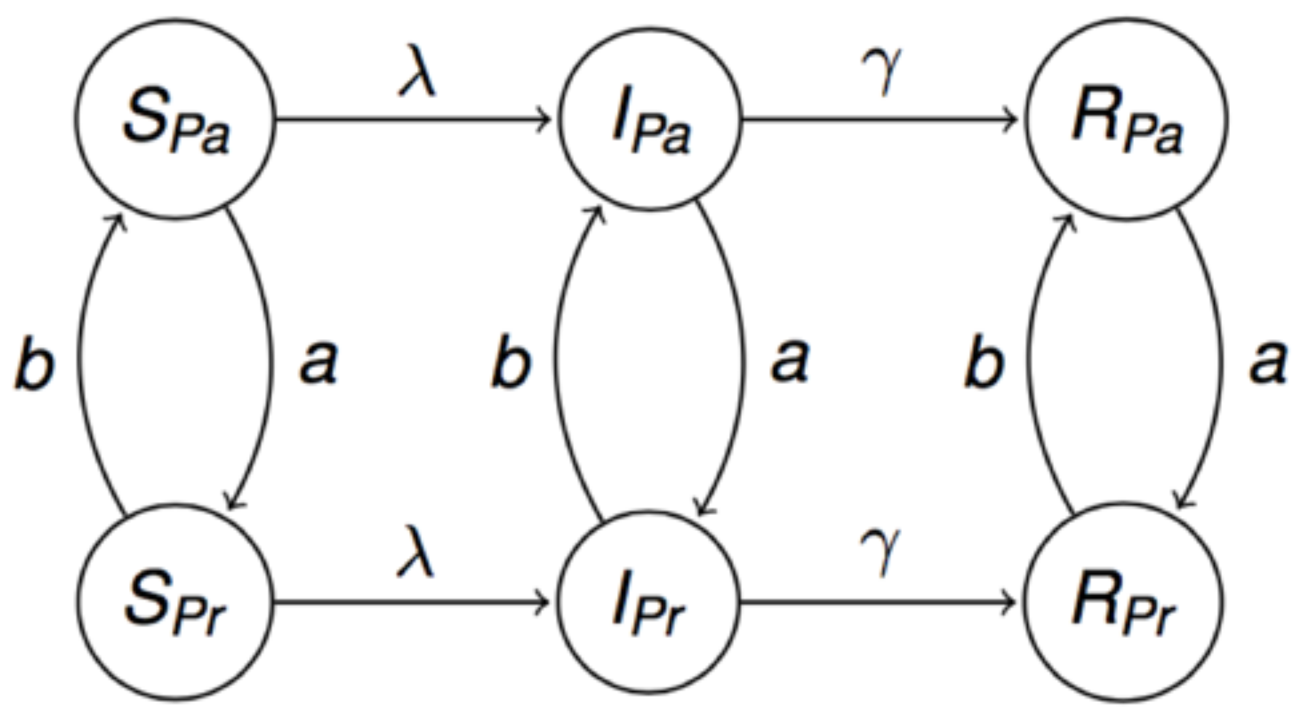
- Generic mathematical meta-model that provides abstractions to define epidemiological concerns
- Models are expressed as Stochastic Automata
- Composition operator (tensor sum)
- Transforms concerns to modify automata

Tensor Composition of a spatial concern with SIR model





$SIR \oplus Spatial$



Kendrick DSL

- Implemented the generic mathematical meta-models in Smalltalk
- Embedded DSL in Pharo
- Allows definition and composition of concerns
- https://github.com/UMMISCO/kendrick

```
KendrickModel Influenza.

Concern SIR
  attribute: #(status -> S I R);
  parameters: #(beta lambda gamma);
  lambda: #(beta*I/N);
  transitions: #(
    S -- lambda --> I.
    I -- gamma --> R.
  ).

Concern Demographical
  attribute: #(city -> Paris Prague);
  parameters: #(rho);
  transitions: #(
    Paris -- rho --> Prague.
    Prague -- rho --> Paris
  ).
```

```
Composition SIRSpatial
  model: 'Influenza';
  concern: 'Demographical';
  concern: 'SIR';
  populationSize: 25000;
  gamma: 0.233;
  rho_city: #(0.1 0.05);
  beta_city: #(0.42 0.28);
  lambda: #(beta*I_city/N sum);
  N: #(city);
  S_city: #(14490 10000);
  I_city: #(10 0).
```

Kendrick Demo

If you want to try Kendrick
download a MOOSE 6.0 image from
http://agilevisualization.com/

Participants

- Fabrice Atrevi (IFI-VN)
- Bui Thi Mai Anh (UMMISCO-VN)
- Ho Tuong Vinh (UMMISCO-VN)
- Cheikhou Oumar Ka (UMMISCO-Senegal)
- Nick Papoulias (UMMISCO-France)
- Benjamin Roche (UMMISCO-France)
- Aboubakar Sidiki (UMMISCO-Cameroon)
- Serge Stinckwich (UMMISCO-France)
- Mikal Ziane (LIP6, UPMC)

Conclusions

- Kendrick is an embedded DSL for epidemiology modeling/simulations that promote separation of concerns
- Chapter 15 in “Agile Visualisation Book”: http://agilevisualization.com/
- We rely a lot on the Pharo community: PolyMath, Roassal, PetitParser, Moose, STon, SmalltalkCI

Ongoing Work

- GPU (VirtualGPU) implementation of stochastic algorithms - Cheik Oumar Ka
- Network concerns - Aboubakar Sidiki (April-May 2016)
- Metamorphic tests of Epi models - Herman Mekontso Tchinda
- User eXperiments (in collaboration with Nick Papoulias during 2017)