## System Dynamics Modelling with Kotlin: From Hierarchical Models to Interactive Simulators

Siniša Sovilj

**Darko Etinger** 

Zlatko Sirotić

Krešimir Pripužić

Juraj Dobrila University of Pula, Faculty of Informatics HR-52100 Pula, CROATIA <u>http://hr.linkedin.com/in/ssovilj</u> <u>sinisa.sovilj@unipu.hr</u> Juraj Dobrila University of Pula, Faculty of Informatics HR-52100 Pula, CROATIA <u>darko.etinger@unipu.hr</u> Istra Tech d.o.o. HR-52100 Pula, CROATIA zlatko.sirotic@istratech.hr University of Zagreb, Faculty of Electrical Engineering and Computing HR-10000 Zagreb, CROATIA <u>kresimir.pripuzic@fer.hr</u>

Keywords: system dynamics, kotlin, modelling, complex system, hierarchical model, simulator

**Funding Source/Alternate Affiliations**: This work has been supported in part by Croatian Science Foundation under the project UIP-2017-05-9066.

**Extended Abstract:** System dynamics (SD) modelling is traditionally done by graphically using mostly proprietary software or with couple of open and free software exceptions. Building large hierarchical models with reusable SD modules is not supported well and building fast and customizable simulators for interactive learning on multiple target platforms is not trivial.

There are couple of framework/toolkit solutions that fulfill that gap using general-purpose languages (Java, Python, JavaScript) but they have their own limitations. We asked whether Kotlin, as a new, modern, statically-typed, null-safe, object-oriented and functional language can do any better and overcome limitations of other programming languages and frameworks/toolkits. Therefore, we started to develop *Kotlin SD Toolkit* which does not exist so far at our best knowledge.

We found that Kotlin as a new, programming language and our Kotlin SD Toolkit as a new tool are both suitable for modelling larger, hierarchical SD models that support modules (see Table 1) and for easier development of interactive simulators for multiple target platforms: desktop, web or mobile (see Figures 1, 2 and 3).

We also measured speed properties without any a priori code optimizations and for now Kotlin SD Toolkit on our computer needs 7 seconds to numerically integrate simple testing SD model with 1E7 time steps, which is very fast in comparison with others and can be additionally improved.

We will continue to further expand capabilities of the Kotlin SD Toolkit as an open source project and our contribution to SD community.

Steps	Kotlin code			
0) Setup	<pre>// Static properties (optional) companion object {</pre>			
	<pre>const val TOTAL_POPULATION_VALUE = 10000 // [customer]</pre>			
	<pre>const val ADVERTISING_EFFECTIVENESS_VALUE = 0.011 // [1/year]</pre>			
	<pre>const val CONTACT_RATE_VALUE = 100 // [1/year]</pre>			
	<pre>const val ADOPTION_FRACTION_VALUE = 0.015 // []</pre>			
	const val INITIAL TIME VALUE = 0 // [year]			
	const val FINAL TIME VALUE = 10 // [year]			
	const val TIME STEP VALUE = 0.25 // [year]			
	}			
1) Model	init {			
	val model = Model()			
	// Override default model properties			
	model.initialTime = INITIAL TIME VALUE			
	model.finalTime = FINAL TIME VALUE			
	model.timeStep = TIME STEP VALUE			
	model.integration = EulerIntegration()			
	model.name = "Innovation/Product Diffusion Model" // optional			
2) Entities				
- Constants	val TOTAL POPULATION = model constant ("TOTAL POPULATION")			
	val ADVERTISING EFFECTIVENESS =			
	model constant ("ADVERTISING EFFECTIVENESS")			
	CONTROL DATE = model constant ("CONTROL DATE")			
	Val CONTACT_KAIE - MODEL.CONStant ( CONTACT_KAIE )			
- Conventora	val adoption_rRACTION - model.constant( ADOPTION_rRACTION )			
- COnverters	vai adoptionrionAdvertising -			
	model.converter("adoptionFromAdvertising")			
	<pre>val adoptionFromWordOfMouth =     model converter("adoptionExemMordOfMouth")</pre>			
Steele				
- SLOCKS	<pre>val Potential_Adopters = model.stock("Potential_Adopters")</pre>			
Flows	val Adopters = model.stock("Adopters")			
- Flows	val adoptionRate = model.llow("adoptionRate")			
- modutes				
2) Triticl				
3) Initial values				
- Stocks	Potential_Adopters.initialValue = { 'TOTAL_POPULATION }			
	Adopters.initialValue = { 0.0 }			
4) Equations				
- Constants	TOTAL_POPULATION.equation = { TOTAL_POPULATION_VALUE }			
	ADVERTISING_EFFECTIVENESS.equation =			
	{ ADVERTISING_EFFECTIVENESS_VALUE }			
	CONTACT_RATE.equation = { CONTACT_RATE_VALUE }			
	ADOPTION_FRACTION.equation = { ADOPTION_FRACTION_VALUE }			
- Converters	adoptionFromAdvertising.equation =			
	{ Potential_Adopters * ADVERTISING_EFFECTIVENESS }			

Table 1. Kotlin code for Innovation/Product diffusion model (also known as Bass diffusion model).

	adoptionFromWordOfMouth.equation =				
	{ CONTACT_RATE * ADOPTION_FRACTION *				
	Potential_Adopters * Adopters / TOTAL_POPULATION }				
- Stocks	Potential_Adopters.equation = { - adoptionRate }				
	Adopters.equation = { adoptionRate }				
- Flows	adoptionRate.equation =				
	{ adoptionFromAdvertising + adoptionFromWordOfMouth }				
- Modules					
5) Simulation	val simulation = Simulation(model)				
6) Outputs	simulation.outputs {				
- Text	CsvExporter("output.csv", ";"))				
- Image	<pre>PngExporter("chart.png"))</pre>				
- Desktop	WinSimulator()				
- Web	WebSimulator()				
- Mobile	MobSimulator()				
	}				
7) Run	simulation.run()				
	}				



Figure 1. Desktop Simulator – an interactive simulation environment of the Bass diffusion model. The interactive simulation environment is a desktop application window and is automatically generated based on entities' type (we can change constants interactively, and we can enable or disable plots of any model's entities).

38th International Conference of the System Dynamics Society, Bergen, Norway, July 19-23, 2020

$\rightarrow$ C $\triangle$ 0 localho	jocanostaueu x T						
	st:8080						
		WebSimulator					
INPUT (parameters):		MODEL (backend):	OUTPUT (graph):				
CONTACT RATE	100		12.5k				
	100	Adapters		- Adopters			
ADVERTISING_EFFECTIVENESS	0,011	Adopters		<ul> <li>CONTACT_RATE</li> <li>adoptionFromAdvertising</li> </ul>			
		adoptionRafe	10k	→ ADVERTISING_EFFECTIVENESS			
TOTAL_POPULATION	10000		$\mathbf{X}$	Potential_Adopters			
ADOPTION ERACTION			7.5k	→ adoptionRate → adoptionEcomWordOfMourth			
APOCHON_RACTION	0,015	adoptionPromMdvertising adoptionPremWordOMouth	e	- TOTAL_POPULATION			
			Aalu	➡ ADOPTION_FRACTION			
		ADVENTISING EPPECTIVENESS CONTACT NATE ADOPTION PRACTION	5k				
			Δ				
			$\wedge$				
			2.5k				
			0	_			
			0 5	10			
			Time				
CONTROL (ciguilation):			OUTPUT (coreadcheat))				
CONTROL (simulation):			OUTPUT (spreadsheet):				
CONTROL (simulation):			OUTPUT (spreadsheet):	B C D E F			
CONTROL (simulation): Simulation Settings			OUTPUT (spreadsheet):	B C D E F 0.00 0.25 0.50 0.75 1.00			
CONTROL (simulation): Simulation Settings Run Simulation			OUTPUT (spreadsheet): A 1 Time (month) 2 Adopters []	B         C         D         E         F           0.00         0.25         0.50         0.75         1.00           0.00         27.50         65.21         116.82         187.30			
CONTROL (simulation): Simulation Settings Run Simulation			OUTPUT (spreadsheet): A 1 Time (month) 2 Adopters [] 3 CONTACT, RATE []	B         C         D         E         F           0.00         0.25         0.50         0.75         1.00           0.00         2.750         65.21         116.82         187.30           100.01         100.00         100.00         100.00         100.00			
CONTROL (simulation): Simulation Settings Run Simulation			OUTPUT (spreadsheet): A 1 Time (month) 2 Adopters [] 3 CONTACT_RATE [] 4 adoptionFromAdvertising [] 5 AdoptionAdvertising []	B         C         D         E         F           0.00         0.25         0.50         0.75         1.00           0.00         75.00         65.21         11.682         187.01           0.00         10.00         10.00         10.00         10.00           10.01         10.00         10.01         10.00         10.01           0.00         0.01         0.00         0.01         0.01         0.01			
CONTROL (simulation): Simulation Settings Run Simulation			A           1         Time (month)           2         Adopters []           3         CONTACT_RATE []           4         adoptonFromAdventising []           5         ADVERTISING_EFFECTIVENESS []           6         Potential Adopters []	B         C         D         E         F           0.00         0.25         0.50         0.75         1.00           0.00         2750         65.21         116.82         187.30           10.04         100.00         100.00         100.00         100.00           10.04         109.70         109.28         108.71         107.94           0.01         0.01         0.01         0.01         0.01         0.01			
CONTROL (simulation): Simulation Settings Run Simulation			OUTPUT (spreadsheet):           A           1         Time (month)           2         Adoptent []           3         CONTACT_RATE []           4         adoptoniformAdvertising []           5         ADVERTISING_EFFECTIVENISS []           6         Potential_Adoptens []           7         adoptionifate []	B         C         D         E         F           0.00         0.25         0.52         0.05         1.00           0.00         27.50         65.21         11.68.2         187.30           0.00         10.00         100.00         100.00         100.00           0.00         100.00         100.00         100.01         0.01         0.01           0.01			
CONTROL (simulation): Simulation Settings Run Simulation			OUTPUT (spreadsheet):           A           1         Time (month)           2         Adopters (]           3         CONTACT_RATE (]           4         adoptionFormAdvertising []           5         ADVERTISING_EFFECTIVENESS []           6         Potential_Adopters (]           7         adoptionFormVordOfMouth []	B         C         D         E         F           0.00         0.25         0.50         0.75         1.00           0.00         75.00         65.21         11.82         187.30           0.00         10.00         10.00         10.00         10.00           10.01         100.00         10.00         10.01         0.01           0.01         0.01         0.01         0.01         0.01           10.02         9.72.5(9.934.7)         9.83.11         9.812.70           10.01         5.03         2.64.6         281.09         38.35           0.00         4.14         9.717         17.31         275.69			
CONTROL (simulation): Simulation Settings Run Simulation			OUTPUT (spreadsheet):           A           1         Time [month]           2         Adopters []           3         CONTACT_RATE []           4         adoptionFromAdventsing []           5         ADVERTISING_EFFECTIVENESS []           6         Potential_Adopters []           7         adoptionFrame           8         adoptionFrame           9         TOTAL_POPULATION []	B         C         D         E         F           0.00         0.25         0.50         0.75         1.00           0.00         75.05         52.11         1.68.21         8.08           10.04         10.05.01         10.00.01         10.00.01         10.00.01           0.01         0.01         0.01         0.01         0.01         0.01           10.04         9.07.25         9.34.79         9.88.14         9.82.70           10.04         9.07.75         9.73.44         7.98.81         9.82.70           10.04         9.07.75         9.73.47         9.73.93         7.99.94           10.04         9.77.75         1.73.94         7.99.94         9.98.14           0.00         4.1.44         9.71.77         17.31.94         7.59.94           0.01         1.00.02         0.00.01         1.00.00			

Figure 2. Web Simulator – an interactive simulation environment of the Bass diffusion model. The interactive simulation environment is a web application rendered in any Internet browser and is automatically generated based on entities' type. We can change model constants, simulation settings and display the model diagram as an option.



Figure 3. Mobile Simulator – automatically generated interactive simulation environment for Bass diffusion model. The interactive simulation environment is an Android mobile application. We can change model constants and re-run the simulation on a mobile phone.

## References

- [1] Simantics Team, February 2018. [Online]. Available: https://www.simantics.org/end\_user\_wiki/index.php/Simantics\_System\_Dynamics#Installation\_Instructions.
- [2] F. Lardinois, "Kotlin is now Google's preferred language for Android app development," May 2019. [Online].
- [3] S. Drost and M. Stein, "System Dynamics Java-Framework," February 2017. [Online]. Available: https://github.com/matthiasstein/SystemDynamics-Framework.
- [4] D. Schroeck, "Business Prototyping Toolkit for Python (BPTK-Py)," January 2020. [Online]. Available: https://pypi.org/project/BPTK-Py/.
- [5] B. Powers, "sd.js. open source SD," Spetember 2019. [Online]. Available: https://sdlabs.io/.
- [6] B. Powers, "Simulation Speed: Fast," September 2019. [Online]. Available: https://sdlabs.io/speed/.