User manual

System requirement

- Mono 2.0 and above or .Net Framework 2.0 and above
- GTK# 2.12.0 and above
- library monoclips
 - For Linux it is nessesary to make symbolic link to libCLIPSLib.so from one of the standart paths, listed in the LD_LIBRARY_PATH, or to add current location of the file to the variable.
 - For Windows it is enough to place CLIPSLib.dll in the same directory as
 CLIPSNet.dll, or to place it to Windows/SYSTEM32 folder.

Run an example

To run a machine learning platform, one should run file MLPlatform.GtkInterface.exe. Once it is loaded, to see an example, user have to open list of platform's components (Figure 1)

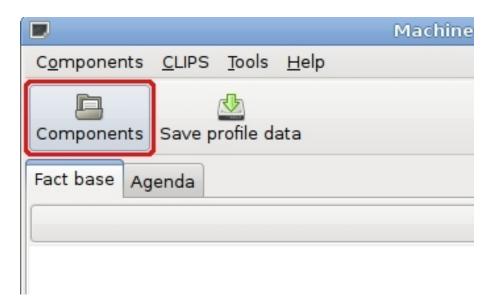


Figure 1: Open components list

For the first launch, the list is empty so assembleys that contains components should be loaded (Figure 2).

🔲 Pl	atform components list	
Add	Description	
	Find more	
	rind more	

Figure 2: Empty list of components

To run an example files marked on Figure 3 should be added (JSM.dll and WeaponTest.dll).

Имя	∨ Изменён
🛅 add-onns	
ClipsConsole.exe	12:45
CLIPSNet.dll	Четверг
♦ CommonUIStaff.dll	Четверг
Interface.exe	29.06.2009
(v jsm.dll)	12:45
MLPlatform.Base.dll	Вчера в 18:46
MLPlatform.Gtkinterface.exe	12:45
PlatformEngine.dll	Вчера в 18:46
SexDeterminer.dll	12:07
WeaponTest.dll	Четверг

Figure 3: Example files

As these assembleys are opened, components are loaded but not registered in the system. The checkboxes should be checked to register the components (Figure 4).

Add	Description
\checkmark	🚓 V.Finn's JSM-method realization
\checkmark	WeaponTest.Peculiarity
	📊 WeaponTest.StringStorage
	WeaponTest.Text

Figure 4: Registering components

After registering components, rules should be loaded (Figure 5).

C <u>o</u> mponents	<u>C</u> LIPS	<u>T</u> ools	<u>H</u> elp
Components Fact base Ag	Clea Load Rese Run		ta
	Sett Drib	-	

Figure 5: Load education rules

Rules for this example contains calls of external (for CLIPS) functions, and if one loads rules before registering an education engine, CLIPS would fail to parse the rules file. For this example JSM by Finn.clp file should be used (Figure 6).

ml	ect CLIPS rules			×
n.clp				
\sim	Имя		Изменён	
	📄 JSM by Finn.clp		13:20	
= = ! ть				≡
		Cancel	Open	

Figure 6: Education rules

After successfull load of rules, CLIPS should be reseted (Figure 7).

C <u>o</u> mponents	<u>C</u> LIPS	<u>T</u> ools	<u>H</u> elp
	Clea	r	
Components	Load	d	ta
	Rese	et	
Fact base Ag	Run		
	Settings		_
	Drib	ble	

Figure 7: Reseting CLIPS

After CLIPS is successfully reseted, the education process can be started (Figure 8).

C <u>o</u> mponents	<u>C</u> LIPS	<u>T</u> ools	<u>H</u> elp
Ē	Clea	r	
Components	Load	d	ta
	Rese	et	
Fact base Ag	Run		
	Sett	ings	
f-0 (initial-fac	Drib	ble	
f-1 (Fact (dat	a <poir< td=""><td>nter-5-0</td><td>×1>) (d</td></poir<>	nter-5-0	×1>) (d
f-2 (Fact (dat	a <poir< td=""><td>nter-5-0</td><td>x2>) (d</td></poir<>	nter-5-0	x2>) (d

Figure 8: Starting learning process

During learning process (as before it and after), user can change the view to see facts of certain template by chosing that template's name in the form (Figure 9).

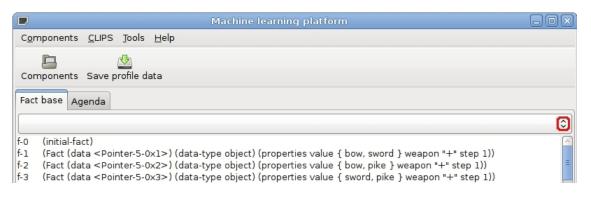


Figure 9: Show certain facts only

During learning process (as before it and after), user can switch the view, to see the agenda (Figure 10).

Co	mponents	<u>C</u> LIPS	<u>T</u> ools	<u>H</u> elp			
	Ē						
Co	mponents	Save p	rofile da	ata			
Fac	ct base 🔼	enda					
0	reason-se	earch: f-3	30,f-29,	*			
0	reason-se reason-se						
-		earch: f-3	30,f-28,	*			
0	reason-se	earch: f-3 earch: f-3	30,f-28, 30,f-27,	*			
0	reason-se reason-se	earch: f-3 earch: f-3 earch: f-3	30,f-28, 30,f-27, 30,f-26,	* *			

Figure 10: Show agenda

As it is with facts view, user can change the view of agenda to see activations of a certain rule, by choosing the rule name in the form (Figure 11).

	Machine learning platform	
Components <u>C</u> LIPS <u>T</u> ools <u>H</u> elp		
Components Save profile data		
		0
0 reason-search: f-30,f-29,* 0 reason-search: f-30,f-28,* 0 reason-search: f-30,f-27,*		

Figure 11: Show activations of a certain rule

After education process is complete, profiling data can be saved to an XML file (Figure 12).

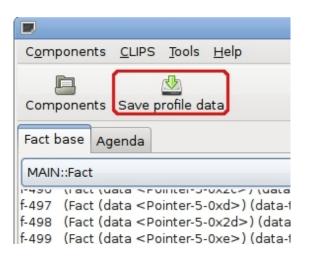


Figure 12: Save profiling data

Or it can be viewd in a form (Figure 13, 14).

			Mach
C <u>o</u> mponent	s <u>C</u> LIPS	Tools	<u>H</u> elp
Component	s Save p		w profiler data
Fact base 🖌	Agenda		
MAIN::Fact			
			0xd>) (data-type of

Figure 13: Show profiling data

	Profiler data viewer	×
Ē		
Открыть		
×		
similarity calls	864	
is-in calls	213	
change-property calls	24	
get-properties calls	18	
to-string calls	24	
cpu time (ticks)	150322874	
memory used	241664	
rules fired	888	
cpu time (ms)	15032,2874	
mem-requests	7811	
mem-used	1147022	
Components		
Clips environment settings		
▽ Semantic data		
initial facts	30	
total facts generated	484	
significant facts generated	48	
inference steps	19	
hypothesys generated	18	
		_

Figure 14: View profiling data