

Robotics Systems Commons - Enhancement #2184

improve logging system to nicely play with log4cxx

02/23/2015 03:44 AM - R. Haschke

| | | | |
|------------------------|----------|------------------------|------------|
| Status: | Feedback | Start date: | 02/23/2015 |
| Priority: | Normal | Due date: | |
| Assignee: | | % Done: | 50% |
| Category: | Logging | Estimated time: | 0.00 hour |
| Target version: | rsb-0.18 | | |

Description

Current implementation of logging doesn't play nice with log4cxx, i.e. a LoggingSystem that already provides hierarchical loggers.

In this case explicitly setting the log levels down the hierarchy, overrules any settings made in log4cxx's own config files.

Hence, I modified the code along the following guide lines:

- loggers are created lazily on demand
 - The LoggerTreeNode maintains the hierarchy (and explicitly set log levels).
 - However, Loggers themselves will be only created by LoggerFactor::getLogger().
- LoggingSystem::createLogger expects second argument now with desired log level.
 - LoggingSystems like log4cxx want to ignore this argument, but rely on their hierarchy.
- bool LoggingSystem::needsRecursiveLevelSetting() is used to choose
 - recursive TreeLevelUpdater for non-hierarchic LoggingSystems
 - SimpleLevelUpdater for hierarchic systems
- Do not use shared_ptrs for LoggerTreeNode::assignedLevel.
 - This was only used to know whether a level was explicitly set. Use a bool instead.
- Simplified LoggerTreeNode::Visitor
 - parentLevel argument was never used --> removed, can be retrieved by node->getLevel()
 - visiting includes called node -> reduces code duplication

Consequences:

As long as no log levels are set explicitly by rsc means, the

LoggingSystem can continue to use its own configuration mechanism.

Calling setLevel() overwrites these settings, but only for the current node.

History

#1 - 02/23/2015 03:51 AM - R. Haschke

- Status changed from New to Feedback
- % Done changed from 0 to 90

The proposed implementation is in branch enhancement-2184

#2 - 03/05/2015 01:42 PM - J. Wienke

- Description updated
- Status changed from Feedback to In Progress

#3 - 03/05/2015 01:43 PM - J. Wienke

- Description updated

#4 - 03/05/2015 03:23 PM - J. Wienke

- % Done changed from 90 to 50

I am not exactly sure about the consequences of this as this would effectively result in two different places that can be used to define logger levels. The current logging API was designed so that implementors of LoggingSystem instances never need to think about the hierarchy of loggers. This was all performed externally and LoggingSystem@s merely serve as output devices (and formats). Therefore, the intended separation of concerns is that the RSC configuration mechanism defines the levels and their hierarchy and @LoggingSystem instances deal with the question of how to output logging systems to somewhere.

With your proposed changes, this separation of concerns is washed out since now also logging systems can and define the levels and their hierarchy and I suspect that there are many corner cases and conceptual things that can be overlooked when doing so. Moreover, it might be harder to understand for users which part effectively defines the level of a logger. One corner case we already saw in your code is that it is now possible to define levels with log4cxx that are not visible in RSC at all, because LoggerProxy::getLevel never asks the real Logger implementation for the level. This way, logging statements might even be unintentionally missed because client code can make the generation of strings for logging dependent on the effective level of a logger to prevent unnecessary computational load.

But to prevent all these complicated details, we should first find out why this approach actually necessary at all: Why do you think is it necessary to also define effective levels with log4cxx configuration files? Or stated the other way around: Isn't it sufficient to define levels via the RSC config and only define output formats via the log4cxx configuration?

Apart from that I am fine with 240b4175 and have committed this one on master

#5 - 03/24/2015 10:10 AM - R. Haschke

Sorry, I haven't seen your update on this. Let's meet in person to discuss these enhancements.

#6 - 09/30/2015 04:37 PM - J. Wienke

- Status changed from *In Progress* to *Feedback*
- Assignee deleted (J. Wienke)

#7 - 09/30/2015 04:37 PM - J. Wienke

- Target version changed from *0.7* to *rsb-0.14*

#8 - 06/08/2016 09:03 PM - J. Moringen

- Target version changed from *rsb-0.14* to *rsb-0.15*

#9 - 11/07/2016 07:24 PM - J. Moringen

- Target version changed from *rsb-0.15* to *rsb-0.16*

#10 - 04/10/2017 12:31 PM - J. Moringen

- Target version changed from *rsb-0.16* to *rsb-0.17*

#11 - 10/16/2017 09:23 PM - J. Moringen

- Target version changed from *rsb-0.17* to *rsb-0.18*