The fact that an imperfect version of the library was in active use in KDE served as a motivating factor for quickly porting it to Qt4 when that became usable in beta versions. Thus it was available rather early in the KDE 4 development cycle, if only in a secondary module and not yet as part of KDELibs. Over the course of two years, the author gave a number of presentations on the library, presenting an ever-easier and more complete API as he kept improving it. It was, one could say, a solution looking for a problem. The majority of developers working on KDE needed time to realize that this library was not only academic, but could improve their software significantly given that they make the investment in taking a step back and rethinking some of their architectural structures. There was no concrete need by a group of developers driving the library's progress; it was progressed by an individual because of his belief in the growing relevance of the problem and the importance of making available a good solution for the KDE 4 platform. Especially following the 2005 Akademy conference in Malaga, Spain, more programs started to use ThreadWeaver, including KOffice and KDevelop, which created enough momentum for it to be integrated into the main KDE 4 set of libraries.

ThreadWeaver represents the case of an alternative solution to a problem that once it had matured to critical point and once the author and the prospective user community agreed that the time had come for it to be adopted by developers in their projects, it was quickly promoted to a cornerstone of KDE 4. After that, the attitudes of community members changed from mild amusement to appreciation and recognition of the effort that had gone into it. This is an example of how efficient this community can be at making technical decisions and adapting its stance when an approach proves itself in practice. There can be no doubt that ThreadWeaver is a much better library now than it would have been if it not taken three to four years of rubbing up against the KDE project until its inclusion. And this includes the rogue premature adoption by the KMail developers. There is also little doubt that applications written for KDE 4 can deal with concurrency a lot better and thus provide a better experience to their users, because it succeeded in the end.

ThreadWeaver will be extended mostly by adding GUI components to visually represent queue activity, and by including more predefined job classes. Another idea is the integration with operating system IPC mechanisms (to allow for host-global resource restrictions, for example), but those are hindered by the requirement to be cross-platform. The approaches taken by the different operating systems are very diverse. With the public availability of the KDE 4 line, it became visible to a large audience. Since ThreadWeaver is not really KDE-specific, the question of where to go next (Freedesktop.org?) is in the air. For now, the focus remains to provide developers of applications and the desktop with a reliable scheduler for concurrency.