ABSTRACT Designing user interfaces for Symbolic Computation tools like Maple, Mathematica, Reduce, etc., implies solving some challenging difficulties, including the display and manipulation of mathematical formulas, efficient and user-friendly manipulation of possibly large expressions, deep extensibility of the user interface, etc. This also implies dealing with some complex communication issues in order to efficiently integrate under a common user interface a set of tools from different origins running across a distributed architecture. This includes transparent management of remote computations, hiding command language variations, and programmability of the whole environment so that one can solve complex problems by sequentially and/or concurrently applying any needed tools. Very few of these problems are addressed by the user interfaces of commercially available packages. To improve user interfaces and integration of Symbolic Computation tools, a prototyping system named CAS/PI has been implemented. Its main characteristics are to be highly flexible and extensible, and to be based on pre-existing software engineering technologies. Using CAS/PI, each of the previously stated problem may be solved in a generic way, allowing further experimentations and optimizations.