

DARWIN/PRICE Integration Plan

William Henry Jones

December 7, 1998

1 Abstract

A persistent desire expressed on various levels is to integrate the information contained/generated within the PRICE architecture into the DARWIN data distribution sphere. The following note discusses possibilities in this area.

Note: This document is not intended as an introduction to either DARWIN or PRICE.

2 Synopsis

Instead of adapting PRICE to DARWIN, DARWIN should be adapted to PRICE.

3 What They Are In a Nutshell

3.1 DARWIN

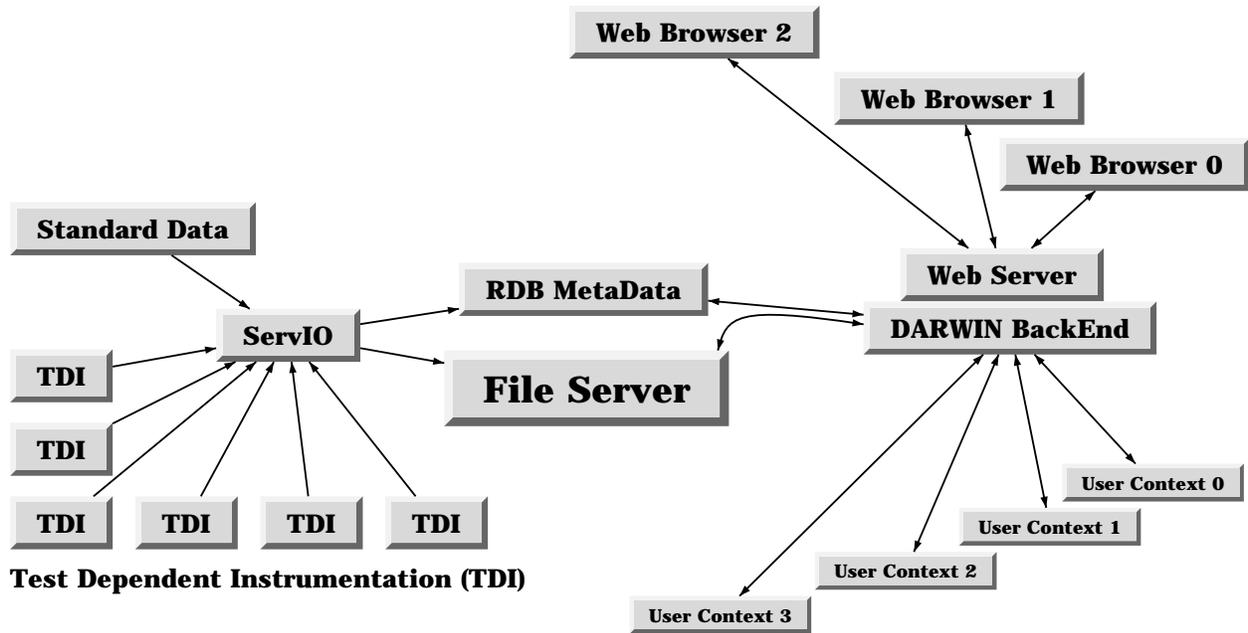
DARWIN is a sophisticated engineering browser/server (see Figure 1) of static data archives, those archives consisting principally of airframe windtunnel data. New data originates in various interfaced facility and analytic modules (depicted as the Standard Data source as well as the various Test Dependent Instrumentation (TDI) sources) and enters the environment by special communications between the originating source and the ServIO component. The actual data often spends its life resident on one or another file server while

meta data identifying the data is held in a relational data base (RDB).

DARWIN's ability to organize data into sets and add engineering annotations (and the like) to that data is done within the context of the DARWIN user, not within the context of the data. That is, should a user make a notation such as 'Reading 22 looks like a loose cable', that notation is made in an entry about Reading 22 in the user's 'Log Book' and appears only to that user. It is not attached to Reading 22 wherever Reading 22 is maintained and, consequently, cannot be discovered by other users consuming Reading 22.

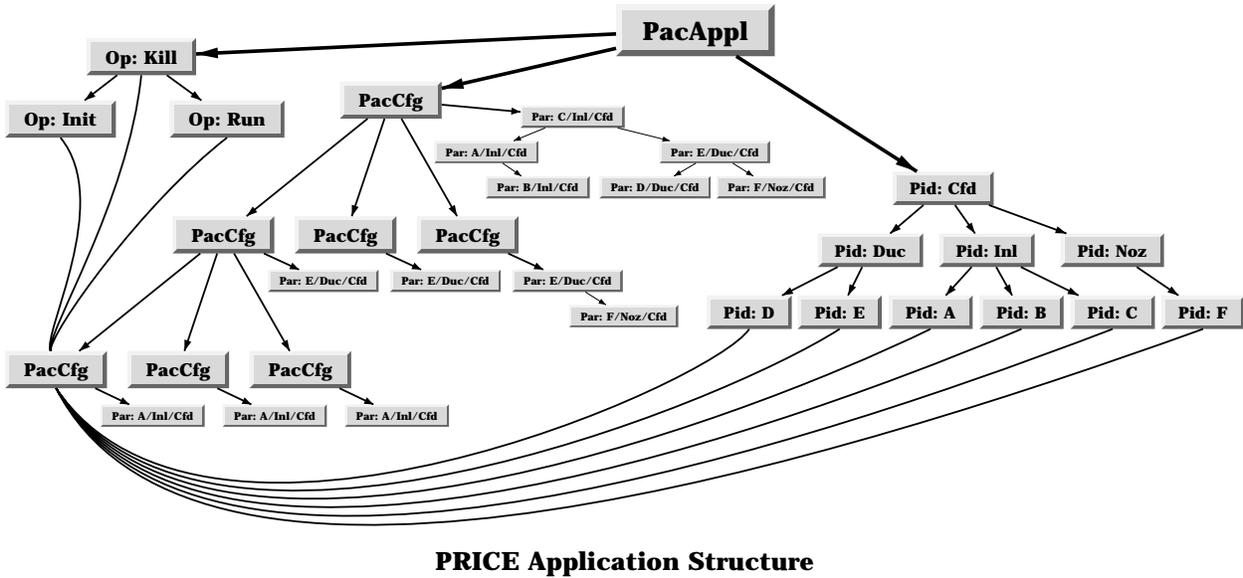
3.2 PRICE

PRICE is an integrated engineering application operational environment consisting of one (or more) interaction tools and an application architecture (see Figure 2) behind which is wrapped various engineering applications, some of which may be experimental data applications. Conceptually, data originates within PRICE; a researcher examines the current state of his work, decides the result is not quite what he wants, tiggles a parameter a little bit (often causing the creation of a new data configuration object), tells the application to reanalyze the situation with the modified parameter, and new data results within the PRICE framework. (In this sense, experimental data applications represent a very modest application of the PRICE concept.) Thus, PRICE is much more an operational working environment than



Test Dependent Instrumentation (TDI)

Figure 1: DARWIN Architecture as I Understand It



PRICE Application Structure

Figure 2: PRICE Application Architecture

a browser/server of (essentially) static data archives.

PRICE is paperless, permitting the capture of all relevant forms of documentation (notes, sketches, and the like); however, in PRICE such information is not associated with the user in some private context, but with the work itself, being directly attached at whatever relevant point to the actual object encapsulating the structure of the work and, thus, being discovered by other consumers of the work.

4 Difficulties Between DARWIN and PRICE

DARWIN is an admirable accomplishment that gives every appearance of meeting the design goals set for it while operating within the constraints that existed during the time of its development. To say that there are 'Difficulties with DARWIN' is not to impugn its achievement; however, certain of the design choices of DARWIN make assimilation of PRICE information content less than easy.

The difficulties, for the most part, originate from the fact that DARWIN was conceived of within the airframe/windtunnel test environment. Indeed its explicit purpose was to make easily available the results of experimental and numerical windtunnel testing. As such, PRICE applications find difficulties with simply emitting data into ServIO, as though PRICE were a collection of TDIs. Key among these difficulties are the following.

1. DARWIN meta-data records (the means by which actual data is identified and located) are formulated with parameters of consistent relevance to the airframe/windtunnel community (angle of attack, angle of yaw, dynamic pressure, coefficients of lift, drag, and the like, and so on). Only a small subset of these parameters are ever relevant to the propulsion community and, for a large

majority of the time, are either not relevant at all or are completely redefined in situation-specific use.

2. Complementary to the above, DARWIN meta-data records are largely absent of the great agglomeration of parameters encompassed by the total range of propulsion testing and analysis (low spool speed, high spool speed, nozzle exit area, fuel flow, turbine inlet temperature, mass flow, engine pressure ratio, and so on). It should be noted that, even if DARWIN were amended to accommodate all such parameters (presuming that a comprehensive list could be generated) in its fixed meta-data records, those meta-data records would then be sparsely populated as a matter of custom, leading to a great deal of effort sorting meaningless, default field entries.
3. The DARWIN fixed organization of data into Tests, Configurations, and Points (see Figure 3) does not accommodate the far more flexible n-ary tree of PRICE data configurations, principally in that the depth of the DARWIN arrangement is limited to 3 while the PRICE arrangement attains unto a practical infinity.
4. The DARWIN data organization constrains all configurations, points, and files to strictly filial relationships while the PRICE scheme adds a useful ancestral relationship. Further, the PRICE method of data identification allows the user to browse available parameters in a structured manner while the flat, filial presentation of DARWIN simply presents all parameters at once.

There are, of course, certain difficulties with PRICE.

1. PRICE, as a project, is not nearly as mature as DARWIN. At this time PRICE has a firm understanding of its project/programming

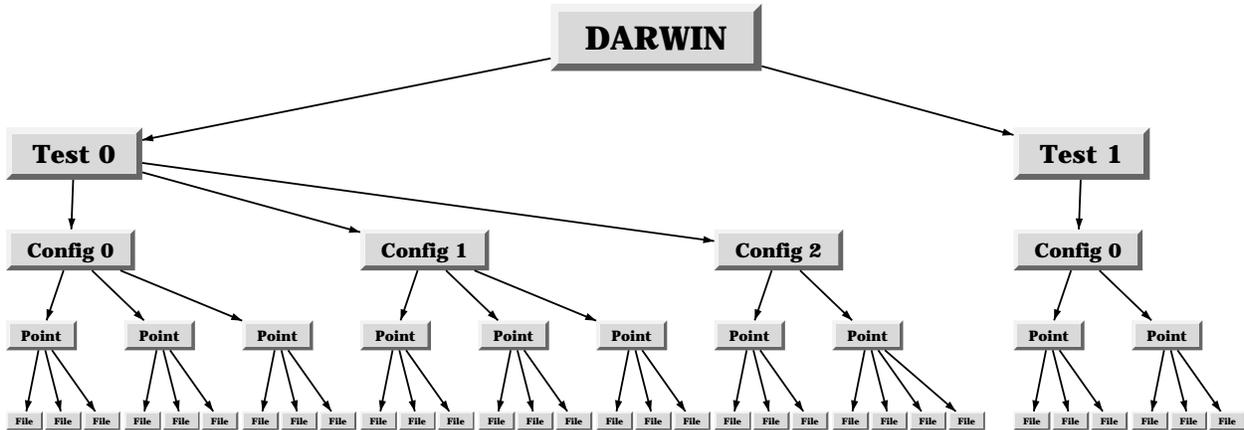


Figure 3: DARWIN Meta-Data Record Hierarchy

architecture; however, implementation in the exploratory C++ environment is, at most, well begun and implementation in the final, net-capable, CORBA environment is only just beginning.

2. No real application has yet been comprehensively 'wrapped' in the PRICE architecture and no researcher is using PRICE tools in day-to-day work. Thus, many expectations as to how PRICE will be remain, at this point, only expectations.

(The only application currently 'wrapped' by PRICE is an inlet unstart experiment for the High Speed Research office; however, due to C++ environmental limitations in opposition to the sheer volume of data acquired in testing, this application of PRICE can be considered little more than a demonstration/test-bed effort.)

It should be noted that PRICE's faults are faults of immaturity, rather than faults of inflexibility. In time, PRICE will be, of course, perfect in every way. In time.

5 What Could Be (But Shouldn't Be) Done

A first possibility (see Figure 4) for DARWIN/PRICE integration would be to teach such PRICE applications as seem appropriate to emit, on command, data into the DARWIN environment through pseudo-TDI and pseudo-Standard Data modules. The selection of TDI and Standard Data modules used would be a choice of each such adapted application. This would be quite doable within the flexibility of the PRICE Application Architecture and would, in a short-term way, meet the goal of DARWIN/PRICE integration.

The following philosophical difficulties can be seen with such an effort.

1. Such an adaptation of PRICE to DARWIN would be done on an application by application basis, as appropriate to each application willing, able, and desiring to participate in the DARWIN expository world. The PRICE Application Architecture allows neither the specificity of always conforming to such a standard nor the infrastructural knowledge to do this on an automatic ba-

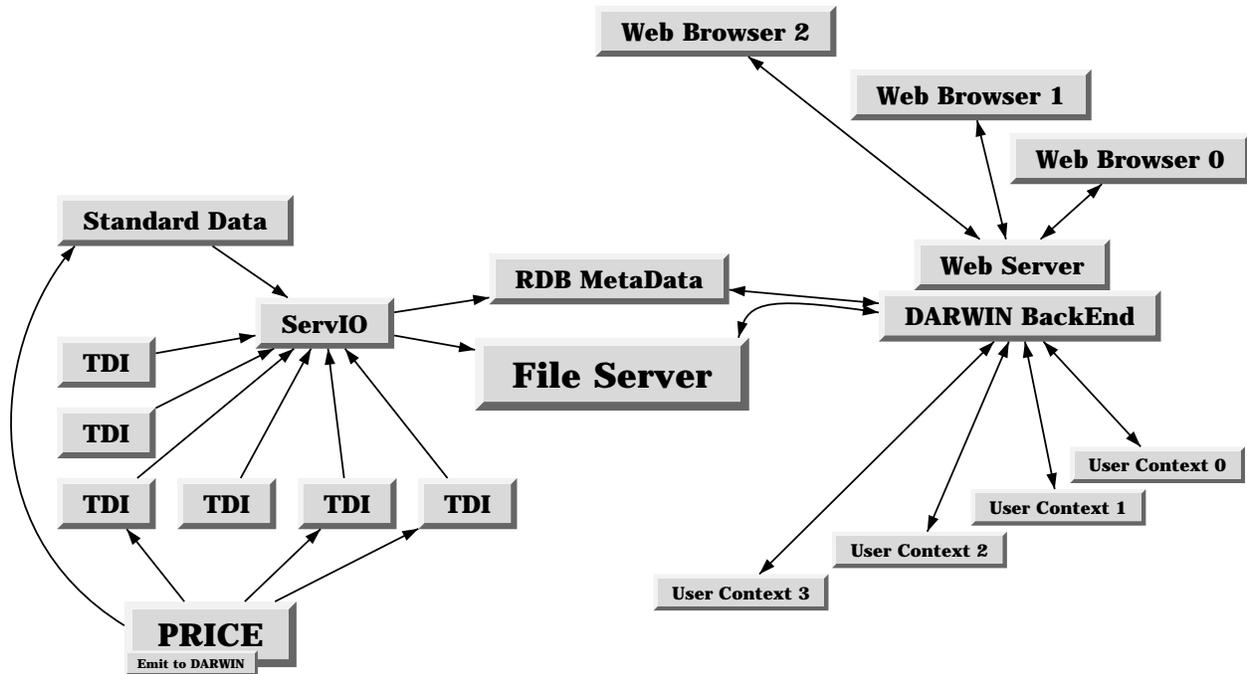


Figure 4: A Poor Choice for DARWIN/PRICE Integration

sis.

2. By making adaptation to DARWIN an application-by-application burden, the probability of DARWIN service utilization is measurably decreased while the cost of DARWIN service utilization is made linear with PRICE application. It is clearly far more desirable to make the DARWIN/PRICE connection a one-time cost rather than an n-time cost.
3. PRICE is “growing up” in a propulsion-development environment. As a consequence, few of its wrapped applications will find much point in emitting data records into the windtunnel-oriented environment of DARWIN. In application by projects at ARC and LaRC, PRICE-wrapped applications might well make good use of DAR-

WIN services; however, should PRICE blossom beyond into the wider fields of engineering development (for instance space propulsion and vehicle development, automotive design development, and the nearly endless engineering fields beyond) few of these are likely to find windtunnel orientation an inviting adjunct. Thus, adapting PRICE to DARWIN would give such effort to a limited window of useful result.

4. There appears to be no mechanism for the import of the paperless aspects of PRICE into the paperless aspects of DARWIN. The principal difficulty is that DARWIN's content is in the context of a user while PRICE's content is in the context of the project which many users may be interacting with. It is possible that PRICE paperlessness could be adapted to become DAR-

WIN data records.

It should be said, though, that DARWIN's user-context concept is not something to be discarded. There is clearly a difference between do-ers and consumers. PRICE treats this poorly through access controls which can deny "consumers" the ability to add notations. PRICE has no ability to allow a consumer to keep private notes.

6 What Makes More Sense

The one key point of difficulty is that DARWIN uses a single, relational database, fixed-record format, known-field view of access to the world of data. PRICE does not support such a view and to require adaptation of PRICE to match DARWIN expectations is to make available to and through DARWIN only a narrow slice of what PRICE may ultimately do.

The obvious choice is to reverse the process: instead of adapting PRICE to the limited data-world view of DARWIN, adapt DARWIN to the limitless data-world view of PRICE. Here, the data repositories of PRICE-wrapped application instances would be used by DARWIN either as an addition or a replacement to the relational database server currently in place.

If such an adaptation of roles were accomplished, a valuable two-phase relationship would result: PRICE would be the working environment of researchers, developers, and the like, while DARWIN would become the knowledge consumer's browser of all such work. DARWIN would then be poised to follow PRICE application into propulsion, space technology, automotive, applicances, civil engineering, indeed all manner of areas in which the basic engineering cycle of think, figure, change, think, figure, change is applied.

There are a few points to be made with regard to this proposal.

1. It is quite likely that a substantial reworking of DARWIN (DARWIN 3?) in the area

of data acquisition and expectation will be required. On the surface, it would seem possible that the new may be blended with the old so that existing DARWIN archives need not be lost.

Perhaps a data access application programming interface (API) could be defined for DARWIN's needs and interposed between the DARWIN BackEnd and the actual repositories of data. Individual adapters in this API would translate the standard DARWIN data interrogatories into forms or actions appropriate to the adapted data repository form. Thus, DARWIN could be a common data consumer of many sources, not just DARWIN Classic and PRICE. For instance, this could allow DARWIN to be a consumer of the UniGraphics complex currently gaining popularity in the automotive industry. In this way, DARWIN could become a consuming portal onto a vast array of engineering information forms.

2. Having robbed DARWIN of its known-field outlook, it then falls to PRICE to provide a mechanism by which DARWIN, having obtained some sort of data item from a PRICE repository, can then ascertain just what sort of thing the data item is in order to do appropriate things with the item. Fortunately, an appropriate expectation, if not implementation, is already in place.

PRICE derives all data objects from a common base class. Derivative classes then specialize the nature of the item, for instance to scalars, vectors, matrices, and the like, thence on to scalar longs, vector doubles, and so forth. Such specialization is expected to continue; a scalar double may eventually be specialized to a scalar double free stream Mach number. (The CGNS standard seems a highly appropriate source of definitions for such specializations in the Computational Fluid Mechanics field.) By using such specialization, the

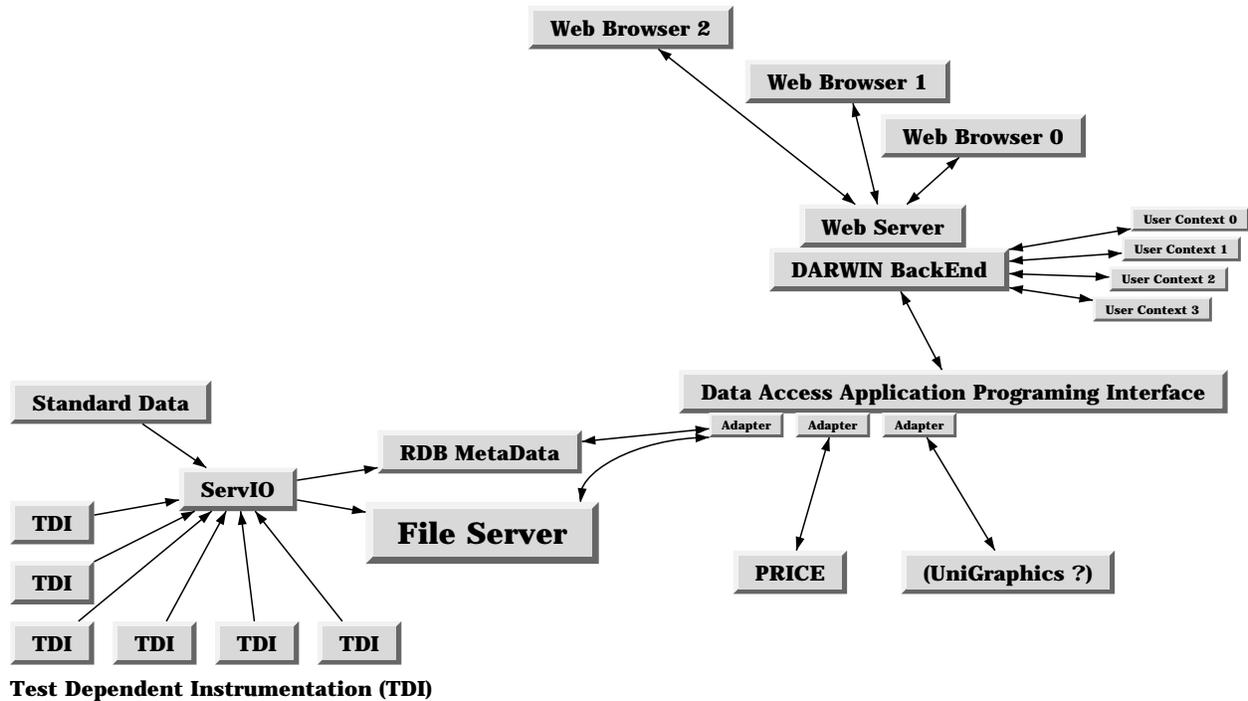


Figure 5: A Better Choice for DARWIN/PRICE Integration

field content can be conveyed at run-time to a discovery-oriented DARWIN browser.

3. Currently, PRICE does not recognize the concept of a sorted data base (for instance, all of the tests sorted by angle of attack so all the tests with an angle of attack over 45 degrees may be found), but only acts as a repository of data organized by the exploratory progression of the work (the n-ary tree of data configurations). It has long been expected that, when the need arose, PRICE would (and could with relative ease) implement auxillary indicies sorting its data configurations by particular parameters. Were DARWIN to adapt to the PRICE architecture, it might be appropriate to allocate this organization of data to a global role to be performed by DARWIN,

rather than a local role of the individual PRICE application.

4. Currently, the DARWIN system operates with knowledge of the database servers which serve it and, through them, locates all the data that is available to DARWIN. CORBA-Served PRICE application instances, on the other hand, are expected simply to float out in the ether of the net, served by their servers to whomever comes along (minding, of course, access controls not discussed here). There is, at present, no central source directory of PRICE application instances; however, there are a number of possibilities, one of which will have to be chosen.

Because PRICE implementation in a net-ready, CORBA-served form is still some time

away, ample time exists to debate and plan such a course adjustment/addition for DARWIN. Until sample PRICE applications are actually available in CORBA-served form, it will be difficult to take any serious implementation steps for such a DARWIN adaptation.

7 Concluding Suggestion

A sensible course of action is to adjust a future evolution of DARWIN to an infinitely flexible, discovery-oriented, data-world view.