{"sec":"Related Work"}

In this section I present academic literature related to this project, to derive my methodological approach from previous and established practices. It is crucial that I ground my approach in related work, as I aim to alter preexisting and complex structures with decades of history and development.

From this work I not only extrapolate organised information, but also methods for gathering, processing, structuring, and presenting data, features or other aspects of the investigated or developed material.

Previous projects have focused on documenting and comparing object serialisation formats in terms of features, efficiency, performance, file size, and programming language support. The central theme in these comparisons is the ability of a format to be applied across heterogeneous structures, systems and languages, thus achieving interoperability despite differences in data structure.

Data formats

The choice of data format is informed by the intended use of the stored data. An application utilising a proprietary data format is more likely to store information in a binary file, to reduce its file size. The proprietary nature of such an application does not necessitate interoperability, and thus transparency of the source data is a non-issue.

The inverse is true of an application interacting with other systems, such as a client receiving information from a platform developed by a source serving multiple clients. This necessitates interoperability, as the application must conform to the format defined by the source, and it is more likely to receive information in a plaintext file, to ensure correct interpretation. Plain text (ASCII) data is human-readable, resulting in a more transparent and approachable format at the cost of decreased performance and increased file size.

Data syntax

Plain data can be human-readable, but the decriptive syntax for the information contained is not necessarily accessible to humans. A proprietary format with limited scope can prescribe meaning implicitly to data objects, but an interoperable format must be explicit. This is achieved by surrounding information with descriptive tags, nesting it into hierarchies, and/or prescribing meaning to special characters such as tabs.

The data format must be interoperable with different languages and thus various parsers. The explicit nature of these data format streamlines the design of parsers, as data can be validated without implicit meaning.

{"break":true}