MetaDEx – From Data Dictionary to Complete Blaise Data Model Code Generation

Farit Vakhaetov, Department of Population Health Research, Alberta Health Services – Cancer Care

Abstract:

Population Health Research is a department focused on population-based epidemiological research in the areas of cancer incidence, causes, survival, and prevention. Blaise has been broadly used in the department to create CAPI data collection instruments from the early 2000s. Based on understanding of the crucial role of metadata definition for effective and robust data collection, management, and analysis, the Meta Data Extended (MetaDEx) Tool was initiated as a utility to create and maintain data dictionaries mainly for surveys, but it can be also used for analytical and other data sets. Its functionality was later extended to generating complete Blaise instrument code. The program was developed in C# and Microsoft Visual Studio 2008. All metadata is stored using eXtended Markup Language (XML).

MetaDEx facilitates a two-step approach in which subject matter specialists prepare specifications, and then IT specialists develop the Blaise CAI instrument. Data specification is structured hierarchically and defines the questionnaire content, text formatting, data flow, constraints, etc. While the system supports the generation of simple Blaise instruments directly from basic data definition, more complex data models require additional programming. The code generation module is built on advanced Blaise specific attributes and parameters which might be specified for any entity in the data dictionary. Additional ability to attach direct Blaise language elements (i.e. LOCALS, PARAMETERS, AUXFIELDS, SETTINGS, etc.) and entire code constructions (i.e. RULES or PROCEDURE definitions) to the section (block) provides virtually unlimited flexibility. Code generation is not limited to initial coding only, but also supports an iterative process of Blaise questionnaire authoring. MetaDEx completely eliminates any manual modifications in the data model script after it is generated, which guarantees congruency between the data dictionary and the actual production instrument in Blaise.