March 2009

منظمة الأغذية والزراعة للأمم المتحدة | 联合国粮食及农业组织 | Food and Agriculture Organization of the United Nations | Organisation des Nations Unies pour l'alimentation et l'agriculture | Продовольственная и сельскохозяйственная организация Объединенных Наций | Organización de las Naciones Unidas para la Agricultura y la Alimentación

FAO DESERT LOCUST CONTROL COMMITTEE

Thirty-ninth Session

Rome, 10-13 March 2009

The Desert Locust Contingency Planning Assistant

SIDE EVENT

The Desert Locust Contingency Planning Assistant (DLCPA) is a planning and analysis tool for both national Desert Locust Units and the UN Desert Locust Group. It is being developed by The Heron Group and FAO.

Effective Desert Locust monitoring and control is best achieved when the Desert Locust control organizations are well organized and properly prepared. The Desert Locust Planning Assistant evaluates the preparedness of a national unit by investigating the underlying organizational (structural) and operational (functional) status of these units. Data is collected through a web-browser like interface (see Figure 3) and is stored in a database for future re-evaluations and results mapping in a geographic Information System. The interface has provision for a multi-lingual presentation to facilitate use in native tongues.

The analysis report is presented both in narrative and graphical forms. The graphical presentation uses a "stoplight" approach where the degree of preparedness is indicated graphically by markers on colored bars. Green represents preparedness to a higher degree; red indicates a lack of preparedness while yellow indicates a need for improvement (see Figures 4 & 5). Lack of supporting data is indicated by the colored bars fading to white (see Figure 5).

The report is also interactive. Reports on preparedness are generated from antecedent questions which can be viewed to see which factors are contributing individually to a poor result (see Figures 1 &2). This kind of assessment can quickly show the most optimal places to make changes to improve Desert Locust Unit performance.

The DLCPA is built using the NetWeaver Runtime Framework which encapsulates a NetWeaver knowledge base. The current version of the DLCPA is a "proof of concept" and currently focuses at the national level.

Knowledge Bases

Knowledge bases are collections of formalized logic. Also known as symbolic reasoning, knowledge-based reasoning is the name for a general modeling methodology in which phenomena are described in terms of abstract entities and their logical relations to one another. Several different approaches to knowledge based representation have been developed over about the last 20

years. The DLCPA uses the NetWeaver inference engine for knowledge-based reasoning and NetWeaver Developer for knowledge engineering.

The basic building blocks of a NetWeaver knowledge base are dependency networks (see Figures 1 &2): hierarchical logic diagrams that describe how information is related. Put as generally as possible, a knowledge base embodies knowledge about how to solve a problem in some problem domain. By itself, a knowledge base doesn't actually do anything. The knowledge base that a team constructs with the NetWeaver development system is best thought of as a meta-database. It tells the inference engine how to process the data.

Some valuable aspects of the NetWeaver approach are:

- The ability to deal with incomplete information
- Consistency in evaluation
- Transparency of the evaluation process
- Analysis of data needs

The NetWeaver Runtime Framework

The NetWeaver Runtime Framework creates stand-alone knowledge-based reasoning applications from NetWeaver knowledge bases. These applications are very user friendly: users do not need to understand the logic model behind the application, nor understand the development tools, but can still view and investigate the logic if they choose to.

A key feature of this system is its smart question directing. NetWeaver ranks each unanswered question by its relative influence on the current investigation and asks the most influential question, re-ranking the questions each time data is entered until either all questions are answered (or skipped by the user) or until none of the questions left can influence the outcome. This method eliminates much unnecessary data entry and lets the user see which questions are the most important.

With the NetWeaver Runtime Framework, interface and delivery languages can be changed on the fly. Delivering the application in as many languages as needed, all in one concise package, is easy with its internal object-oriented database.

The framework has extensive embedded documenting and hyper linking capabilities so that external references can be easily accessed while using the software.

NetWeaver runtime applications are important because they push knowledge and intelligent decision making to where it is needed, whether it is data analysis, condition assessment, or guided decision making. NetWeaver runtime applications put the wisdom and experience of the few into the hands of the many.

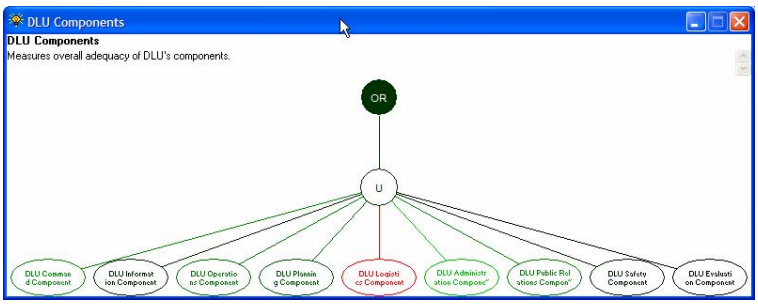

Figure 1 - An upper level dependency network describing the components of a Desert Locust Unit.

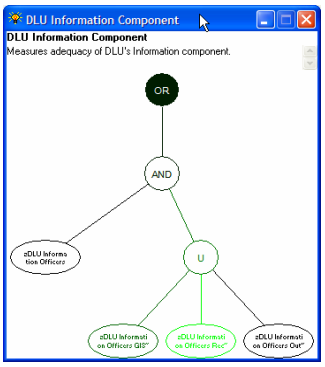

Figure 2 - A lower level dependency network describing the Information Component

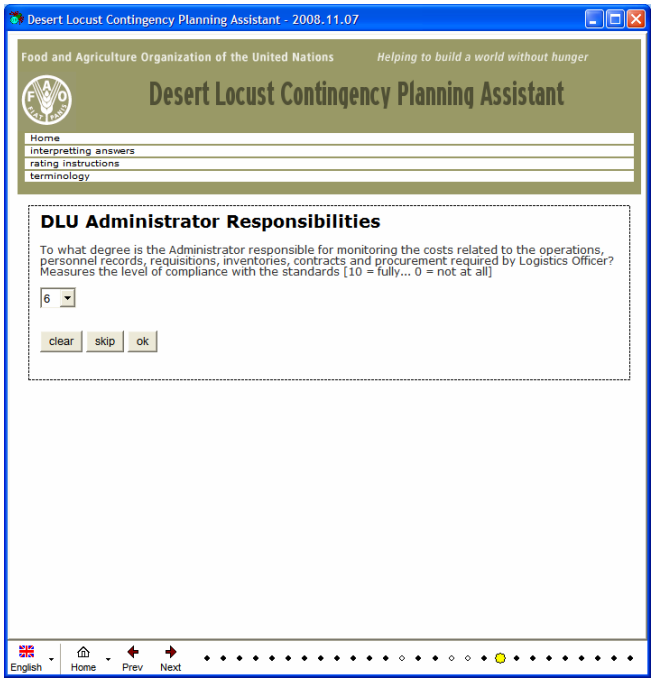

Figure 1 - A typical data entry page from the DLCPA. Data is also entered through a database when it's available.

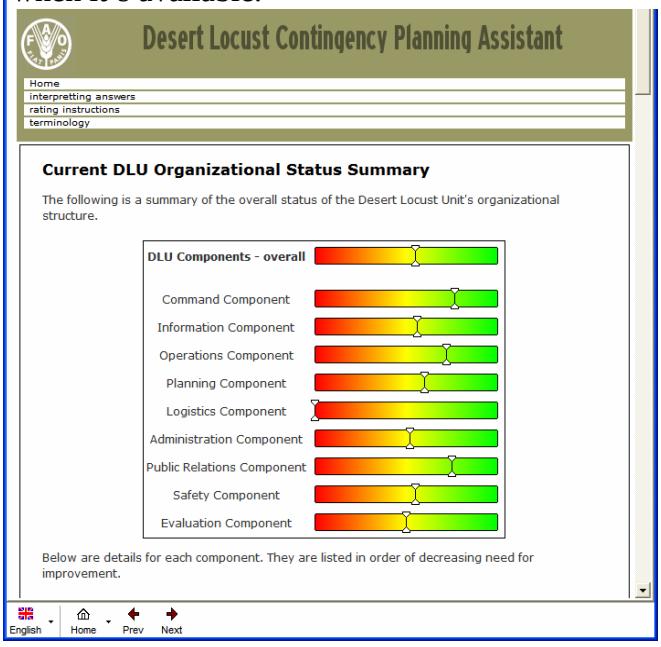

Figure 4 - Sample output showing a summary of the overall status of each component in the Desert Locust Unit.

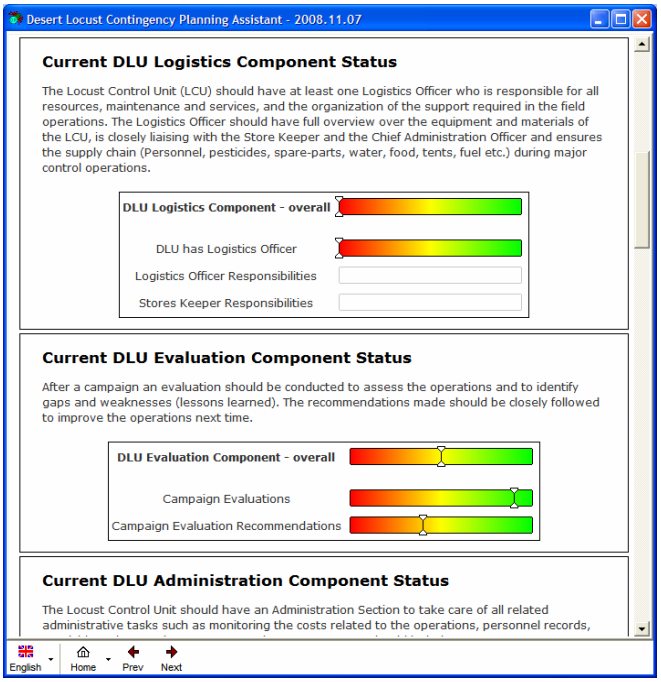

Figure 5 - Part of the more detailed section of the report. These details are listed in order of importance: the items nearer the top of the list are those that are most in need of improvement.

Note the position of markers to indicate preparedness level. Also note the two faded out bars that indicate no data was collected for those two topics. In this case the data was not asked for because the inference engine understood that the two topics were irrelevant because in this case the answer to the "DLU has Logistics Officer" was "no".