



Universidad Politécnica
de Madrid



**Escuela Técnica Superior de
Ingenieros Informáticos**

Grado en Ingeniería Informática

Trabajo Fin de Grado

**Generación de informes y salida gráfica
en el sistema de ayuda la decisión
WEB-MAUT-DSS**

Autor: Daniel Fernández-Pello San Román

Tutor: Antonio Jiménez Martín

Cotutor: Alberto Gómez Jiménez

Madrid, Julio 2025

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Ingeniería Informática

Título: Generación de informes y salida gráfica en el sistema de ayuda la decisión WEB-MAUT-DSS

Julio 2025

Autor: Daniel Fernández-Pello San Román

Tutor: Antonio Jiménez Martín

Departamento de Inteligencia Artificial

ETSI Informáticos

Universidad Politécnica de Madrid

Cotutor: Alberto Gómez Jiménez

Departamento de Tecnologías de la Información

Escuela Politécnica Superior

Universidad San Pablo CEU

Resumen

Este Trabajo de Fin de Grado se ha centrado en la extensión y modernización de la aplicación desarrollada por la Universidad Politécnica de Madrid WEB-MAUT-DSS que es una aplicación basada en los sistemas de ayuda a la decisión que utiliza la Teoría de la Utilidad Multiatributo para ayudar al usuario a tomar una decisión en problemas complejos. La motivación surge debido a la necesidad de generar nuevas funcionalidades y nuevas gráficas para mejorar el análisis.

En concreto en este Trabajo de Fin de Grado se han implementado tres funcionalidades claves que son, en primer lugar un submenú “Load Example” en el desplegable “Project” que permite al usuario cargar automáticamente tres proyectos de ejemplo y descargar sus documentos en PDF. En segundo lugar en la sección de Ranking se han incorporado nuevas gráficas para comparar las utilidades de dos alternativas mediante diagramas de barras comparativas y boxplots de Monte Carlo añadiendo opciones para el zoom y la descarga en formato PNG. Por último también se ha incorporado la generación de informes en formato Word que construye documentos estandarizados que incluyen la jerarquía de objetivos, las utilidades parciales y globales, las gráficas de evaluación y la descripción detallada de cada nodo, todo ello gracias a la librería officer de R.

El proyecto se apoya en la Metodología de Análisis de Decisiones, definida por Keeney y Raiffa como un procedimiento “sistemático y lógico basado en axiomas” para abordar problemas complejos de forma racional, esta se descompone en siete etapas, donde primero se identifica el problema, con el objetivo de delimitar con precisión la decisión y sus objetivos, como segunda etapa se identifican los objetivos y atributos para determinar qué importa y cómo medirlo y también se construye la jerarquía de objetivos, la tercera etapa es la generación de alternativas donde se proponen cursos de acción viables y se describen sus consecuencias, ya en la cuarta etapa que es la evaluación de consecuencias se cuantifica el desempeño de cada alternativa en cada atributo, para en la quinta etapa que corresponde a la cuantificación de preferencias se obtienen funciones de utilidad y pesos para reflejar lo que valora el decisor, llegando a la tercera etapa que es el modelo de utilidad multiatributo y evaluación global se agrega la información anterior en una métrica que ordena las alternativas y por último en la séptima etapa corresponde al análisis de sensibilidad, aquí se explora la robustez de la decisión ante variaciones en datos y preferencias.

Abstract

This Final Degree Project has focused on the extension and modernisation of the application developed by the Universidad Politécnica de Madrid WEB-MAUT-DSS, which is an application based on decision support systems that uses the Multi-attribute Utility Theory to help the user to make a decision in complex problems. The motivation arises due to the need to generate new functionalities and new graphs to improve the analysis.

Specifically, three key functionalities have been implemented in this Final Degree Project: firstly, a 'Load Example' submenu in the 'Project' drop-down menu that allows the user to automatically load three example projects and download their documents in PDF. Secondly, in the Ranking section, new graphs have been incorporated to compare the utilities of two alternatives by means of comparative bar charts and Monte Carlo boxplots, adding options for zooming and downloading in PNG format. Finally, we have also incorporated the generation of reports in Word format that builds standardised documents that include the hierarchy of objectives, the partial and global utilities, the evaluation graphs and the detailed description of each node, all thanks to the R officer library.

The project is based on the Decision Analysis Methodology, defined by Keeney and Raiffa as a 'systematic and logical procedure based on axioms' to address complex problems rationally, this is broken down into seven stages, where first the problem is identified, in order to accurately define the decision and its objectives, the second stage is to identify the objectives and attributes to determine what matters and how to measure it and also build the hierarchy of objectives, the third stage is the generation of alternatives where viable courses of action are proposed and their consequences are described, In the fourth stage, which is the evaluation of consequences, the performance of each alternative in each attribute is quantified, and in the fifth stage, which corresponds to the quantification of preferences, utility functions and weights are obtained to reflect what the decision-maker values, reaching the third stage, which is the multi-attribute utility model and global evaluation, the previous information is added in a metric that ranks the alternatives and finally, the seventh stage corresponds to the sensitivity analysis, here the robustness of the decision in the face of variations in data and preferences is explored.

Agradecimientos

Antes de comenzar el desarrollo me gustaría expresar mi agradecimiento a todas las personas que han contribuido a la realización de este Trabajo de Fin de Grado.

Primero de todo me gustaría agradecer a mi tutor Antonio Jiménez Martín y mi cotutor Alberto Gómez Jiménez por la orientación, constante apoyo, el seguimiento realizado y guía mientras realizaba el Trabajo de Fin de Grado, siendo así fundamentales para la elaboración de este trabajo.

También me gustaría agradecer a toda mi familia, sobre todo a mis padres por su incondicional apoyo, amor y comprensión. Además me gustaría agradecer el darme la oportunidad, confianza y recursos para estudiar la carrera y su ayuda a lo largo de esta ya que sin ellos no podría ser posible.

Por último me gustaría agradecer a mis amigos y compañeros que me han prestado su ayuda tanto emocional como académica durante la realización de la carrera.

Tabla de contenidos

| | |
|--|-----------|
| 1 Introducción | 1 |
| 1.1 Contexto del Trabajo | 1 |
| 1.2 Plan de trabajo | 1 |
| 1.3 Objetivos del TFG | 2 |
| 2 Estado del Arte | 3 |
| 2.1 Sistemas de Ayuda a la decisión (DSS) | 3 |
| 2.1.1 Motivación y contexto de los DSS en la toma de decisiones | 3 |
| 2.1.2 Definiciones de los DSS | 3 |
| 2.1.3 Evolución de los DSS | 4 |
| 2.1.4 Componentes principales y arquitectura de un DSS | 4 |
| 2.1.5 Casos de uso de los DSS en distintos sectores | 5 |
| 2.2 Metodología de Análisis de Decisiones | 6 |
| 2.2.1 Qué es la metodología de análisis de decisiones | 6 |
| 2.2.2 El ciclo de análisis de decisiones | 7 |
| 2.2.3 Fundamentos de la Teoría de la Utilidad Multiatributo (MAUT) | 9 |
| 2.2.4 Toma de decisiones con información incompleta/imparcial | 11 |
| 2.3 Sistema WEB-MAUT-DSS | 13 |
| 2.3.1 Introducción y motivación de WEB-MAUT-DSS | 13 |
| 2.3.2 Arquitectura general del sistema WEB-MAUT-DSS | 13 |
| 2.3.3 Funcionalidades de WEB-MAUT-DSS | 16 |
| 2.4 Lenguaje de R y su librería Shiny | 17 |
| 2.4.1 Historia y características principales de R | 17 |
| 2.4.2 La librería Shiny | 17 |
| 3 Desarrollo de las Funcionalidades | 19 |
| 3.1 Implementación de la carga de proyectos | 19 |
| 3.1.1 Requisitos funcionales de la carga de proyectos | 19 |
| 3.1.2 Interfaz de usuario de la carga de proyectos | 19 |
| 3.1.3 Servidor de la carga de proyectos | 20 |
| 3.2 Implementación de los diagramas | 21 |
| 3.2.1 Requisitos funcionales de los diagramas | 21 |
| 3.2.2 Interfaz de usuario de los diagramas | 21 |
| 3.2.3 Servidor común | 23 |
| 3.2.4 Gráfico del diagramas de barras de comparación | 25 |
| 3.2.5 Servidor del diagramas de barras de comparación | 26 |
| 3.2.6 Gráfico del diagrama de cajas | 27 |
| 3.2.7 Servidor del diagrama de cajas | 28 |
| 3.3 Implementación de la generación de informes | 29 |
| 3.3.1 Requisitos funcionales de la carga de proyectos | 29 |
| 3.3.2 Interfaz de usuario de la generación de informes | 29 |
| 3.3.3 Servidor de la generación de informes | 30 |
| 4 Pruebas y Validaciones | 39 |
| 4.1 Pruebas de interfaz e integración | 39 |

| | |
|---|-----------|
| 4.1.1 Mockup de la carga de proyectos | 39 |
| 4.1.2 Mockup de los diagramas | 40 |
| 4.1.3 Mockup de la generación de informes | 41 |
| 4.2 Validación de requisitos funcionales | 42 |
| 4.2.1 Mockup de la carga de proyectos | 42 |
| 4.2.2 Mockup de los diagramas | 42 |
| 4.2.3 Mockup de la generación de informes | 43 |
| 5 Conclusiones | 44 |
| 6 Análisis de Impacto | 45 |
| Bibliografía | 46 |
| 7 Anexo | 48 |

1 Introducción

1.1 Contexto del Trabajo

Los Sistemas de Ayuda a la Decisión o DSS son herramientas críticas en entornos donde se requiere evaluar múltiples alternativas bajo criterios complejos. En este contexto, el proyecto se enmarca en la mejora de WEB-MAUT-DSS, esta es una plataforma basada en la metodología MAUT que ha sido diseñada para facilitar la toma de decisiones multicriterio. Las funcionalidades principales desarrolladas en este TFG consisten en la generación de informes, el diseño de un nuevo diagrama de alternativas y la opción de cargar proyectos de ejemplo proporcionados por la plataforma.

1.2 Plan de trabajo

| Id. | Nombre de la tarea | Duración (Semanas) | Febrero | | Marzo | | | | Abril | | | | |
|-----|---|--------------------|---------|-------|-------|-------|-------|-------|-------|------|-------|-------|------|
| | | | 17-21 | 24-28 | 3-7 | 10-14 | 17-21 | 24-28 | 31-4 | 7-11 | 14-18 | 21-25 | 28-2 |
| 1 | Estudio de la Metodología de Análisis de Decisiones y ejemplos | 3 | | | | | | | | | | | |
| 2 | Estudio de la herramienta WEB-MAUT-DSS | 3 | | | | | | | | | | | |
| 3 | Repaso de R y sus librerías de visualización y aprendizaje Shiny | 5 | | | | | | | | | | | |
| 4 | Análisis y compresión de la estructura de datos (iGraph) del sistema WEB-MAUT-DSS | 2 | | | | | | | | | | | |
| 5 | Identificación de requisitos funcionales y diseño. | 3 | | | | | | | | | | | |
| 6 | Desarrollo de las herramientas asignadas | 8 | | | | | | | | | | | |
| 9 | Redacción de la memoria del TFG | 7 | | | | | | | | | | | |

| Id . | Nombre de la tarea | Duración (Semanas) | Mayo | | | | Junio | | | | Julio | | |
|------|---|--------------------|------|-------|-------|-------|-------|------|-------|-------|-------|------|-------|
| | | | 5-9 | 12-16 | 19-23 | 26-30 | 2-6 | 9-13 | 16-20 | 23-27 | 30-4 | 7-11 | 14-18 |
| 6 | Desarrollo de las herramientas asignadas | 8 | | | | | | | | | | | |
| 7 | Integración de los desarrollos en el sistema WEB-MAUT-DSS | 3 | | | | | | | | | | | |
| 8 | Validación de la implementación | 3 | | | | | | | | | | | |
| 9 | Redacción de la memoria del TFG | 7 | | | | | | | | | | | |
| 10 | Preparación presentación oral | 2 | | | | | | | | | | | |

1.3 Objetivos del TFG

Los objetivos planteados se alinean con las necesidades identificadas en WEB-MAUT-DSS:

- Conocer la Metodología de Análisis de Decisiones.
- Conocer las características del sistema de ayuda a la decisión WEB-MAUT-DSS.
- Conocer la implementación del sistema de ayuda a la decisión WEB-MAUT-DSS
- Desarrollar la funcionalidad que genere informes con información del problema analizado.
- Desarrollar la funcionalidad que sea capaz de cargar proyectos de ejemplos y descargar sus pdf.
- Desarrollar la funcionalidad que permite visualizar dos nuevos diagramas de comparación de alternativas.

2 Estado del Arte

2.1 Sistemas de Ayuda a la decisión (DSS)

2.1.1 Motivación y contexto de los DSS en la toma de decisiones

La toma de decisiones basada en datos busca aprovechar la abundancia de información para fundamentar elecciones estratégicas y operativas, esta se basa en extraer conocimiento útil mediante técnicas de análisis y ciencia de datos [1]. Tecnologías como la minería de datos y el aprendizaje automático permiten descubrir los patrones ocultos incluso en los conjuntos de datos más grandes y complejos [1], mientras que la visualización de datos ayuda a los responsables a comprender mejor los hallazgos y que estos los incorporen en sus decisiones [1]. En este consejo surge la necesidad de las herramientas especializadas que ayuden en el proceso de la toma de decisión, integrando datos, análisis y presentación de información de forma intuitiva. Los Sistemas de Ayuda a la Decisión o DSS (Decision Support Systems) responden a esta necesidad proporcionando entornos interactivos donde se puede, explorar la información relevante, evaluar alternativas y tomar decisiones informadas basadas en datos. En resumen la necesidad se solventa con los DSS ya que estos permiten a individuos y organizaciones convertir datos en conocimientos accesibles enfrentados a la complejidad e incertidumbre con apoyo analítico en lugar de depender exclusivamente de métodos más irracionales como la intuición.

2.1.2 Definiciones de los DSS

Actualmente no existe ninguna definición única y aceptada universalmente de los DSS ya que cada autor enfatiza aspectos de los DSS diferentes. De esta manera estos se han definido los DSS como *"un sistema basado en computador que ayuda en el proceso de toma de decisiones"* [2]. Esta definición básica propuesta por Finlay en 1994 destaca el rol de apoyo que ofrece a la computadora el decisor humano. Otras definiciones más específicas como la propuesta por Turban en 1995 amplían el concepto de los DSS definiéndolo como *"un sistema de información basado en un computador interactivo, flexible y adaptable, especialmente desarrollado para apoyar la solución de un problema de gestión no estructurado para mejorar la toma de decisiones. Utiliza datos, proporciona una interfaz amigable y permite la toma de decisiones en el propio análisis de la situación"* [3]. Otros autores subrayan particularidades como Little en 1970 que habla de un *"conjunto de procedimientos basados en modelos para procesar datos y juicios para asistir a un gerente en su toma de decisiones"* [4], mientras que Keen en 1980 resalta que un DSS *"combina recursos intelectuales individuales con las capacidades de un ordenador para mejorar la calidad de las decisiones"* [5], también encontramos a Sprague y Carlson que en 1982 describieron los DSS como *"Sistemas informáticos interactivos que ayudan a los encargados de tomar decisiones utilizando datos y modelos para resolver problemas no estructurados"* [6]. En conclusión un DSS no pretende automatizar la decisión ni reemplazar las decisiones humanas, sino que sirve para apoyar el proceso cognitivo proporcionando la información relevante, los modelos de análisis y herramientas interactivas para estimar, evaluar y comparar alternativas [7].

2.1.3 Evolución de los DSS

El concepto de apoyo a la decisión nace a mediados del siglo XX donde los pioneros teóricos de la toma de decisiones como Herber Simon en *Administrative Behavior* en 1947 sentaron las bases sobre cómo la gente toma decisiones en organizaciones [8]. Por otro lado, los avances técnicos en sistemas informáticos interactivos durante los años 60 demostraron el potencial de los ordenadores para asistir a los usuarios en tareas complejas. Estas dos corrientes confluyen en la década de 1970, cuando el término DSS comienza a formarse convirtiéndose así un campo de estudio propio [5] siendo los primeros en proponer un marco para sistemas de apoyo a decisiones Gorry y Scott-Morton en 1971.

Durante la década de 1980 se vivió una expansión importante ya que aparecen por primera vez los Sistemas de Información Ejecutiva (EIS) orientados a altos directivos, los Sistemas de Apoyo a Decisiones en Grupo (GDSS) que permitían la colaboración de equipos para tomar decisiones y los Sistemas Organizacionales de Decisión (ODSS) que abarcan escenarios a nivel organizacional. Esto evidenciaba la necesidad de soportar decisiones a distintos niveles y de facilitar la colaboración grupal además de las decisiones individuales.

Ya en la década de 1990 con la explosión de los almacenes de datos y las herramientas de procesamiento en línea amplió enormemente el alcance de los DSS, ya que ahora era posible para las empresas centralizar grandes volúmenes de datos históricos además de interactuar con ellos para extraer información estratégica. Esto provocó que los DSS se volvieran más data-driven apoyando decisiones mediante análisis de grandes bases de datos y visualización dinámica de la información.

Con el cambio de milenio y las nuevas aplicaciones web, en la década de los 2000, los DSS entraron en una siguiente etapa [8] en la que comienzan a surgir sistemas accesibles vía web que permiten la toma de decisiones remota y distribuida aprovechando internet para integrar fuentes de datos externas en tiempo real, también al mismo tiempo aparecen paneles de control y portales ejecutivos que muestran indicadores clave con una actualización inmediata.

Finalmente del 2010 en adelante los DSS han evolucionado de la mano del Big Data, la analítica predictiva y la analítica prescriptiva ya que ha empezado a incorporar algoritmos de machine learning e incluso modelos de optimización que además de realizar predicciones sugieren el curso de acción óptimo. Esta última evolución ha provocado que algunos hablen de sistemas de inteligencia de decisiones o Decisión Intelligence que integra análisis avanzados con la lógica de negocio.

2.1.4 Componentes principales y arquitectura de un DSS

Aunque pueden existir variaciones, la arquitectura típica de un DSS se asemeja a una estructura de tres capas que consta de datos, modelos e interfaz. No obstante un DSS se divide en los siguientes tres componentes fundamentales [6]:

- **Base de datos o base de conocimiento:** Se refiere al repositorio de información utilizado por el DSS, esto incluye los datos internos de la organización y potencialmente también los datos externos. Esta capa de datos puede implementarse en forma de almacén de datos centralizado o simplemente mediante conexiones a las bases de transacciones

existentes. El DSS se encarga de extraer, ordenar, filtrar y manipular los datos para convertirlos en información. En sistemas avanzados la base de datos puede ampliarse a base de conocimientos, incorporando también reglas, casos pasados, modelos de aprendizaje... que enriquecen la información disponible.

- **Motor de modelos o sistema de análisis:** Se refiere al módulo que provee las herramientas analíticas y de modelado para procesar los datos [9]. Aquí es donde residen los algoritmos, modelos matemáticos, procedimientos de simulación... que el DSS pone al servicio del usuario, este puede contener modelos cuantitativos estándar o modelos específicos desarrollados para el dominio del problema. Sprague y Carlson se referían a esta parte como sistema gestor de modelos, el cual estaba encargado de representar situaciones mediante modelos además de ejecutar análisis de escenarios o análisis de sensibilidad [6]. La flexibilidad de un DSS se suele medir por la variabilidad y adaptabilidad de sus modelos, desde cálculos simples hasta optimizaciones complejas como simulaciones Monte Carlo o evaluaciones multicriterio.
- **Subsistema de interfaz de usuario:** Se refiere al componente que permite la interacción entre el usuario decisor y el DSS en cuestión [6], esto incluye las pantallas, paneles de control, formularios de consulta, herramientas de visualización gráfica, reportes e incluso interfaces del lenguaje natural en sistemas modernos. Este apartado es crítico en un DSS ya que esto determina en gran parte la usabilidad y aceptación por parte del usuario decisor, puede ir desde ventanas de menú sencillas o líneas de comando hasta a entornos gráficos avanzados con mapas, gráficos dinámicos... La interfaz además cumple la función de gestionar el código guiando al usuario a un flujo lógico. Sprague y Carlson llamaron a esta capa sistema generador de diálogos reconociendo en aquella época que un DSS no es únicamente los datos y los cálculos sino que es una herramienta con la cual el humano conversa para explorar la decisión a tomar [6].

Estas tres partes conforman la arquitectura básica propuesta para los DSS más básicos [6], no obstante distintos autores han extendido este esquema añadiendo elementos, como Power que considera la infraestructura tecnológica y de red como un componente adicional, esto sobre todo en los DSS modernos distribuidos [9], otros en cambio incluyen al usuario y el contexto dentro de la arquitectura como Hättenschwiler, en 1999, que identifica al tomador de decisiones, equipo de asesores como una parte integral del sistema así como el entorno de trabajo en el que se aplica y el entorno de trabajo donde se preparan y documentan las decisiones [10].

Asimismo, se habla en los DSS más avanzados de un módulo de gestión del conocimiento que se encarga de manejar el conocimiento organizado y a veces de un aprendizaje incorporado como consistir en el almacenaje de fallos anteriores y soluciones aplicadas para corregir el fallo en futuras ocasiones.

2.1.5 Casos de uso de los DSS en distintos sectores

Los DSS tienen aplicaciones prácticamente cualquier ámbito que implique la toma de decisiones complejas basadas en datos, a continuación se describe casos de uso ilustrativos en distintos sectores para mostrar su amplia utilidad y uso:

- **Salud:** En este sector abundan los Sistemas de Apoyo a la Decisión Clínica (CDSS) los cuales asisten a los médicos y personal de salud en diversas tareas como el diagnóstico a pacientes, elección de los tratamientos o la gestión de pacientes. Un caso citado es un DSS desarrollado por la Universidad de Pensilvania que ayuda a determinar cuándo retirar a un paciente de un ventilador mecánico de la forma más rápida y segura. En los hospitales modernos los DSS se integran a menudo en el historial clínico electrónico mostrando alertas o recomendaciones en tiempo real mientras el médico introduce datos.
- **Administración pública:** En este sector los gobiernos y administraciones emplean los DSS para mejorar la toma de decisiones públicas y la formulación de políticas. Un ejemplo es el uso de DSS para la seguridad ciudadana y la lucha contra el crimen que analizan grandes conjuntos de datos policiales, económicos y sociales, ya que aquí un data-driven DSS puede identificar puntos calientes de delincuencia cruzando bases de datos delictivos con datos demográficos para orientar la asignación de patrullas policiales. También pueden ser utilizados en planificación urbana donde se utilizan DSS participativos con mapas interactivos donde se visualizan capas de información como el tráfico, contaminación o densidad poblacional y simulan el impacto de decisiones como la creación de una nueva infraestructura.
- **Industria y manufactura:** En este sector los DSS apoyan a decisiones que van desde la gestión de inventarios y cadenas de suministro hasta el control de producción en planta. Una solución interesante la reporta Semantic Systems en su solución PRODEC, que consiste en un DSS que acelera la configuración de productos industriales complejos guiando al personal no experto en base a unas reglas derivadas de configuraciones previas existentes [11].

Estos serían algunos ejemplos de la utilidad de los DSS, en estos casos se evidencia que los DSS no reemplazan la decisión humana sino que amplifican la capacidad humana al procesar más información de la que una persona podría manejar y mostrarla de una manera estructurada para la decisión ofreciendo así perspectivas adicionales a las que una persona puede pensar inicialmente.

2.2 Metodología de Análisis de Decisiones

2.2.1 Qué es la metodología de análisis de decisiones

El análisis de decisiones es una disciplina cuyo objetivo es estructurar y simplificar la toma de decisiones complejas, para ello se proporciona un marco sistemático para elegir racionalmente entre alternativas. Según Keeney y Raiffa el análisis de decisiones constituye un procedimiento sistemático y lógico basado en axiomas para analizar problemas decisionales complejos de forma racional [12]. A diferencia de enfoques más tradicionales como el análisis coste-beneficio el análisis de decisiones incorpora las preferencias y los juicios del decisor en el modelo, logrando así integrar información cuantitativa y cualitativa de manera formal.

El análisis de decisiones parte de la premisa de que lo atractiva que es una alternativa depende de las probabilidades de sus distintas consecuencias y las preferencias del decisor respecto a las consecuencias [13], esto implica combinar la incertidumbre con la valoración subjetiva de cada resultado, siendo el cómo cuantifica y fusiona estos valores (mediante la teoría de la probabilidad para las incertidumbres y la Teoría de la Utilidad para las

preferencias) la característica distintiva del análisis de decisiones [14]. De este modo toda la información relevante disponible se organiza en una estructura manejable, descomponiendo el problema en partes más sencillas que posteriormente se combinarán para evaluar cada decisión [6].

2.2.2 El ciclo de análisis de decisiones

El proceso de análisis de decisiones suele representarse como un ciclo iterativo de etapas interrelacionadas, podemos ver en la figura 1.1 que se ilustran las fases del ciclo que se retroalimentan hasta converger en una única decisión robusta.

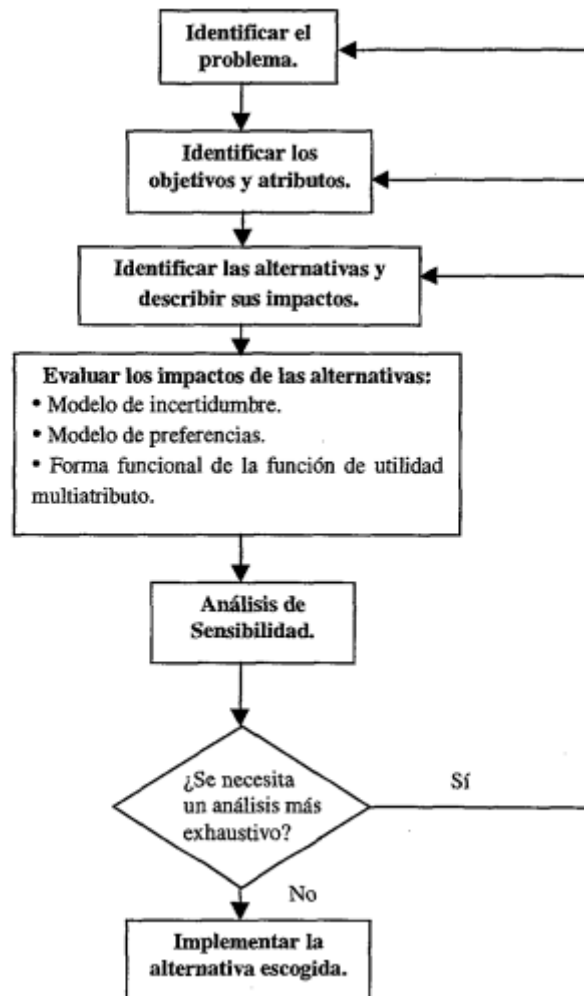


Figura 1.1: Ciclo del Análisis de Decisiones [15]

A continuación hablaré de cada una de las etapas del ciclo de análisis de decisiones:

- **Identificación del problema:** Es el punto de partida del análisis de decisiones, consiste en definir con claridad la decisión a tomar y lo que se pretende lograr. Un analista debe evitar errores de definición asegurándose de abordar la raíz del problema y no solo síntomas superficiales, ya que una correcta formulación sienta las bases del análisis posterior [16].
- **Identificación de objetivos y atributos:** Una vez claro el problema, el decisor debe preguntarse qué es lo que importa y cómo medirlo, para ello se identifican los objetivos fundamentales los cuales reflejan las

preocupaciones y valores del decisor, para cada objetivo se deben definir atributos o criterios que permitan medir en qué grado las alternativas satisfacen al objetivo [18]. Los objetivos se suelen encontrar en una jerarquía de objetivos que ayuda a descomponer la decisión y clarificar las prioridades, llevando cada objetivo del nivel más bajo asociado uno o más atributos o indicadores. A la hora de definir los atributos hay que asegurarse que el conjunto sea completo, operativo, descomponible, no redundante y mínimo, además los atributos pueden clasificarse según cómo miden al objetivo o según el tipo de escala utilizan [13].

- **Generación de alternativas:** En paralelo a la definición de objetivos, se identifican y generan las alternativas de decisión a considerar [17], estas alternativas son los diferentes cursos de acción o soluciones posibles al problema, dependiendo del caso puede haber muchas o muy pocas alternativas pero en todos los casos cada alternativa se describe mediante sus consecuencias o impacto en los atributos identificados.
- **Evaluación de consecuencias:** Para cada alternativa se determina un perfil de consecuencias donde se especifica el nivel de desempeño de la alternativa en cada atributo [18], bajo certeza esto se puede representar con un vector $x_i = (x_{i1}, \dots, x_{in})$ que indica el valor de la alternativa a_i en cada atributo X_j [19]. En problemas reales es común enfrentar incertidumbre sobre los impactos, ya que las consecuencias pueden modelizarse como rangos o distribuciones de probabilidad en cada atributo, para ello el análisis de decisiones proporciona herramientas para manejar la incertidumbre, identificando las fuentes y cuantificando de manera explícita haciendo que la incertidumbre forme parte del análisis.
- **Cuantificación de preferencias:** Esta etapa consiste en extraer del decisor información sobre cómo valora las consecuencias en cada criterio y que importancia relativa concede a los diferentes objetivos, para ello se construyen funciones de utilidad individuales por atributo y se determinan los pesos para los objetivos o criterios. Todo esto es sustentado por la Teoría de la Utilidad de von Neumann y Morgenstern [13] que proporciona axiomas y métodos para asignar números a preferencias bajo incertidumbre.
- **Modelo de utilidad multiatributo y evaluación global:** Conocidas las utilidades individuales $u_j(x_{ij})$ para cada atributo y los pesos w_j asociados a cada criterio se construye una función de utilidad multiatributo que agregue toda esta información y asigne a cada alternativa a_i un valor global $u(a_i)$ que representa la deseabilidad global de la alternativa para el decisor, teniendo en cuenta todos los criterios, además de la actitud al riesgo y compensaciones entre criterios [18]. El desarrollo teórico de la MAUT demuestra que bajo ciertas condiciones de independencia entre preferencias de los criterios la función multiatributo se puede descomponer en combinaciones más simples de las utilidades unatributo.
- **Análisis de sensibilidad:** Una vez identificada la mejor alternativa el ciclo del análisis de decisiones no acaba aquí sino que entra en una fase de análisis de sensibilidad que es un proceso iterativo que tiene como objetivo evaluar la robustez de la decisión obtenida y comprender qué factores influyen críticamente en el resultado. Si pequeños cambios en algún parámetro provocan una alteración del orden de preferencia se

concluye que la decisión es sensible a ese parámetro y alerta al decisor de la necesidad de revisar esa parte del modelo [20].

Finalmente el ciclo de AD, con sus etapas de modelización y su análisis iterativo continúa hasta que el decisor esté satisfecho con la solución obtenida o en su defecto hasta que no se puede refinar más el modelo sin nueva información externa [16].

2.2.3 Fundamentos de la Teoría de la Utilidad Multiatributo (MAUT)

La Teoría de la Utilidad Multiatributo o MAUT es la base teórica para comparar y valorar alternativas cuando intervienen múltiples criterios, esta es una extensión de la teoría de la utilidad unidimensional de von Neumann-Morgenstern a casos multicriterio. Su fundamento es el asignar a cada alternativa un índice global a partir de desempeños en los distintos atributos siendo coherente con las preferencias del decisor. Los principios más importantes de MAUT serían:

- **Jerarquía de objetivos y criterios:** MAUT consiste en una estructura de descomposición del problema en objetivos y subobjetivos hasta llegar a criterios operativos. La jerarquía permite asumir condiciones de independencia entre ramas facilitando la separación de las utilidades por criterio [21]. Los objetivos de la jerarquía tienen que estar bien definido con su atributo asociado teniendo una escala de medición, esta puede ser natural, ya que a menudo se puede asignar directamente una utilidad creciente o decreciente acorde al sentido común, pero también puede ser subjetiva, esto suele requerir de un proceso de calibración más cuidadoso con el decisor para mapear niveles cualitativos a valores de utilidad. Otra característica de la MAUT es que las utilidades uniatributo se normalizan entre dos números donde el más bajo representa el peor nivel y el más alto el mejor nivel en cada criterio, esto junto con la posibilidad de combinar métricas permite agregar criterios medidos en unidades incompatibles.
- **Funciones de utilidad uniatributo:** La utilidad refleja la preferencia del decisor por los distintos resultados posibles en el criterio analizado. La MAUT recurre a técnicas de elicitación para construir la curva $u_j(x)$ en cada atributo X_j , aunque a veces en problemas complejos puede resultar difícil el construir la curva $u_j(x)$, en estos casos se puede introducir información parcial [13]. Estas funciones de utilidad imprecisas se utilizan luego en el análisis de sensibilidad para verificar que funciones dentro de esas clases podrían cambiar el resultado.
- **Asignación de pesos y agregación aditiva:** Los pesos w_j en la MAUT expresan la importancia relativa de cada criterio en la utilidad real. En un modelo aditivo los pesos se normalizan para sumar 1 de modo que w_j indica la proporción de la decisión atribuible al criterio j . Al calcular la utilidad global con $u(a) = \sum_j w_j u_j(x_j)$, se obtiene una ordenación de alternativas de mayor a menor utilidad. La alternativa con mayor $u(a)$ sería la recomendada como mejor decisión bajo el modelo y preferencias actuales, llegados aquí se debe verificar si la diferencia entre la primera y las siguientes alternativas es significativa y robusta porque a veces pueden tener utilidades muy cercanas haciendo que se pueda considerar ambas como prácticamente empatadas teniendo así que realizar un análisis más fino y recopilar más datos para desempatar.

- **Caso de utilidades no aditivas:** Si la independencia aditiva no se cumple, la MAUT ofrece modelos más complejos aunque estos requieren información adicional del decisor y aumenta la complejidad. De entre los múltiples métodos sobresale el modelo multiplicativo de Keeney-Raiffa el cual es capaz de manejar cierto grado de sinergia entre criterios mediante parámetros de interacción, y que en ese modelo si todos los parámetros de interacción son nulos, se reduce al caso aditivo [14], lo que de nuevo resalta a este último como caso base y punto de partida recomendado.

También dentro de la MAUT encontramos el análisis de sensibilidad que permite comprobar la robustez de la decisión frente a cambios en los parámetros para clave del modelo como son los pesos o los valores de la utilidad, para ello encontramos distintos métodos:

- **Análisis clásico:** Donde se varía uno o varios pesos para observar si existen o no cambios en el ranking de alternativas.
- **Monte Carlo:** Donde se generan múltiples simulaciones aleatorias de pesos para obtener diagramas de cajas que representan la distribución de las utilidades reales. Un ejemplo de esto sacado de la aplicación de WEB-MAUT-DSS lo podemos ver en la figura 2.2 donde se simula unos pesos aleatorios y se muestran en un diagrama de cajas.

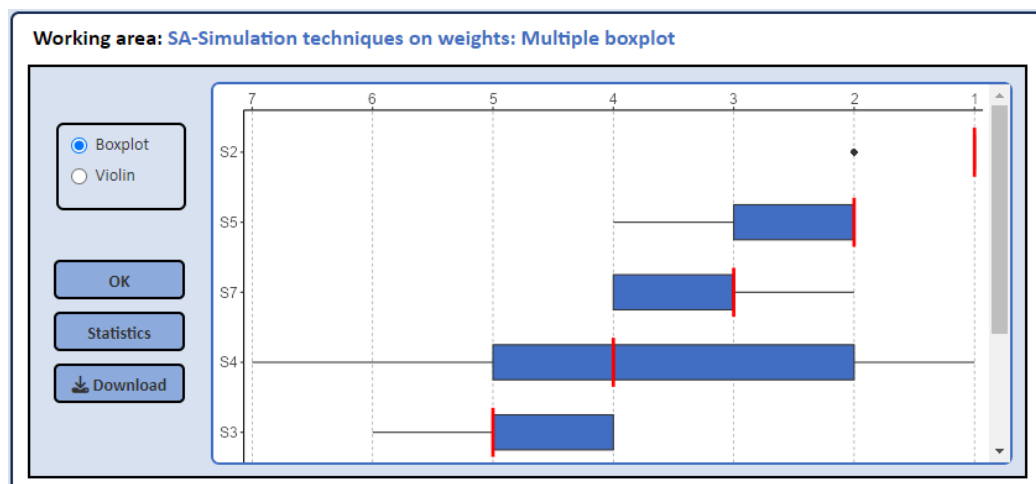


Figura 2.2

- **Intervalos de estabilidad:** Donde indican los rangos de pesos en los que el ranking se mantiene inalterado, teniendo una gran utilidad a la hora de conocer la estabilidad de una alternativa en relación con sus alternativas vecinas. Como ejemplo de esto encontramos la figura 2.3 en la que se muestra dentro de la aplicación una simulación donde se cambia el ranking de alternativas.

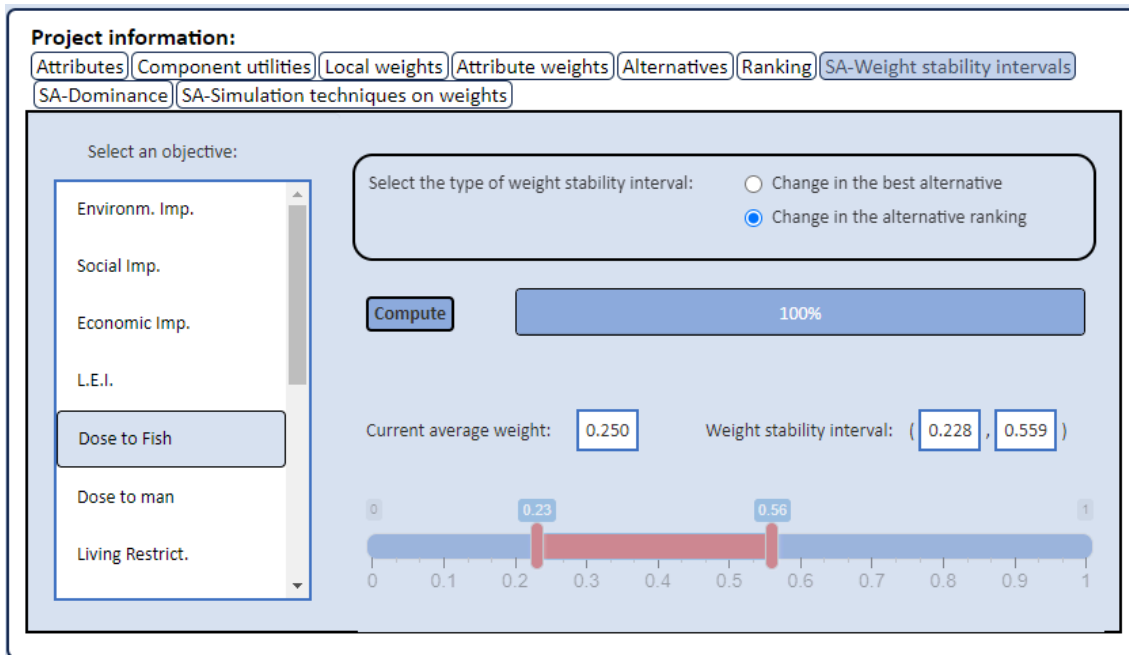


Figura 2.3

Como resumen la MAUT proporciona un lenguaje formal para articular objetivos, criterios, medidas de desempeño, utilidades y decisión, además su fundamento axiomático garantiza que, si el decisor puede expresar coherentemente sus preferencias y que el modelo producirá recomendaciones consistentes con esas preferencias. Lo importante de la MAUT es que integra toda la información relevante en un único marco cuantitativo, facilitando así decisiones transparentes, justificables y analizables.

2.2.4 Toma de decisiones con información incompleta/imparcial

En la toma de decisiones multicriterio en entornos reales complejos es habitual que los decisores no dispongan de toda la información que necesitan o que estos no estén dispuestos a proporcionar valoraciones precisas sobre sus preferencias. Esto se conoce como la toma de decisiones con información incompleta, imparcial o imprecisa, esto plantea desafíos metodológicos ya que impide aplicar directamente técnicas tradicionales como la MAUT definida en apartados anteriores.

Encontramos autores como Lahdelma, Hokkanen y Salminen que en 1998 explican que, “en muchas aplicaciones prácticas no es realista suponer que se conocen con certeza las utilidades o las distribuciones de probabilidad asociadas” [22], por lo que es necesario emplear modelos que toleran ambigüedad o desconocimiento parcial en las preferencias.

Dentro de esta incertidumbre encontramos que se manifiesta principalmente en dos aspectos, el primero siendo la dificultad para definir funciones de utilidad exactas. porque el decisor puede no conocer con precisión cómo se valoran los impactos de cada alternativa, y el segundo la ambigüedad en los pesos de los criterios ya que no es siempre posible o deseable asignar un número exacto a cada objetivo.

Para el desarrollo de esto se han desarrollado diversos enfoques y modelos adaptativos entre los que destacamos los siguientes:

- **Relaciones ordinales entre criterios:** Se refiere a cuando el decisor quiere expresar únicamente preferencias relativas entre los criterios sin cuantificarlas, aquí es posible utilizar métodos indirectos para estimar pesos aproximados a partir de un orden de importancia como[15]:
 - **Rank Sum (RS):** Asigna más pesos a los criterios mejor posicionados en el ranking usando la siguiente fórmula

$$w_j = \frac{n - r_j + 1}{\sum_{k=1}^n (n - r_k + 1)}$$
 - **Rank Reciprocal (RR):** Se basa en el inverso del rango favoreciendo a los criterios con mejor posición.
 - **Rank-Order Centroid (ROC):** Método propuesto por Stillwell en 1981 [23], calcula pesos como el promedio de todas las distribuciones compatibles con el ranking.
 - **Equal Weights (EW):** Asigna el mismo peso para todos los criterios, esto es válido únicamente cuando no se desea priorizar ninguno.
- **Modelos Stochastic Multicriteria Acceptability Analysis (SMAA):** Este modelo permite analizar la aceptabilidad de las alternativas cuando existe falta de información sobre los pesos o los impactos de las alternativas. Lo hace generando un gran número de simulaciones con distribuciones de pesos aleatorias compatibles con la información parcial disponible, y posteriormente calculando, el índice de aceptabilidad para ver la probabilidad de que una alternativa sea la mejor, el índice central para ver la alternativa más representativa del conjunto, y la dominancia estadística para ver la frecuencia con la que una alternativa supera a otra. Además en este modelo destacamos dos variantes:
 - **SMAA-2:** Que admite información parcial sobre pesos y utiliza funciones de utilidad definidas.
 - **SMAA-O:** Que trabaja con escalas ordinales, siendo útil cuando ni siquiera se conocen los impactos exactos.
- **Dominación y medidas de robustez:** Este es otro enfoque que consiste en evaluar las relaciones de dominación parcial o total entre alternativas mientras vas cambiando los pesos. Aquí destacamos las siguientes métricas [15]:
 - **Dominación neta:** Que mide en cuantas configuraciones una alternativa es superior a otra.
 - **Índices MID1 y MID2:** Que son propuestas en el documento de investigación, permiten cuantificar la robustez relativa de una alternativa frente al resto.
- **Modelos fuzzy y conjuntos borrosos:** Estos modelos permiten calcular imprecisiones en los impactos o preferencias mediante conjuntos borrosos, funciones de pertenencia o intervalos, estos son útiles cuando los juicios del decisor se expresan con términos lingüísticos ya que estos modelos permiten incorporar la incertidumbre semántica.
- **Toma de decisiones en grupo:** Estos modelos se utilizan cuando múltiples decisores participan en el proceso ya que puede haber preferencias divergentes o parciales, estos permiten modelar estos escenarios sin necesidad de obtener una función de utilidad común precisa.

2.3 Sistema WEB-MAUT-DSS

2.3.1 Introducción y motivación de WEB-MAUT-DSS

La toma de decisiones en problemas complejos con múltiples criterios suele superar la capacidad cognitiva humana sobre todo cuando existe incertidumbre [5], en respuesta a esta dificultad surgieron los Sistemas de Ayuda a la Decisión o DSS que son programas informáticos basados en la metodología de análisis de decisiones que asisten al decisor de forma estructurada y racional en la elección de la mejor alternativa. Dentro de este contexto diversos proyectos enfocados en la restauración de ecosistemas contaminados por radionúclidos impulsaron el desarrollo de herramientas especializadas como los proyectos MOIRA, COMETES, EVANET-HYDRA y EURANOS entre 1996 y 2008 [24].

Siguiendo esa línea de investigación se creó el sistema Generic Multi-Attribute Analysis (GMAA) que es un DSS genérico basado en un modelo de utilidad multiatributo capaz de manejar incertidumbre en las consecuencias de las alternativas e impresión en la cuantificación de preferencias del decisor. Sobre la base de GMAA, el grupo de la Universidad Politécnica de Madrid ha desarrollado WEB-MAUT-DSS, una versión web mejorada que actualmente se encuentra en desarrollo de aquel sistema genérico. WEB-MAUT-DSS conserva la metodología de análisis de decisiones con un modelo de utilidad multiatributo aditivo pero implementado en un entorno web moderno lo que permite una mayor accesibilidad y usabilidad ofreciendo en esencia una plataforma en línea para la visualización y apoyo a la toma de decisiones multicriterio.

2.3.2 Arquitectura general del sistema WEB-MAUT-DSS

WEB-MAUT-DSS está concebido como una aplicación web interactiva que interactiva que integra todos los componentes del análisis multicriterio de forma modular. La interfaz de usuario presenta una pantalla principal con varias secciones que se describe a continuación y se ve en la figura 2.1:

The screenshot displays the WEB-MAUT-DSS web application interface. The top navigation bar includes 'Project', 'Alternatives', and 'About' tabs. The main content area is divided into several functional sections. On the left, a hierarchical tree diagram illustrates the 'Ove. objective' structure, branching into 'Environm. Imp.', 'Social Imp.', and 'Economic Imp.', with further sub-objectives like 'L.E.I.', 'Dose to Fish', 'Dose to man', 'Living Restrict.', 'Direct Eff.', and 'Application Costs'. The right side features a 'Project information' panel with tabs for 'Attributes', 'Component utilities', 'Local weights', 'Attribute weights', 'Alternatives', 'Ranking', 'SA-Weight stability intervals', 'SA-Dominance', and 'SA-Simulation techniques on weights'. This panel includes a 'Select an objective:' dropdown, 'Units' (Continuous/Discrete), and input fields for 'Minimum value' and 'Maximum value'. The bottom left section, 'Node Information: Ove. objective', contains fields for 'Label', 'Name', and 'Description'. The bottom center 'Message area' shows 'Project loaded' and 'Loading hierarchy...'. The bottom right 'Working area: Load example projects' lists three projects with 'Download paper' links for each.

Figura 2.1

- **Jerarquía de objetivos:** En esta sección se muestra en formato svg el árbol de la jerarquía de objetivos teniendo un color distinto para mostrar las alternativas de las ramas.

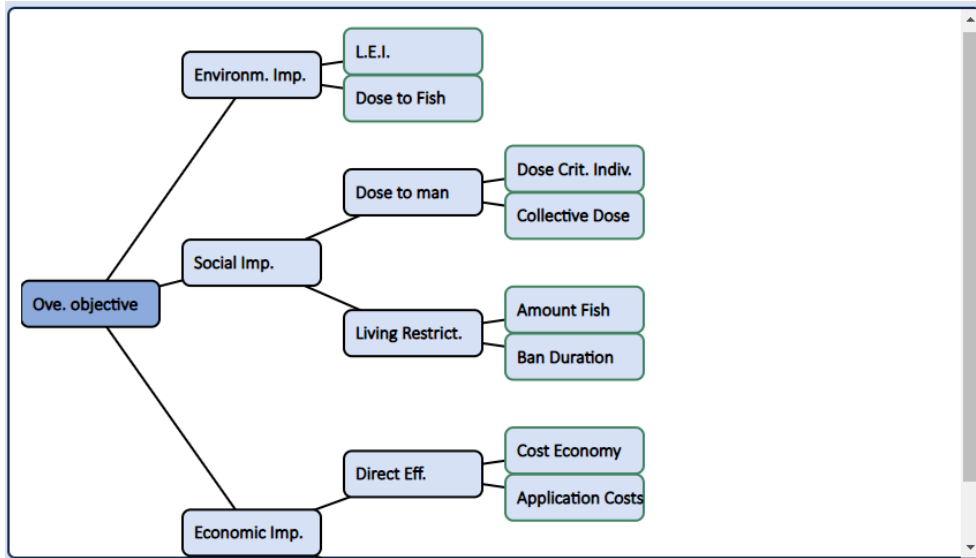


Figura 2.2

- **Área de información del proyecto:** Muestra datos generales del proyecto de decisión en curso, incluyendo el resumen de atributos considerados, funciones de utilidad definidas, pesos asignados, el ranking actual de alternativas y accesos a resultados de sensibilidad.

Figura 2.3

- **Área de trabajo:** Esta es la zona central donde aparecen ventanas o formularios según la tarea que se esté realizando, esta área permite la interacción dinámica con los elementos del modelo, mostrando controles específicos para cada etapa del proceso de decisión.

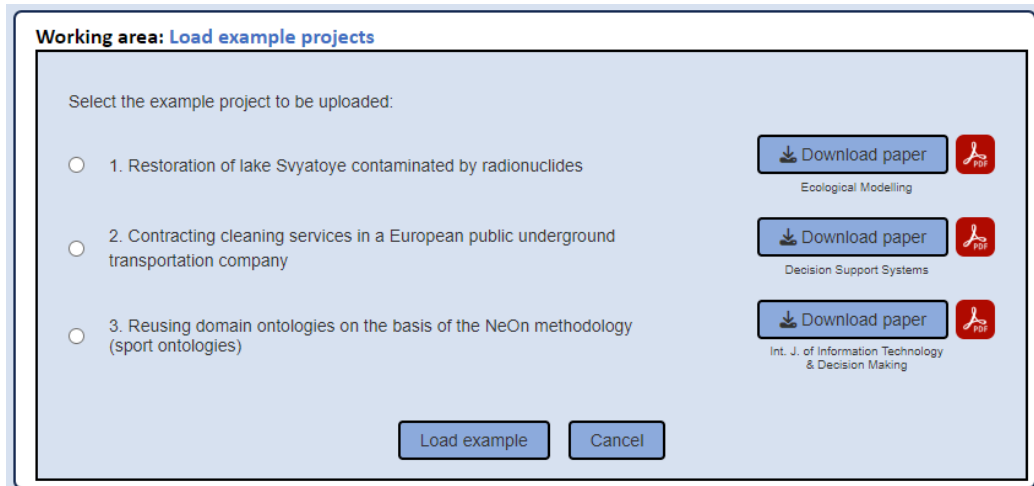


Figura 2.4

- **Área de mensajes:** Esto es el panel donde el sistema notifica información de progreso, advertencias o errores al usuario, facilitando la comunicación y guiando la correcta utilización de la herramienta.

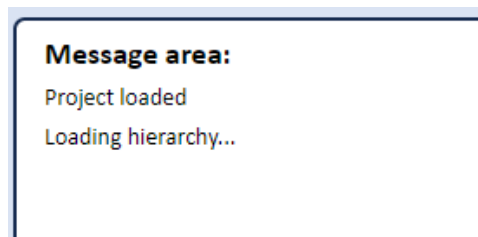


Figura 2.6

- **Panel de detalles del nodo:** Aquí se muestra toda la información detallada del elemento seleccionado en la jerarquía de objetivos.

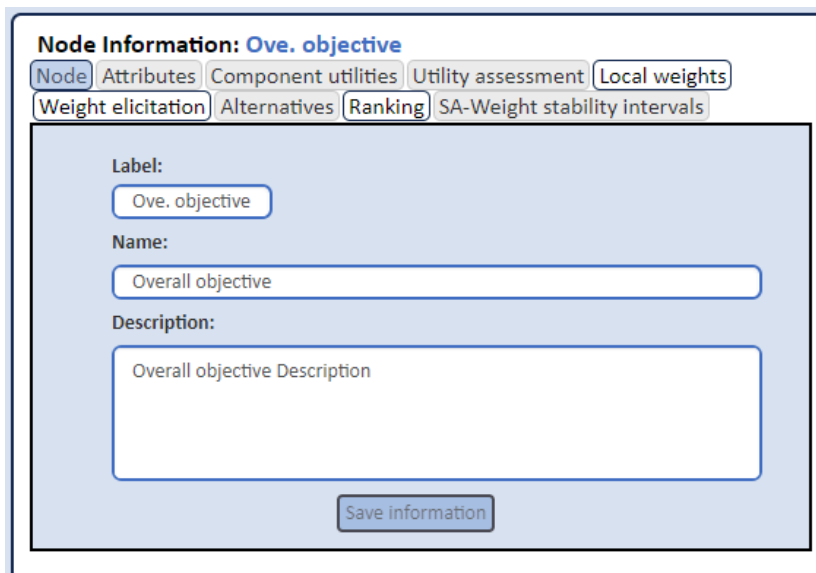


Figura 2.6

El núcleo funcional de WEB-MAUT-DSS se organiza conforme al ciclo del análisis de decisiones, el menú principal de la aplicación tiene un par de botones, el botón project deja al usuario la posibilidad de crear un proyecto, cargar un proyecto que tengas en local, cargar un proyecto de ejemplo, guardar un proyecto o cerrar un proyecto, luego está el botón alternatives que te deja cargar y guardar alternativas, por último tenemos el botón de about que muestra los creadores y colaboradores de la aplicación.

2.3.3 Funcionalidades de WEB-MAUT-DSS

A continuación se describirán las funcionalidades principales del WEB-MAUT-DSS siguiendo las etapas típicas del proceso de una decisión multicriterio:

- **Identificación del problema:** En esta fase inicial se define el contexto de la decisión a tomar, para ello el usuario crea un nuevo proyecto en la herramienta y describe la situación problemática, los objetivos generales perseguidos y los decisores involucrados, aquí se delimita el alcance del análisis, que decisión se debe apoyar y bajo qué condiciones se tomará. Aunque WEB-MAUT-DSS no impone un formato estricto para esta descripción, proporciona un espacio para documentar la motivación y objetivos globales del estudio, lo cual servirá de referencia en las etapas subsiguientes.
- **Estructuración del problema:** Una vez se ha establecido el objetivo global, se procede a estructurar el problema en forma de una jerarquía de objetivos y atributos, esto se descompone en un árbol jerárquico donde los nodos hojas representan los atributos medibles que servirán para la evaluación de alternativas. En esta etapa, WEB-MAUT-DSS guía al usuario a través de ventanas de diálogo donde introduce cada objetivo y atributo, luego, la jerarquía resultante se visualiza claramente en la interfaz, así el sistema garantiza que el decisor especifique sus objetivos y cómo se van a medir, modelizando así el problema de decisión de manera transparente y organizada.
- **Cuantificación de preferencias:** Cuando ya se ha definido la jerarquía toca capturar las preferencias del decisor respecto a cada criterio, para ello WEB-MAUT-DSS cuenta con funcionalidades para la evaluación de utilidades parciales y la elicitación de pesos haciendo que para cada atributo el decisor especifique una función de utilidad el cual refleja su grado de preferencia a lo largo del rango de posibles consecuencias de este criterio. Así el sistema WEB-MAUT-DSS traduce las preferencias subjetivas del decisor en parámetros cuantitativos conforme a la teoría MAUT, construyendo el modelo multiatributo personalizado para el problema en cuestión.
- **Evaluación de alternativas:** En el momento en el que se completa el modelo de decisión el sistema procede a calcular la utilidad multiatributo agregada para cada alternativa. El sistema WEB-MAUT-DSS multiplica, para cada alternativa, la utilidad alcanzada en cada atributo por el peso de dicho atributo, sumando luego estos productos en toda la jerarquía, quedando normalizados en una escala 0-1, donde 1 corresponde a la alternativa ideal y el 0 a la más indeseable. A partir de esta información la herramienta tiene múltiples rankings y gráficas para visualizar la información claramente y ver cual es la mejor decisión.
- **Análisis de sensibilidad:** Ya que toda decisión está sujeta a incertidumbres e imprecisiones, WEB-MAUT-DSS integra robustas

herramientas de análisis de sensibilidad para explorar la estabilidad de la solución obtenida, para comprobarlo el usuario puede llevar a cabo un análisis de sensibilidad clásico modificando gradualmente parámetros como los pesos de los criterios o ciertos valores de utilidad, observando cómo varía el ranking o la utilidad de las alternativas. Además del análisis de sensibilidad clásico el sistema WEB-MAUT-DSS también puede hacer análisis hipotéticos donde el usuario puede definir distintos supuestos, un ejemplo de esto es que el usuario puede realizar simulaciones Monte Carlo sobre los pesos o las utilidades.

2.4 Lenguaje de R y su librería Shiny

2.4.1 Historia y características principales de R

El lenguaje R es un entorno de programación enfocado principalmente al análisis estadístico y gráfico de datos, este fue creado inicialmente en 1993 por Ross Ihaka y Robert Gentleman en la Universidad de Auckland como una reimplementación del lenguaje S desarrollado en los Bell Labs [25]. Desde sus inicios R ha sido el principal lenguaje en la materia de investigación científica y en campos como machine learning, la minería de datos, la econometría, la bioinformática y la inferencia estadística. Todo esto gracias a su amplia colección de bibliotecas que extienden sus capacidades tanto en cálculos numéricos como en visualización gráfica.

Entre las principales características de R destacan su enfoque estadístico y que es un software libre y multiplataforma. R forma parte del proyecto GNU y se distribuye bajo licencia GPL, estando disponible para sistemas Windows, macOS y Linux. En esencia R ofrece un entorno potente para el análisis de datos, combinando su sencillez de uso con la robustez de un lenguaje estadístico altamente extensible.

2.4.2 La librería Shiny

Shiny es una librería del ecosistema R que facilita la creación de aplicaciones web interactivas directamente desde R, su propósito principal es permitir a los usuarios convertir análisis y visualizaciones hechos en R en aplicaciones web reactivas sin usar las herramientas típicas de desarrollo web. En una aplicación Shiny hay constantemente una interacción entre usuario y servidor ya que mientras el usuario manipula ciertos inputs en la interfaz la aplicación responde actualizando los outputs de acuerdo con las reglas lógicas definidas.

La estructura básica de una aplicación Shiny se compone de dos componentes fundamentales:

- **Interfaz de Usuario:** Define el diseño de la aplicación y los elementos visuales con los que esta interactúa.
- **Servidor:** Define toda la lógica de la aplicación, para ello recibe inputs provenientes de la interfaz de usuario y en base a ellos realiza los cálculos o procesos necesarios utilizando R y los resultados de esto son asignados a outputs reactivos de la interfaz de usuario que éste los presenta de manera actualizada.

Como ejemplo del uso de Shiny tenemos la aplicación WEB-MAUT-DSS mencionada anteriormente, ya que en todos los archivos o mockups de esta se encuentran divididos en dos funciones principales que sería la función `mockupNumero_UI` y `mockupNumero_server` también el archivo principal

app.R se encuentra dividido entre ui y server y es este archivo el que funciona como conexión de todos los mockups de la aplicación que realizan los cálculos de la aplicación.

3 Desarrollo de las Funcionalidades

3.1 Implementación de la carga de proyectos

3.1.1 Requisitos funcionales de la carga de proyectos

La primera parte de la interfaz de usuario de la carga de proyectos consistía en añadir un nuevo botón desplegable al pulsar el botón “Project” con el nombre “Load Example” como se muestra en la figura 3.1 y enlazar este botón con el mockup 49 para que cuando se pulse este botón se abra el mockup 49 el cual se ve como muestra la figura 3.2.

3.1.2 Interfaz de usuario de la carga de proyectos

La primera parte de la interfaz de usuario de la carga de proyectos consistía en añadir un nuevo botón desplegable al pulsar el botón “Project” con el nombre “Load Example” como se muestra en la figura 3.1 y enlazar este botón con el mockup 49 para que cuando se pulse este botón se abra el mockup 49 el cual se ve como muestra la figura 3.2.

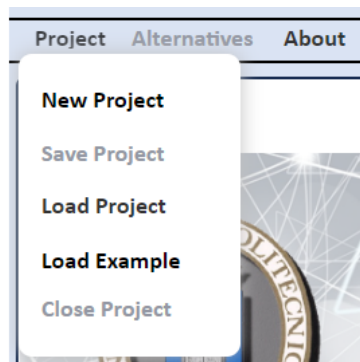


Figura 3.1

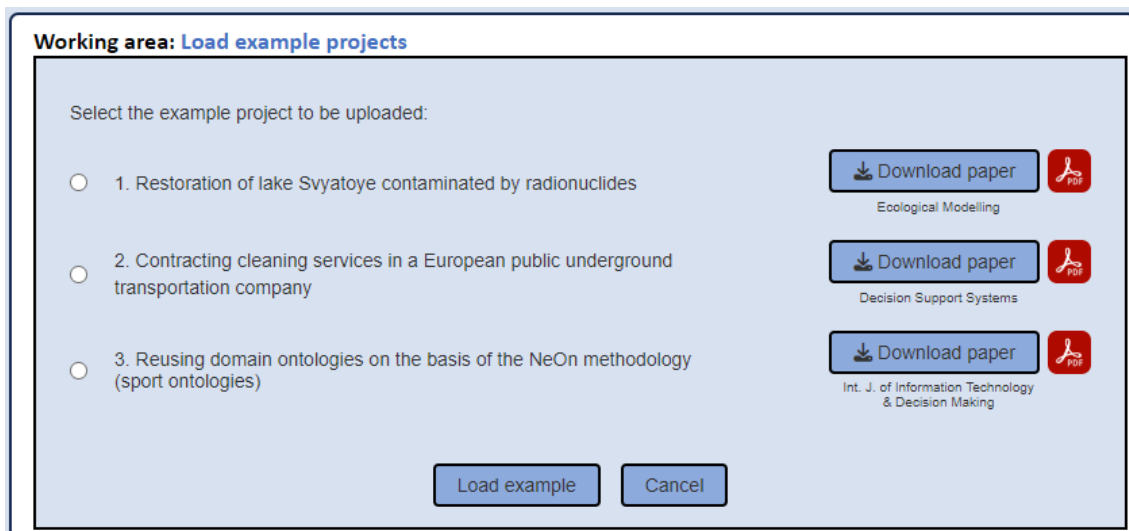


Figura 3.2

Como se puede observar en la figura 3.2 esta interfaz consta de tres radiobutton los cuales representan los tres papers de ejemplo que proporciona la aplicación, cada opción tiene también la opción de descargar un pdf del paper en cuestión en el botón “Download paper” y luego tenemos los dos botones inferiores los cuales por un lado el botón “Cancel” cierra el mockup 49

por completo, por el otro lado el botón “Load Example” carga el proyecto seleccionado en la aplicación como muestra la figura 3.3.

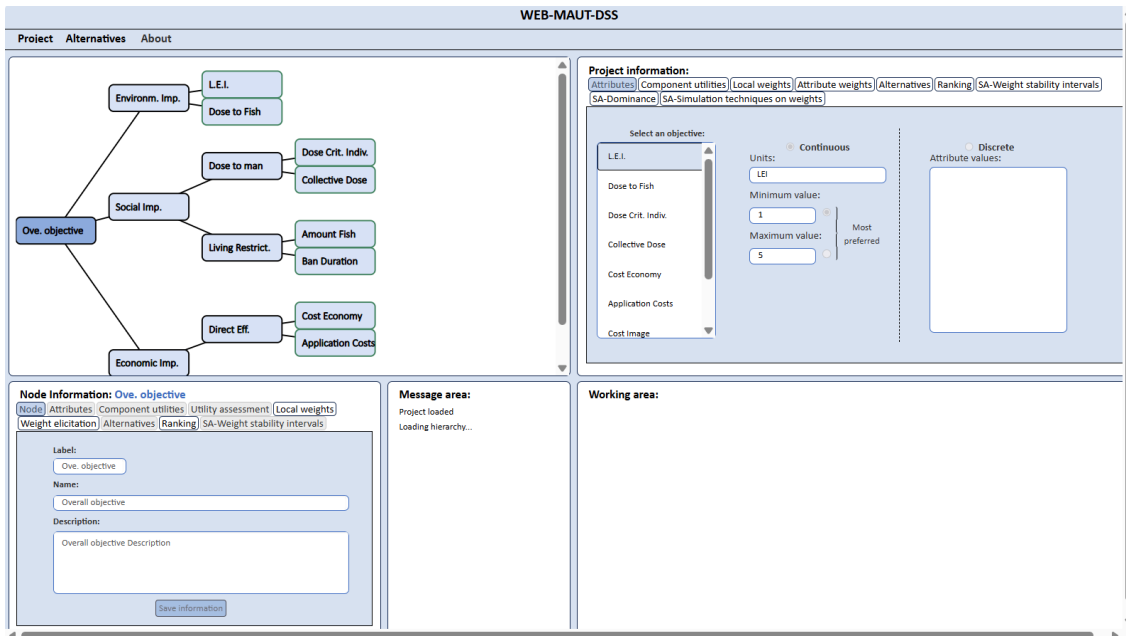


Figura 3.3

En la figura 3.3 se puede observar el resultado de pulsar el botón “Load Example” cuando tenemos seleccionada la primera opción. También se puede ver en la figura 3.3 que al pulsar el botón “Load Example” se cierra automáticamente el mockup 49 y se carga la jerarquía de objetivos del primer ejemplo.

3.1.3 Servidor de la carga de proyectos

Para el funcionamiento correcto del servidor lo primero que se hizo fue el añadir a la carpeta www una subcarpeta de pdfs donde se guardan los papers de ejemplo del mockup, estos pdf se llaman 2013_IJITDM, 2007_DSS y 2006_ECOL_MOD y una carpeta examples donde se guarda la jerarquía de objetivos que se cargaran en el mockup, estos archivos se llaman 1.Svyatoye, 2.Lisbon_metro y 3.Ontology_sport. Esto se visualiza correctamente en la jerarquía de la figura 3.4.

```

\---www
  |
  |   mockup49.R
  |
  +---examples
  |   1.Svyatoye.Rdata
  |   2.Lisbon_metro.Rdata
  |   3.Ontology_sport.Rdata
  |
  \---pdfs
      2006_ECOL_MOD.pdf
      2007_DSS.pdf
      2013_IJITDM.pdf
  
```

Figura 3.4

En cuanto a la implementación del servidor está dividida en tres partes, la primera es la implementación del botón de “Download paper” el cual descarga el paper de la sección que hayas pulsado, para ello se le da primero al downloadHandler el filename del paper seleccionado y luego en el content se

utiliza la función `file.copy()` haciendo referencia al path del archivo en cuestión con la función `file.path()` y con esto se descarga el archivo seleccionado.

La segunda parte del servidor del mockup 49 consiste en lo que sucede cuando se pulsa el botón de “Load Example”, realmente el funcionamiento es prácticamente idéntico a cuando se pulsa en el desplegable de “Project” el botón “Load Project” ya que primero se inicializa el árbol y el proyecto, luego se mira en los radiobutton para ver cual de todos está seleccionado y se selecciona el path del radio button seleccionado una vez seleccionado el path se llama a la auxiliar `load()` con el path y se carga el proyecto completo para finalmente actualizar todos los paneles activandolos todos y desactivando el panel de `working_area_tab`, finalmente se habilita el botón de “Save Project” y el botón de “Close Project” poniendo en el panel de mensajes un mensaje de que el proyecto se ha cargado.

Por último la tercera parte del servidor del mockup 49 consiste en la funcionalidad del botón “Cancel”, este básicamente consiste en actualizar el `tabset panel` con la función `updateTabsetPanel` pero seleccionando en `working_area_tab` como inactivo.

3.2 Implementación de los diagramas

3.2.1 Requisitos funcionales de los diagramas

La primera parte de la interfaz de usuario de la carga de proyectos consistía en añadir un nuevo botón desplegable al pulsar el botón “Project” con el nombre “Load Example” como se muestra en la figura 3.1 y enlazar este botón con el mockup 49 para que cuando se pulse este botón se abra el mockup 49 el cual se ve como muestra la figura 3.2.

3.2.2 Interfaz de usuario de los diagramas

Para comenzar los diagramas la primera parte fue el relacionarlo con el botón de “Ranking” dentro del project information que se muestra en la figura 3.5 y dentro de “Ranking” había que relacionarlo con la opción de “Select chart” que “Comparative bars” y darle a al botón watch todo esto ilustrado en la figura 3.6.

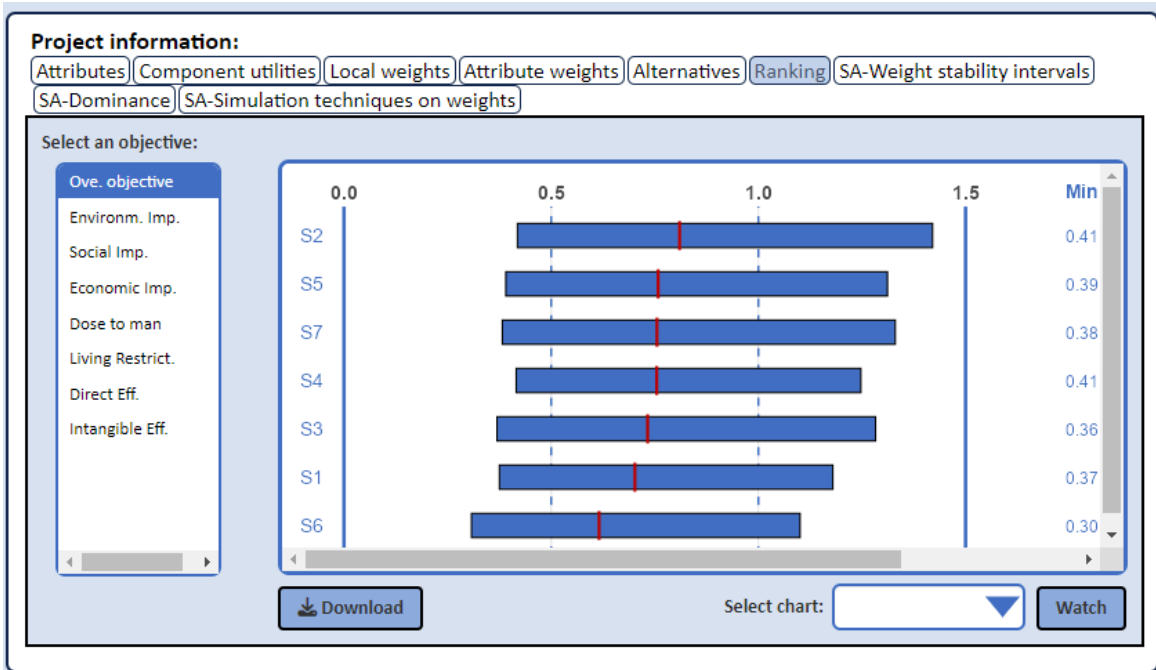


Figura 3.5

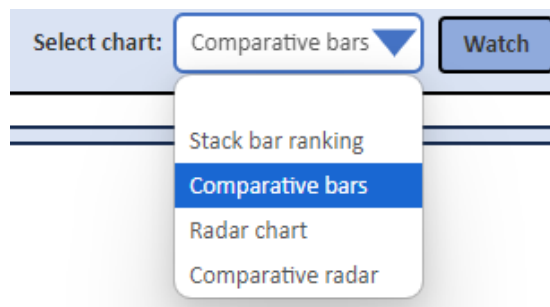


Figura 3.6

Una vez seleccionado el ranking de barras se abrirá el mockup 47 que contiene también la información del mockup 48 y se ve como en la figura 3.7. Estos constan de dos desplegables de color azul y rojo respectivamente centrados para seleccionar las alternativas a comparar que se muestra en detalle en la figura 3.8, un gráfico debajo de los desplegables que se explicará detalladamente que es lo que muestra cada uno en las siguientes secciones, luego consta de el botón “OK” para cerrar la `working_area_tab` una vez finalizado y el botón de download que permite descargar el gráfico en formato imagen.



Figura 3.7

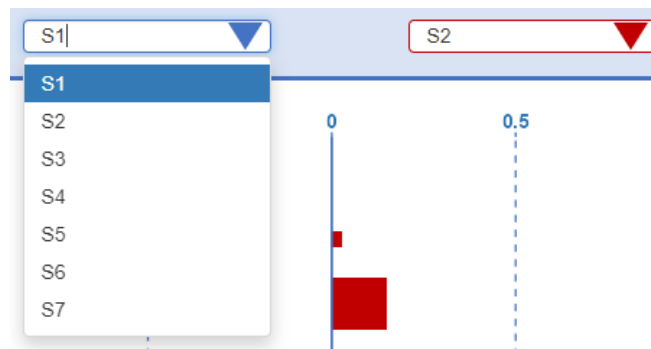


Figura 3.8

3.2.3 Servidor común

Este servidor a diferencia del servidor anterior se divide en dos partes, la parte que es común para los dos mockups y la parte de gráficas y cálculos que es única para cada uno.

En la parte común encontramos dos funciones auxiliares, la primera función llamada `get_alt_names` que sirve para coger el nombre de las alternativas para los desplegables y la segunda función llamada `replicate_df` se utiliza para poner el peso proporcional en un dataframe.

Dentro del servidor encontramos las siguientes funciones comunes, la primera es la función del botón "OK" que básicamente al igual que el botón "Cancel" en el mockup anterior sirve para cerrar toda la zona de `working_area_tab`.

La segunda función común es la del desplegable que actualiza de manera reactiva el gráfico en cuestión cuando cambian los valores de uno de los dos desplegables para ello cuando cambia un valor de las alternativas llamará a la función de `plot data calc()` y dependiendo si existe o no simulación llamará a una función u otra de las de dibujar la gráfica.

La tercera es el apartado interactivo de la gráfica ya que cuando se le hace click a la gráfica muestra otra vez esa gráfica pero todos los elementos en una imagen como se muestra en la figura 3.9 para ello primero se llama a `showModal()` donde aquí llamaremos a las funciones de dibujo del gráfico necesario para mostrarlo en una sola imagen el gráfico completo.

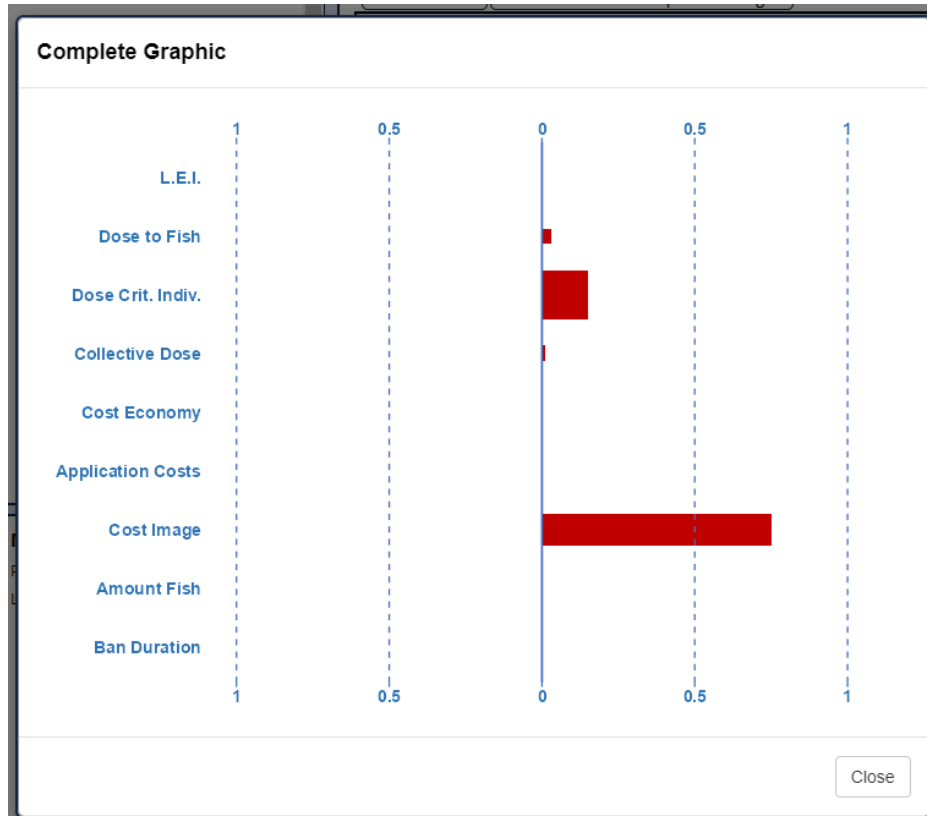


Figura 3.9

De manera similar al zoom funciona el apartado del botón de “Download”, este botón básicamente crea una imagen con un tamaño dinámico adaptándose al número de alternativas del proyecto en cuestión esto lo podemos ver reflejado en las figuras 3.10 y 3.11 donde podemos ver que al tener la figura 3.10 menos alternativas el gráfico se ve mas grande que en la figura 3.11.

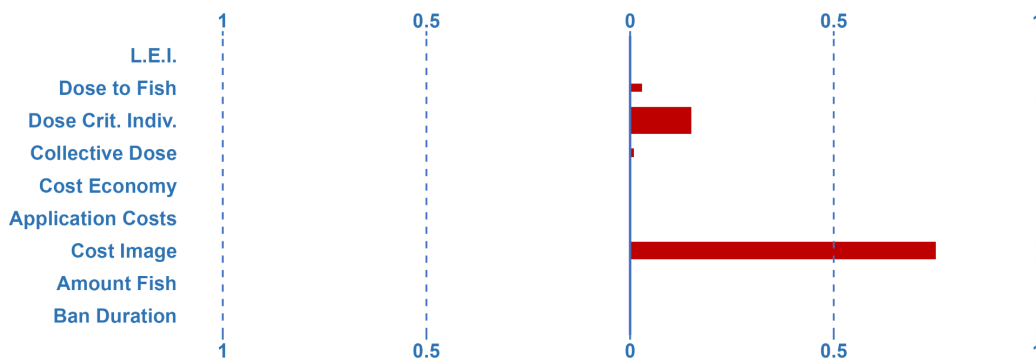


Figura 3.10

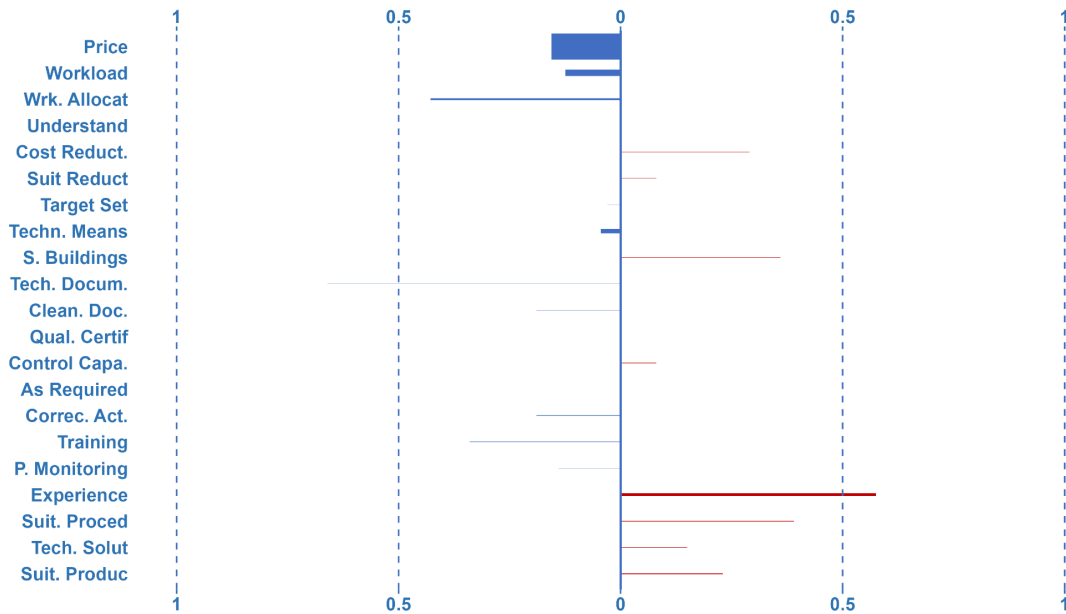


Figura 3.11

Para lograr esto primero se crea el archivo con el downloadHandler y se le pone como nombre comparison_plot.png, una vez creado dependiendo si tiene o no simulación se llama a la función de dibujo correspondiente a ese mockup.

3.2.4 Gráfico del diagramas de barras de comparación

El primer gráfico es un diagrama de barras comparativo ya que no se realiza ninguna simulación, en este gráfico el que el alto de las barras lo determina el peso que lo podemos encontrar en la figura 3.13 y lo largo es la diferencia entre las dos utilidades de las alternativas que podemos encontrar las utilidades en la figura 3.14, si el gráfico va hacia la izquierda y es de color azul entonces es mejor la alternativa S3 y si el gráfico va hacia la derecha y es de color rojo es mejor la alternativa S4, todo esto se puede observar en la figura 3.12.



Figura 3.12

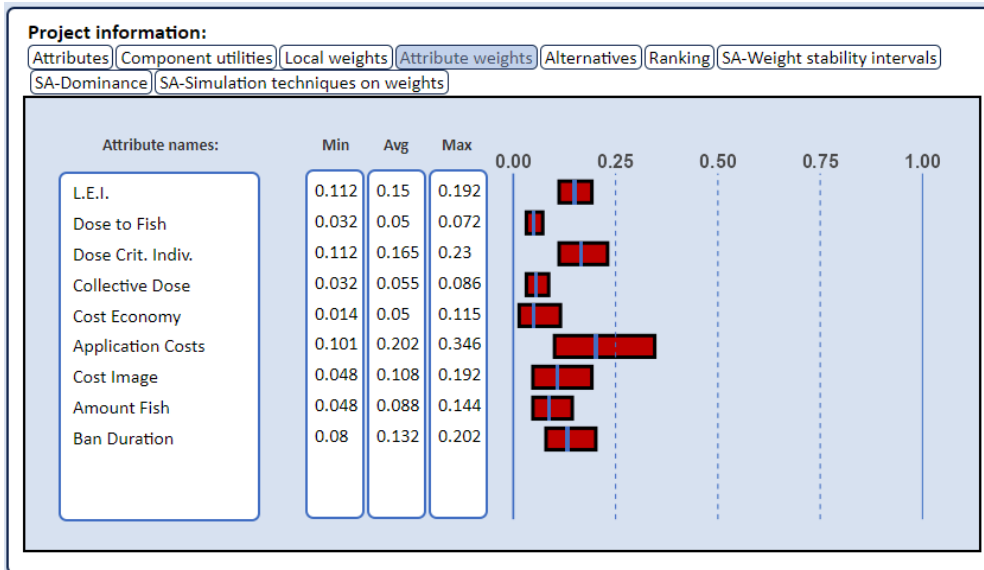


Figura 3.13

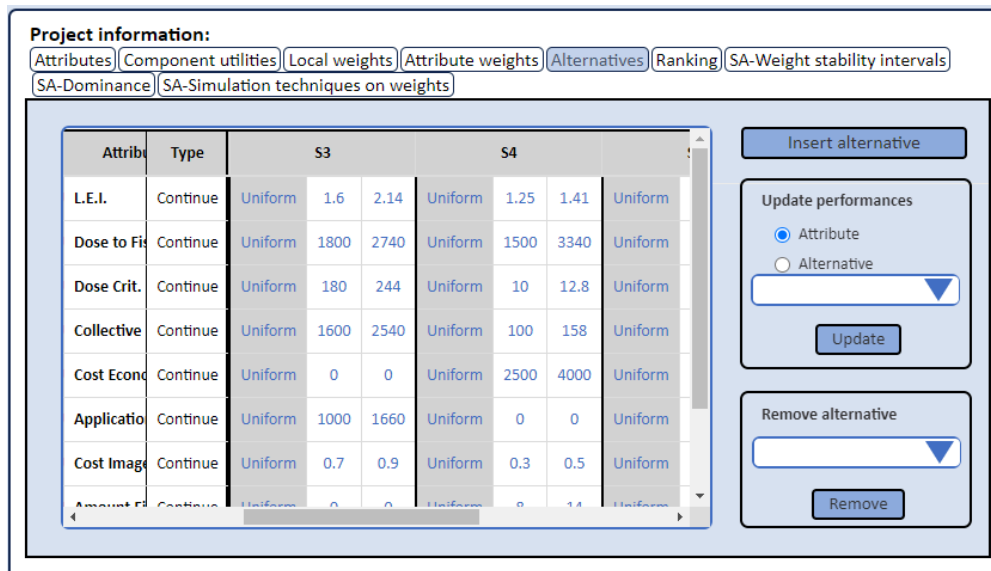


Figura 3.14

3.2.5 Servidor del diagramas de barras de comparación

La parte no común del servidor de este archivo son básicamente dos funciones, la primera es la función que calcula todos los datos necesarios llamada plot data calc() y la segunda es la función para pintar la gráfica que se llama generate_comparison_plot.

La función plot data calc() es una función reactiva que lo primero que se hace es mirar qué alternativas están siendo seleccionadas y luego se asegura de que no sean la misma, una vez mirado eso verifica que no haya simulación y obtiene los nodos y los nodos hoja, cuando obtiene estos datos crea un dataframe llamado df_data que contendrá los siguientes datos, hoja, peso, utilidad alternativa 1, utilidad alternativa 2 y diferencia entre utilidades. Una vez creado el dataframe se procede a calcular los pesos de cada hoja usando la función getAttributeWeights(tree, node_label) recorriendo los pesos del primer nodo y haciendo un vector de ello, una vez creado el vector pasamos a recorrer la lista de hojas para relacionarlas en orden con los pesos obtenidos en la lista

de `weight_list`. En el momento que tenemos todos los pesos pasamos a calcular las utilidades de todas las hojas para las dos alternativas para ello primero creamos dos listas, una por alternativa donde se muestran la lista de hojas para cada alternativa, una vez creado hacemos un bucle para cada nodo donde usaremos la función `getGlobalUtilities(tree,node_label,alternative)` para las dos alternativas y lo vamos guardando en dos vectores `v1` y `v2`, finalmente relacionamos la lista de utilidades creado anteriormente entre sus hojas y sus utilidades. Por último cuando hemos hecho todos los cálculos creamos el dataframe final el cual recorreremos la lista de hojas y vamos haciendo la diferencia entre la utilidad 1 y la utilidad 2 y construimos el dataframe final llamando a las listas creadas anteriormente por el valor de la hoja, y devolvemos así una lista que contiene un string que pone modo determinista y un dataframe completo preparado para dibujar.

La segunda parte es la de la función `generate_comparison_plot(plot_data)`, aquí se recibe el dataframe de `plot_data` `calc()` y lo primero que hacemos es coger el nombre de las hojas, poner los pesos a 0 si existe un valor nulo y poner la dirección del plot dependiendo si la diferencia de utilidad 1 - utilidad 2 es o no un número negativo ya que si es positivo es mejor la alternativa 1 y si es negativo es mejor la alternativa 2, una vez hecho esto creamos un `ggplot` con el dataframe `plot_data` donde la `x` es la diferencia de utilidades, la `y` son las hojas y se rellena en la dirección marcada por sí es positivo o negativo el valor de la diferencia de utilidades, por último ponemos el peso máximo como la altura máxima de cada hoja haciendo que el resto sean proporcionales. Una vez realizado esto ponemos los ejes coloreados que tiene el eje y los valores 1, 0.5, 0, 0.5, 1 con el color azul y la línea discontinua en cada eje, también ponemos los colores para las hojas y los colores para las barras poniendo el azul cuando va hacia la izquierda del eje y el rojo si va hacia la derecha del eje.

3.2.6 Gráfico del diagrama de cajas

El segundo gráfico es un diagrama de cajas, que es lo que se muestra cuando se realiza la simulación Monte Carlo, en este ejemplo únicamente se han simulado 10 escenarios. En el gráfico de la figura 3.15 podemos encontrar que el alto de las cajas dependen del peso del atributo que lo podemos observar en la figura 3.16 y el largo va determinado por las utilidades que se pueden observar en la figura 3.17 la alternativa offer 1 se representa con el color azul y arriba de la alternativa offer 2 que se representa con el color rojo, además se puede observar que la gráfica tiene una escala del 0 al 1.



Figura 3.15

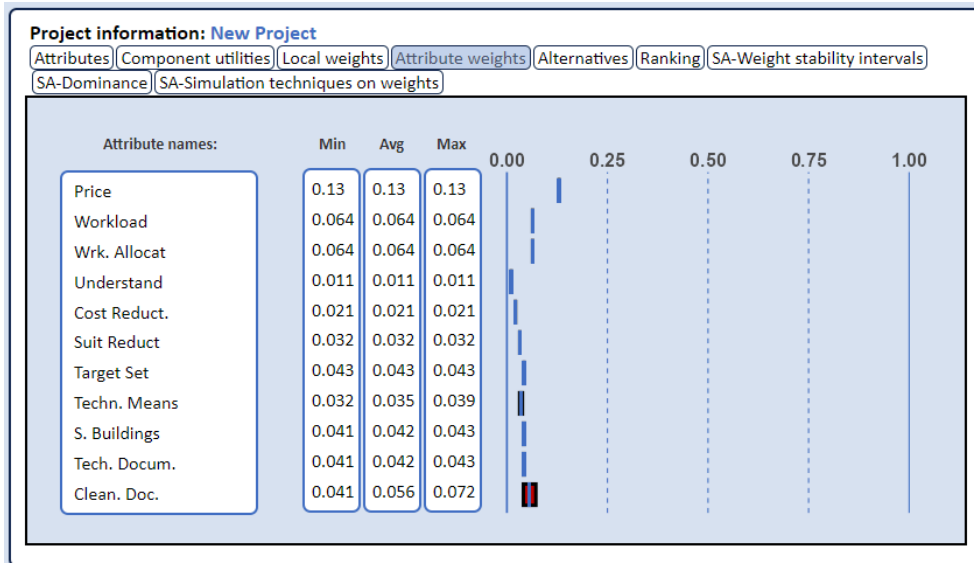


Figura 3.16

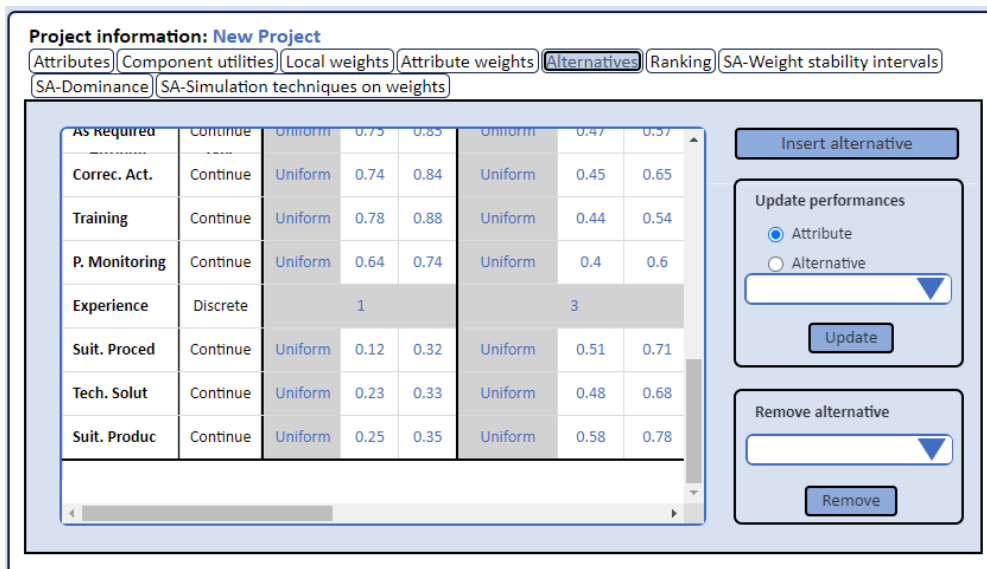


Figura 3.17

3.2.7 Servidor del diagrama de cajas

En esta parte la parte no común consiste en dos funciones, siendo la primera al igual que en el mockup 47 la función que calcula todos los datos necesarios que se llama `plot_data_calc()` y la segunda es la función al igual que en el mockup anterior también sirve para pintar la gráfica que se llama `draw_mc_boxplot(plot_df)`.

Al igual que en el mockup anterior la función `plot_data_calc()` sigue siendo una función reactiva que primero mira las alternativas seleccionadas asegurándose de que no sean la misma, en el momento que se asegura de esto procede a verificar que exista simulación en el árbol y si comienza a calcular primero el número de simulaciones a realizar y posteriormente las cogerá las hojas y creará similar a la función anterior creará una lista de pesos que procederá a relacionarlos con las hojas, para obtener los pesos utilizará la función `getAttributeWeights(tree, node_label)` en el nodo padre y luego relacionará en orden de obtención con el orden de obtención de las hojas consiguiendo la

relación entre el peso y las hojas, en esta función para calcular las utilidades se hará en dos dataframes con los valores, hoja, peso, utilidad, alternativa posteriormente se utiliza una función auxiliar `build_long(alt, alt_lab)` que básicamente mete el peso en el dataframe y además utiliza la función `getUtilitySimul(tree, leaf_label, alternative, num_simulations)` para realizar las simulaciones y para cada simulación se usa una fila del dataframe, esta auxiliar la utilizamos para construir los dos dataframes, uno por cada alternativa rodeado por un try catch para asegurar que la aplicación no deje de funcionar, finalmente se cogerá el máximo peso y se utilizará la auxiliar mencionada anteriormente `replicate_df(df_long, max_w)` para poner el peso proporcional para cada elemento dependiendo del peso máximo, finalmente se combinarán los dataframes en un unico dataframe poniendo el valor de alternative del dataframe primero a “Alt1” y “Alt2” dependiendo de que dataframe se trate y se combinarán en un unico dataframe para devolver una lista con primero el mode “simulation” y el dataframe completo.

Al igual que la anterior la segunda parte no común del servidor es la función para crear la gráfica que aquí se llamará `draw_mc_boxplot(plot_df)`, al igual que en la función anterior lo que haremos aquí es crear un ggplot con el dataframe `plot_df` donde la x es el valor de la utilidad y la y al igual que la anterior son las hojas luego el relleno a lo alto es el campo de las alternativas ya que la alternativa 1 y la alternativa 2 ocupan el mismo espacio, una vez hecho eso se crearán los ejes que en este caso es un eje con línea discontinua en los valores 0, 0.25, 0.5, 0.75, 1 con el color azul y con los números del eje tanto arriba como abajo de este, finalmente se pondrán los valores con “Alt1” en color azul y los valores con “Alt2” con el color rojo.

3.3 Implementación de la generación de informes

3.3.1 Requisitos funcionales de la carga de proyectos

La primera parte de la interfaz de usuario de la carga de proyectos consistía en añadir un nuevo botón desplegable al pulsar el botón “Project” con el nombre “Load Example” como se muestra en la figura 3.1 y enlazar este botón con el mockup 49 para que cuando se pulse este botón se abra el mockup 49 el cual se ve como muestra la figura 3.2.

3.3.2 Interfaz de usuario de la generación de informes

La interfaz de usuario del mockup 46 es la más simple hasta el momento y consta de 10 checkbox para generar reportes en función de las opciones seleccionadas o no seleccionadas como se ve en la figura 3.18

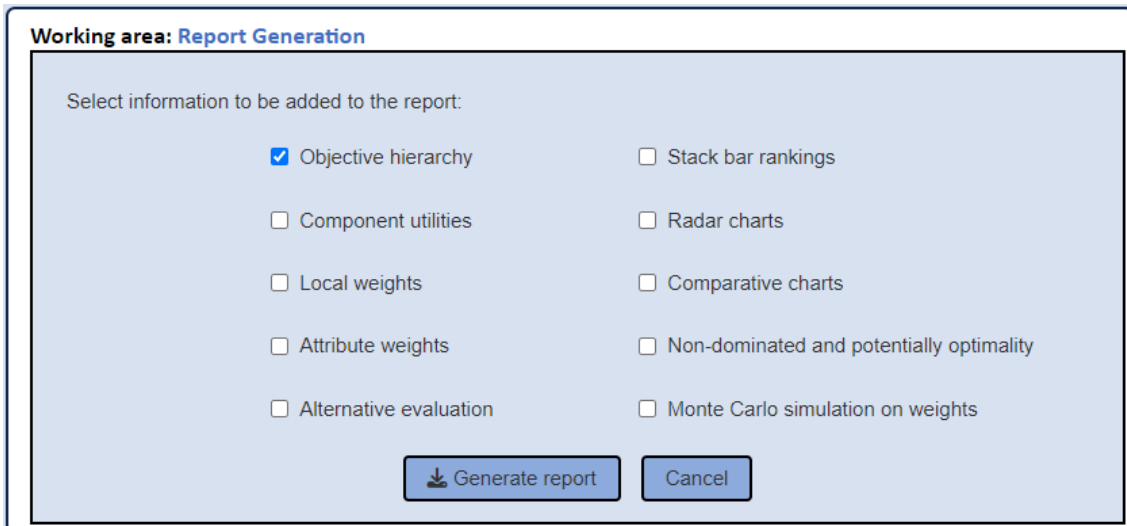


Figura 3.18

También tenemos dos botones bastante descriptivos que son el botón de “Generate Report” que genera un reporte en base a lo seleccionado en formato docx y el botón “Cancel” que al igual que en el mockup 49 simplemente cierra el mockup en cuestión.

3.3.3 Servidor de la generación de informes

Para al servidor de el mockup 46 encontramos un par de funciones auxiliares que serían la función `is_leaf(id, g)` para saber si un nodo es o no un nodo hoja, luego la función `attr_details(id, g)` que básicamente primero mira si un nodo hoja es discreto o continuo y luego en base a eso crea una lista con los elementos del nodo que queremos mostrar, por último en las auxiliares encontramos la función `generate_report_content` que básicamente sirve para generar el contenido completo del report en cuestión y es esta la que llama a `attr_details(id, g)` si es necesario.

Luego encontramos también la función del botón “Cancel” que al igual que en los mockups anteriores básicamente pone el `working_area_tab` a `inactive` haciendo que se cierre.

Por último tenemos la función `generate_report()` con el `downloadHandler` que primero crea un archivo `.docx` con el nombre `project_report.docx` y luego en cuanto al contenido vemos que se utiliza la librería de R `studio office` para crear un `docx`, para ello primero se crea un `doc` con `read_docx()` y luego se crea el título `WEB-MAUT-DSS Project Report` centrado como se muestra en la figura 3.19.

WEB-MAUT-DSS Project Report

OBJECTIVES HIERARCHY AND ATTRIBUTES

Figura 3.19

Posteriormente se realizan las validaciones de que checkbox está igualada a `true`, si la checkbox “Objective hierarchy” está pulsada entonces se mostrará el subtítulo de la figura 3.19, debajo de eso se mostrará una imagen de la jerarquía de objetivos como se muestra en la figura 3.20 la cual será igual que en la aplicación.

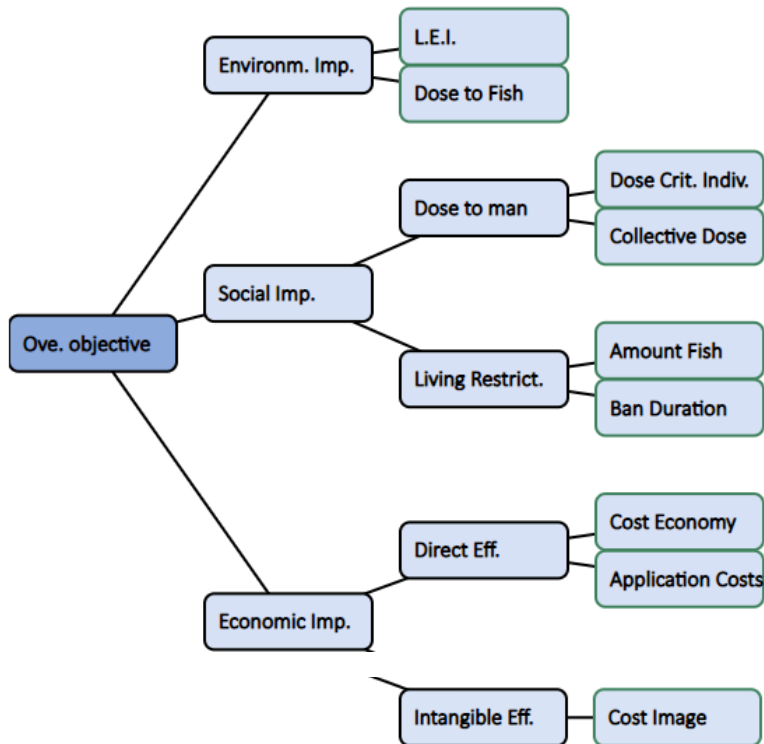


Figura 3.20

Dentro del apartado de “OBJECTIVES HIERARCHY AND ATTRIBUTES” además podemos encontrar distintos tipos de elementos, lo primero serían los nodos intermedios los cuales únicamente tendrán la información correspondiente a “Node label”, “Node name”, “Node description” y “Direct descendants” como se muestra en la figura 3.21.

Node label: Delivery Con)

Node name: Delivery conditions and human resources

Node description: Delivery conditions and human resources (Delivery Con), which accounts for how consistent and coherent the human resources allocated to the services are

Direct descendants: Workload, Wrk. Allocat, Cost optimiz

Figura 3.21

Para los nodos hoja encontramos dos tipos de atributos los continuos y discretos cada uno con sus propios parámetros, por un lado los continuos tendrán el siguiente formato “Node label”, “Node name”, “Node description”, “Attribute type”, “Attribute units”, “Minimum value”, “Maximum value” y “Most preferred value” como se muestra en la figura 3.22.

Node label: P. Monitoring

Node name: Quality performance monitoring

Node description: Quality performance monitoring (P. Monitoring), which accounts for the measurement system to be implemented, i.e., what will be inspected, how the inspection will be carried out and the measurement criteria to be used during inspection are evaluated to verify that they guarantee greater objectivity, completeness and high quality service.

Attribute type: Continuous

Attribute units: Subjective scale

Minimum value: 0

Maximum value: 1

Most preferred value: 1

Figura 3.22

Luego para los atributos discretos similar a lo anterior tenemos el siguiente formato “Node label”, “Node name”, “Node description”, “Attribute type” y “Attribute value”, igual a la figura 3.23.

Node label: Experience

Node name: Past experiences in graffiti prevention and cleanup

Node description: Past experiences in graffiti prevention and cleanup (Experience) verifies supplier competence, assessing details on how the supplier has successfully implemented these specific services for other clients

Attribute type: Discrete

Attribute values: 0, 1, 2, 3, 4

Figura 3.23

Cuando se pulsa la casilla “Component utilities” utiliza en exportMockup14(node14), que es parte del mockup 14 y como se muestra en la figura 3.24, para devolver todos los gráficos de las utilidades de las hojas uno debajo de otro como muestra la figura 3.25 en la cual vemos cómo se verían en el word con el título y la lista de gráficas de las cuales encontraremos tres gráficas por página.

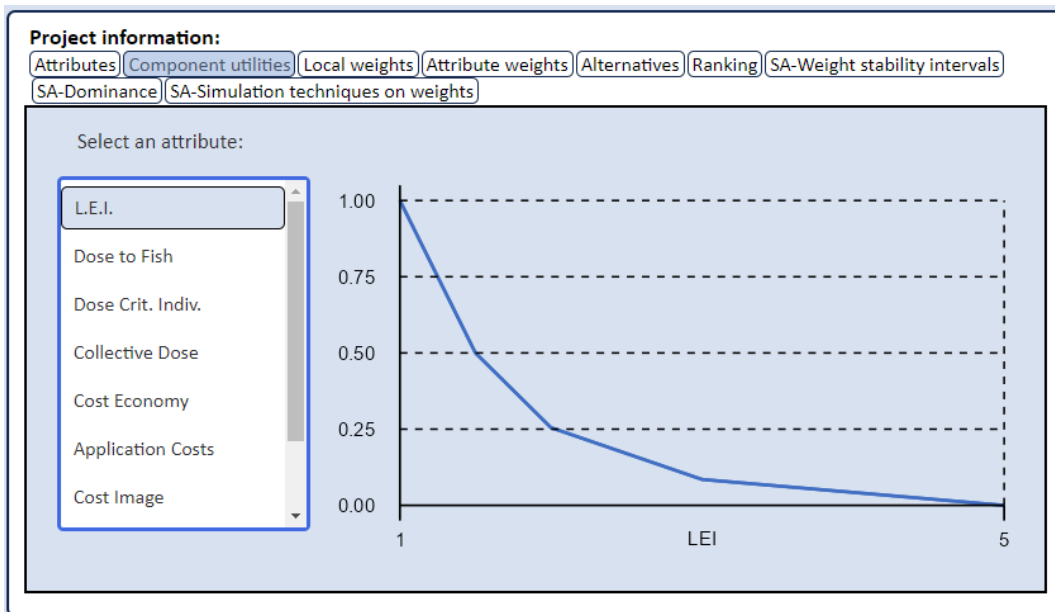
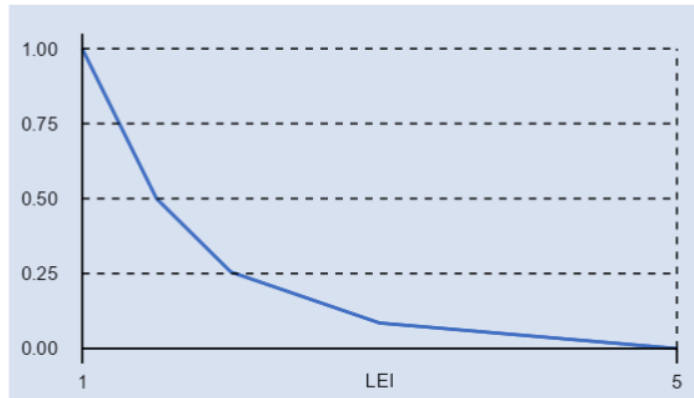


Figura 3.24

Component Utilities

Component Utilities – Node: L.E.I.



Component Utilities – Node: Dose to Fish



Figura 3.25

Para la opción de “Local weights” nos muestra todas las imágenes de los pesos locales de cada nodo usando la función `exportMockup16(node_16)` que forma parte del mockup 16 que es el que se muestra en la figura 3.26, esto corresponde a la figura 3.27 en la que se muestra la lista de gráficas, al igual que en la anterior las páginas tienen como máximo 3 gráficas antes del salto de página.

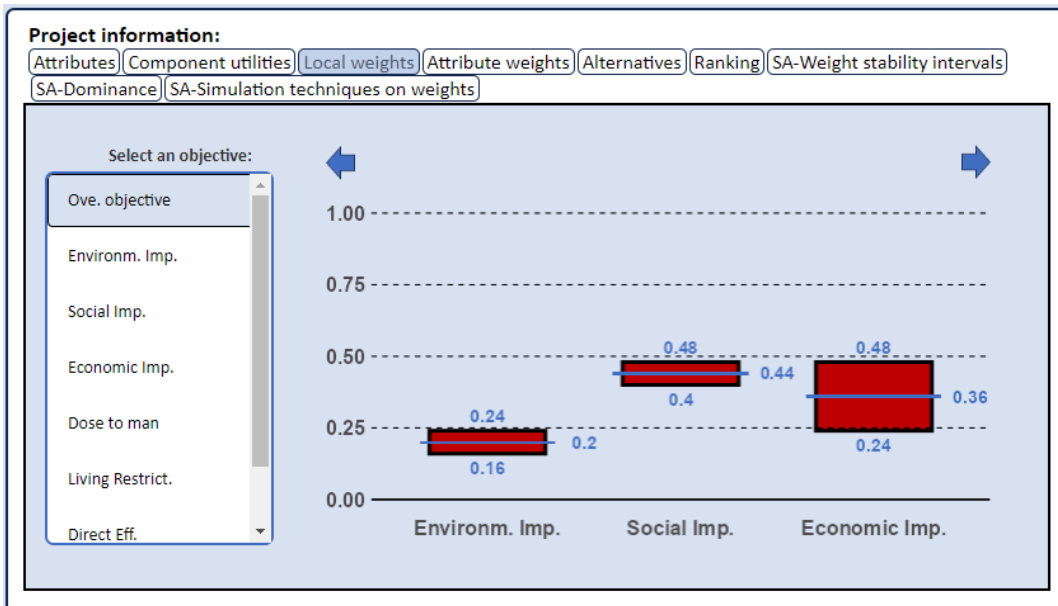
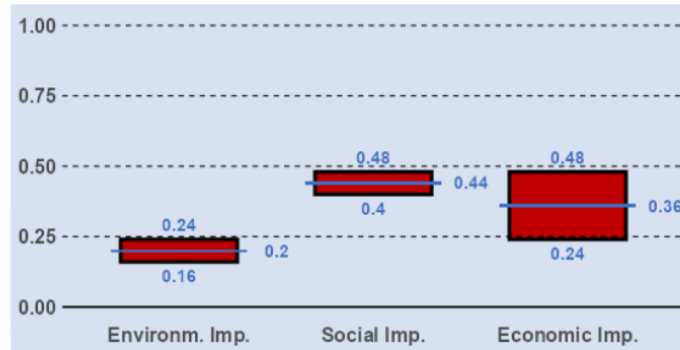


Figura 3.26

Local Weights

Local weights – Node: Ove. objective



Local weights – Node: Environm. Imp.

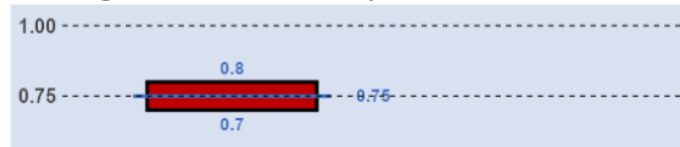


Figura 3.27

Si pulsamos el checkbox de “Attribute weights” nos mostrará dentro del word una gráfica que muestra los pesos de cada atributo o cada nodo hoja dentro del árbol como se muestra en la figura 3.26, para esto se utilizará la auxiliar exportMockup17() que corresponde al mockup 17 que se muestra en la figura 3.28 su gráfico.

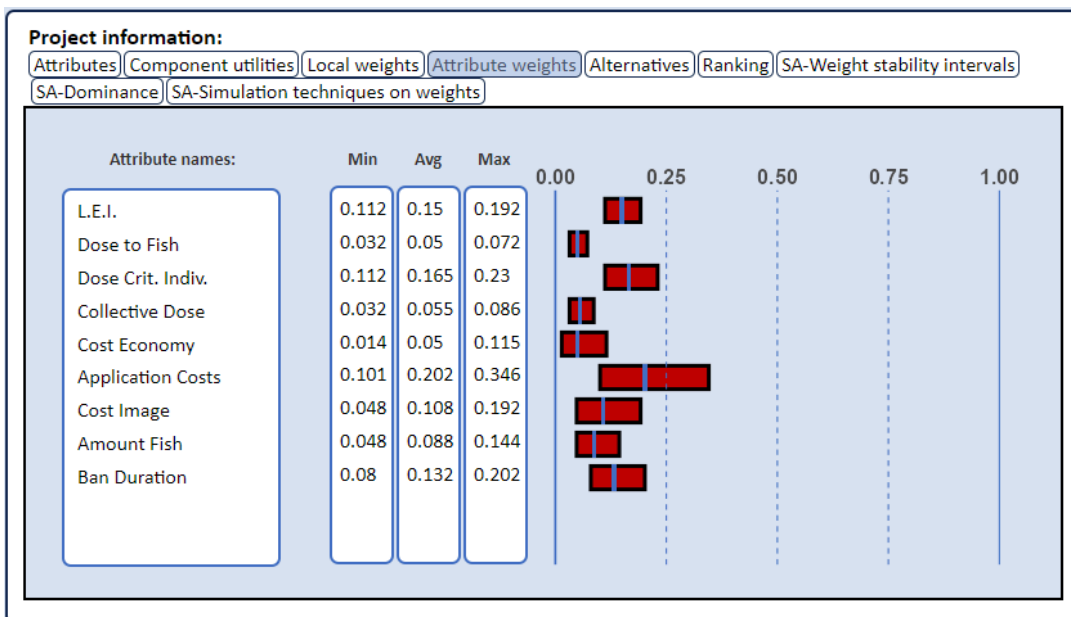


Figura 3.28

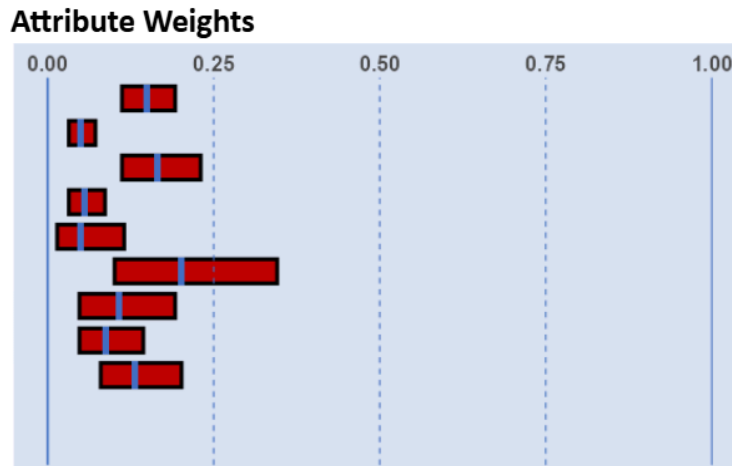


Figura 3.29

La opción “Alternative evaluation” consiste de la función exportMockup19() que se muestra en la figura 3.30 y pertenece al mockup 19 y es una evaluación de la utilidad de las alternativas, en la aplicación corresponde a la opción “Ranking” dentro del project information, se puede ver como se ve este apartado en la figura 3.31.

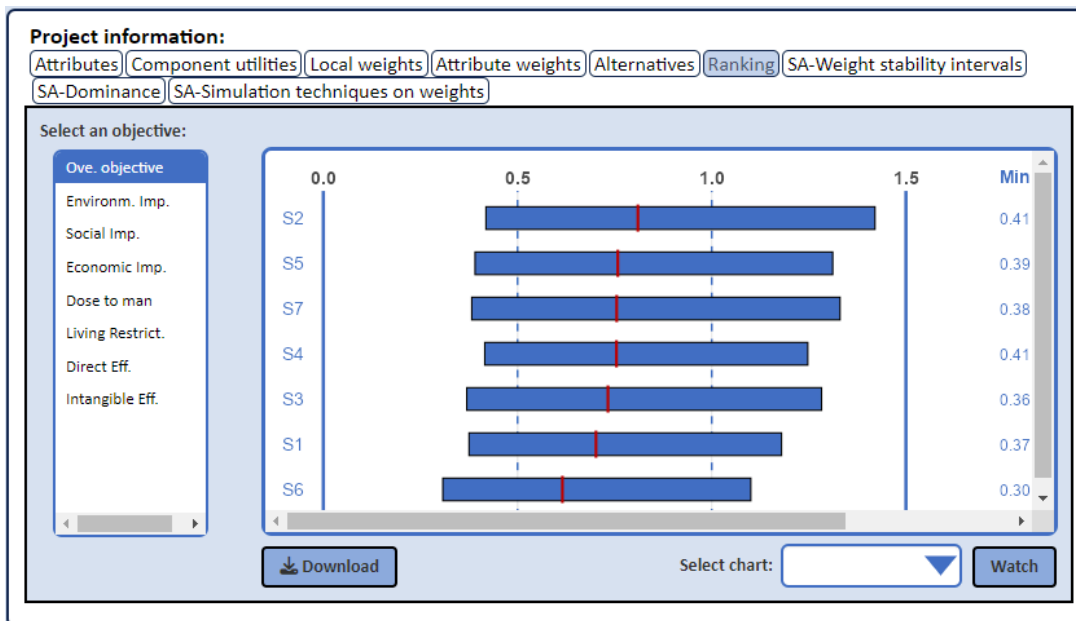


Figura 3.30

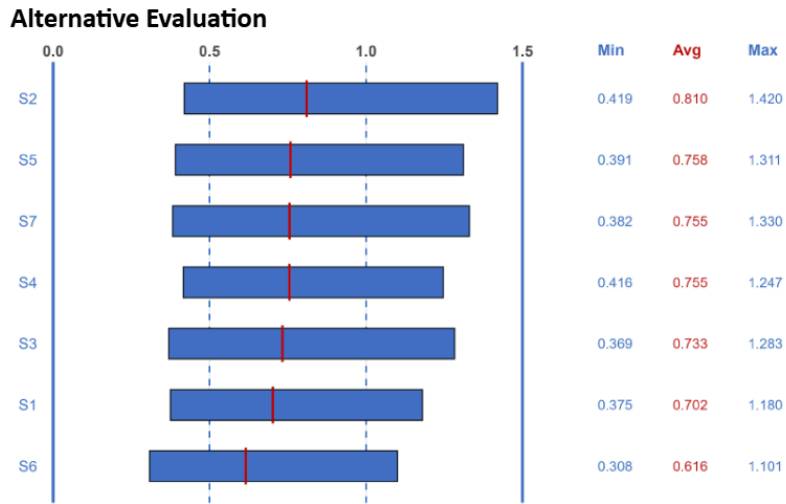


Figura 3.31

Luego para el checkbox de “Stack bar rankings” se utiliza la función `exportMockup37(node_37, alt_sel)`, que muestra el gráfico de la figura 3.32, para mostrar los rankings Stack Bar de cada alternativa para el nodo padre como muestra la figura 3.33.

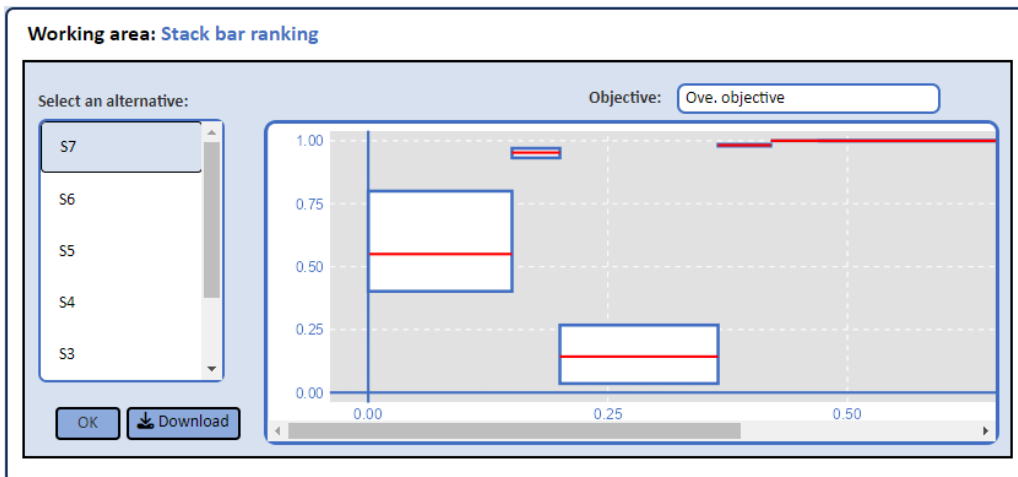
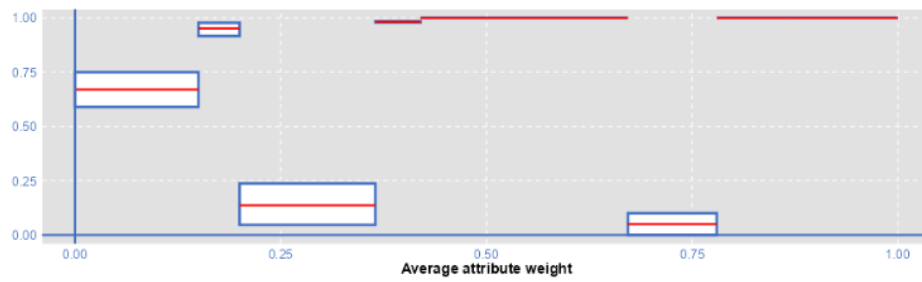


Figura 3.32

Stack Bar Rankings

Stack Bar Ranking - Alternative: S1



Stack Bar Ranking - Alternative: S2

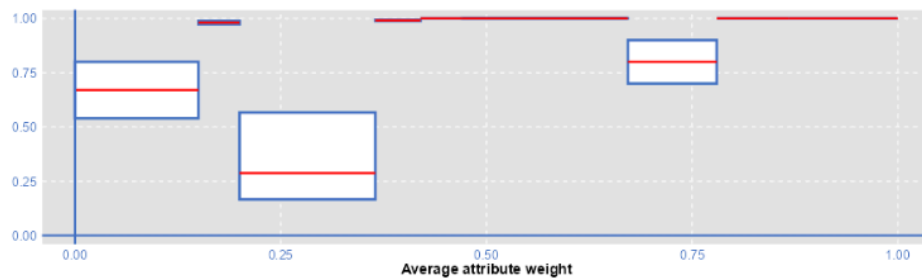


Figura 3.33

Para el checkbox de “Radar charts” utilizaremos la función `exportMockup38()` la cual nos mostrará un `radarchart` de las utilidades de la alternativa del nodo en cuestión utilizando el `mockup 38` como muestra la figura 3.34, únicamente se muestra una lista de `radarcharts` de todas las alternativas del nodo padre como se muestra en la figura 3.35, en este caso solo hay un `radarchart` por página.

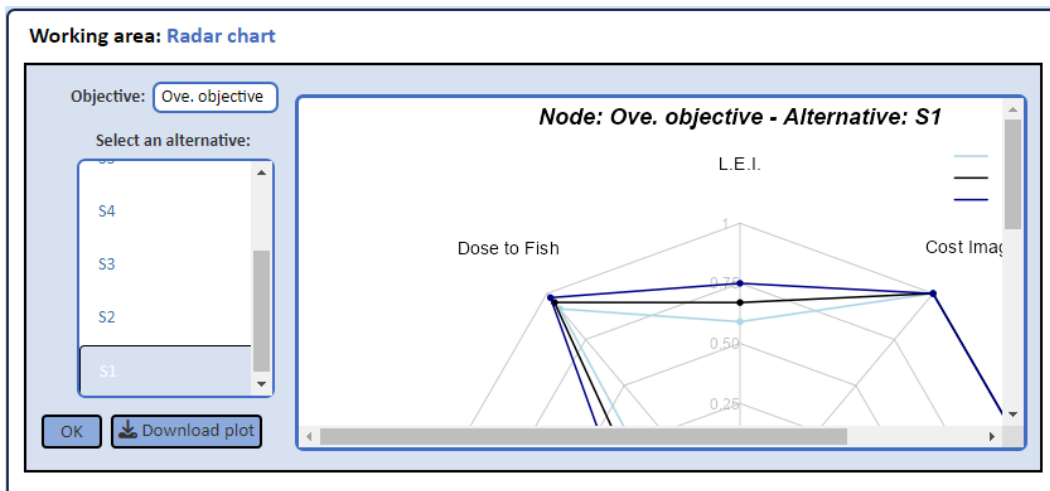


Figura 3.34

Radar Charts

Radar Chart - Alternative: S1

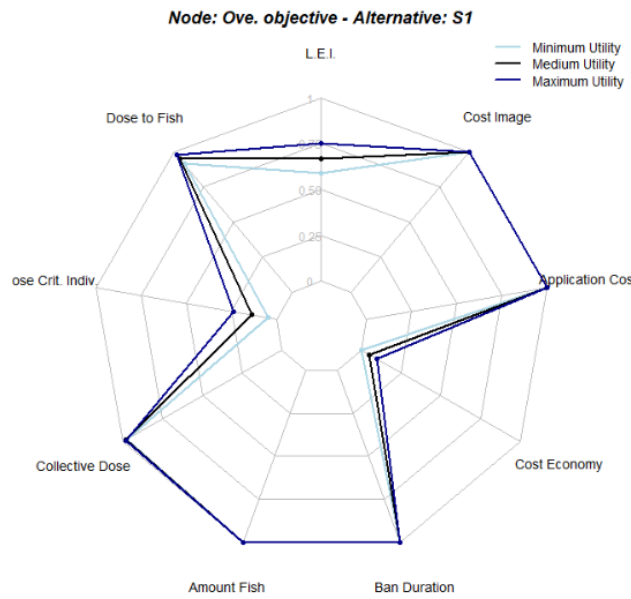


Figura 3.35

Finalmente para el checkbox “Comparative charts” se utiliza la función `exportMockup47(main_alt, alt)` que básicamente corresponde al mockup 47 analizado anteriormente, como se muestra en la figura 3.36, pero únicamente se crean los gráficos de comparación para comparar la primera alternativa con el resto como se muestra en la figura 3.37.

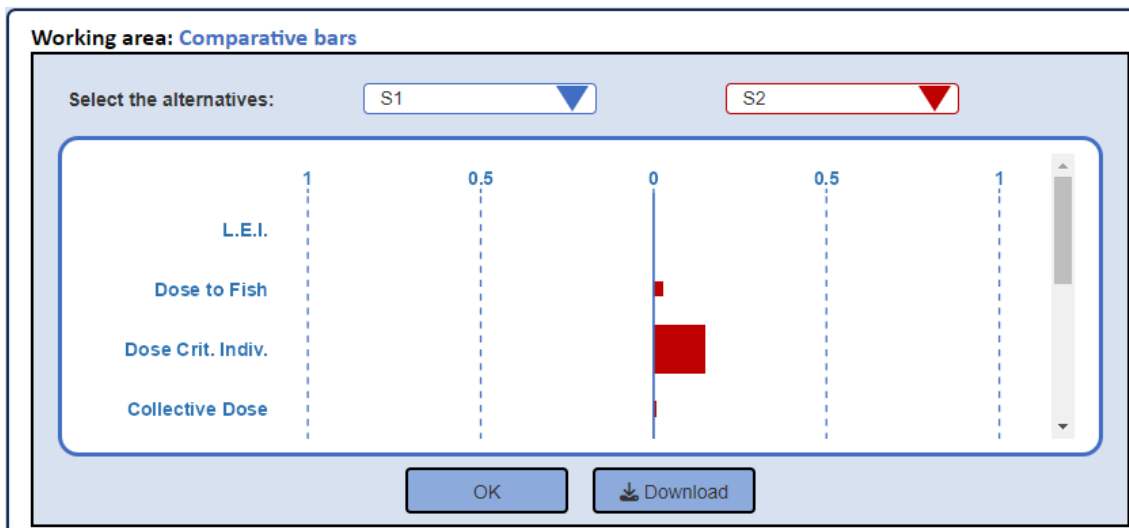
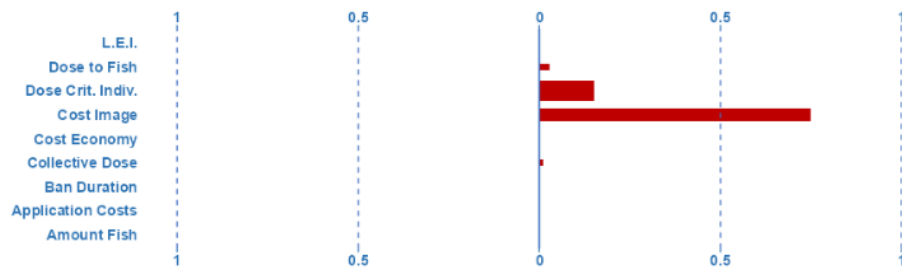


Figura 3.36

Comparative Charts

Comparative Chart - S1 vs S2



Comparative Chart - S1 vs S3

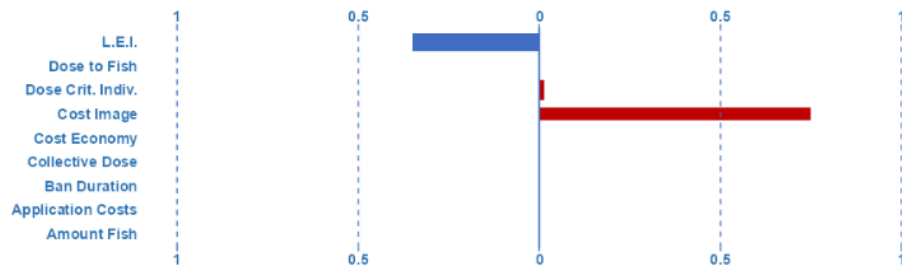


Figura 3.37

4 Pruebas y Validaciones

4.1 Pruebas de interfaz e integración

4.1.1 Mockup de la carga de proyectos

Para el mockup de la carga de proyectos encontramos el estilo inicial definido por el tutor que se indica en la figura 4.1 este estilo consta de apenas diferencias con el estilo final mostrado en la figura 4.2 siendo las más notables que se añade un pequeño dibujo a los botones “Download paper” y que en los radiobuttons no hay nada seleccionado por defecto y el radiobutton de la aplicación finalmente es de color azul como se muestra en la figura 4.2

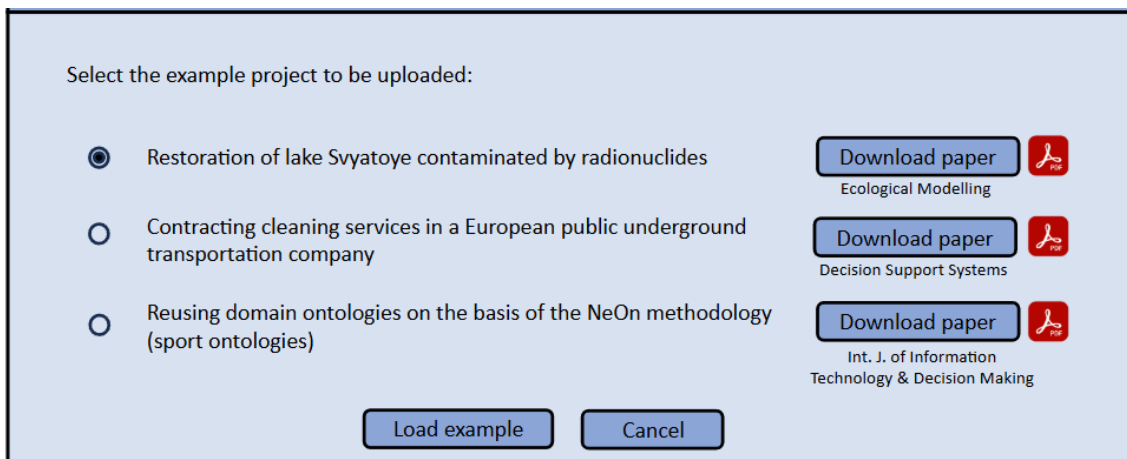


Figura 4.1

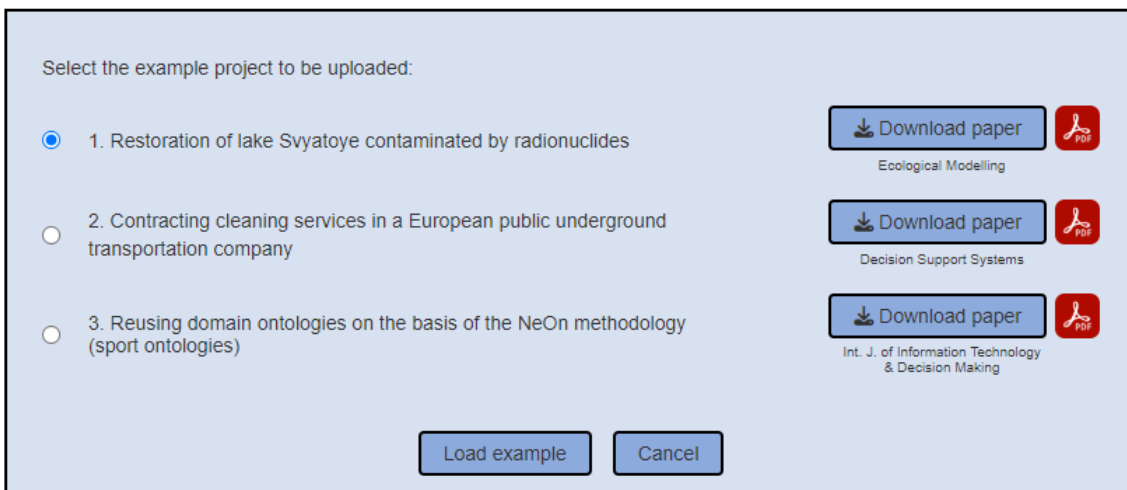


Figura 4.2

Luego el botón “Load Example” no contaba con la plantilla de implementación ya que únicamente tenía que mantener el estilo de los otros botones del desplegable “Project” y como se puede observar en la figura 4.3 el botón mantiene correctamente el formato pedido.

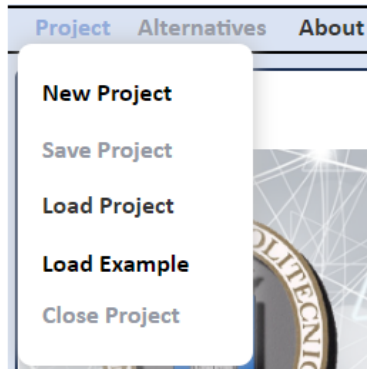


Figura 4.3

4.1.2 Mockup de los diagramas

El diseño de ambos mockups es el mismo cambiando únicamente la gráfica. Para el mockup del diagrama de barras de comparación podemos ver diferencias más notables entre el diseño original de la figura 4.4 y el diseño final 4.5 ya que en el diseño final cambia la orientación de las barras de derecha roja e izquierda azul a derecha azul e izquierda roja, también podemos ver que hay dos líneas discontinuas más y que los ejes se han movido hacia la izquierda en el diseño final, además se ha añadido colores a los desplegables para una mayor calidad, el resto es similar al diseño original.

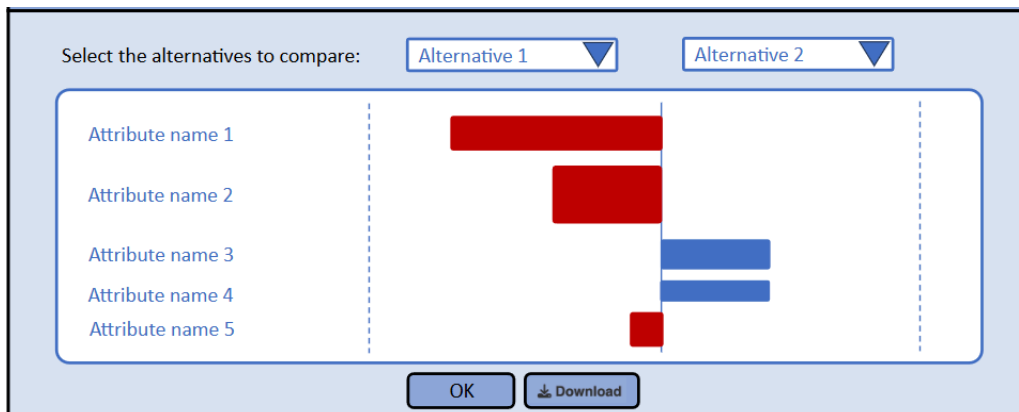


Figura 4.4



Figura 4.5

Para el mockup diagrama de cajas el diseño es bastante más similar solo que cambia que la línea dentro del boxplot es negra en lugar de azul y roja además que otra vez el diagrama se ha desplazado a la izquierda.

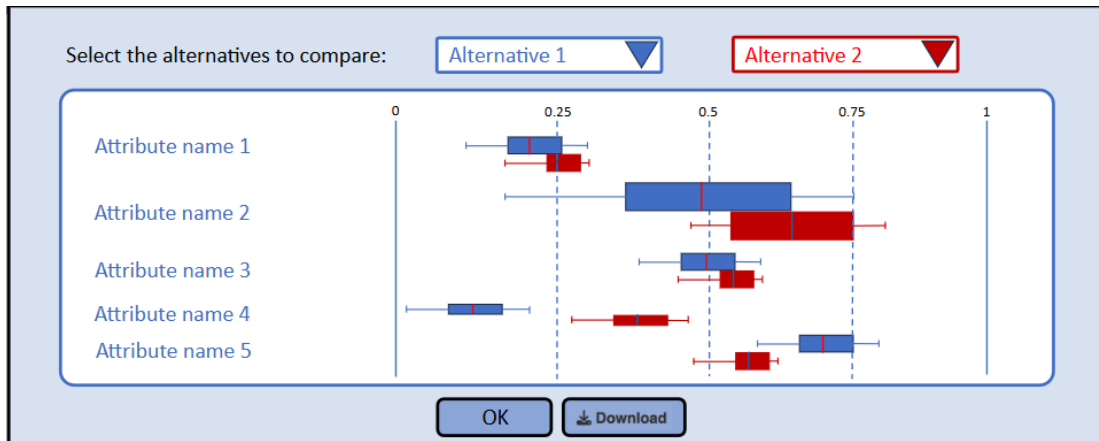


Figura 4.6

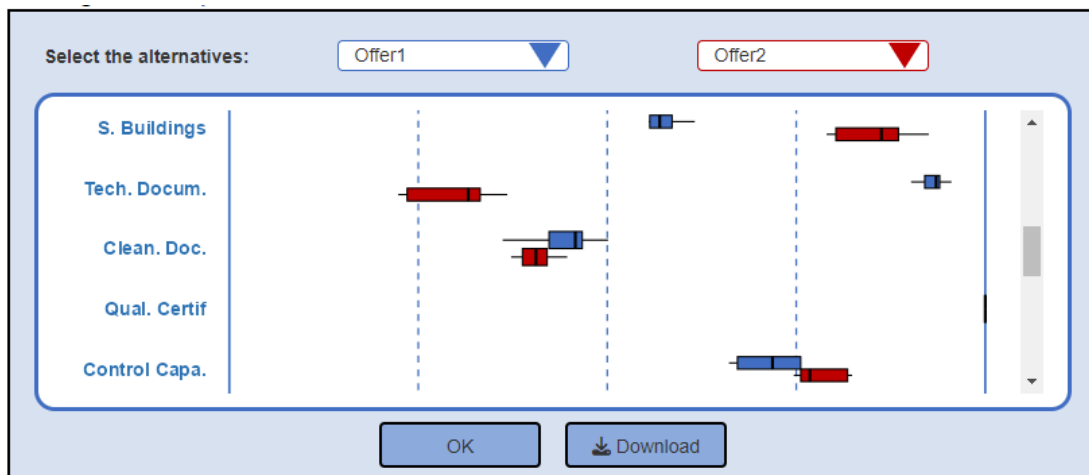
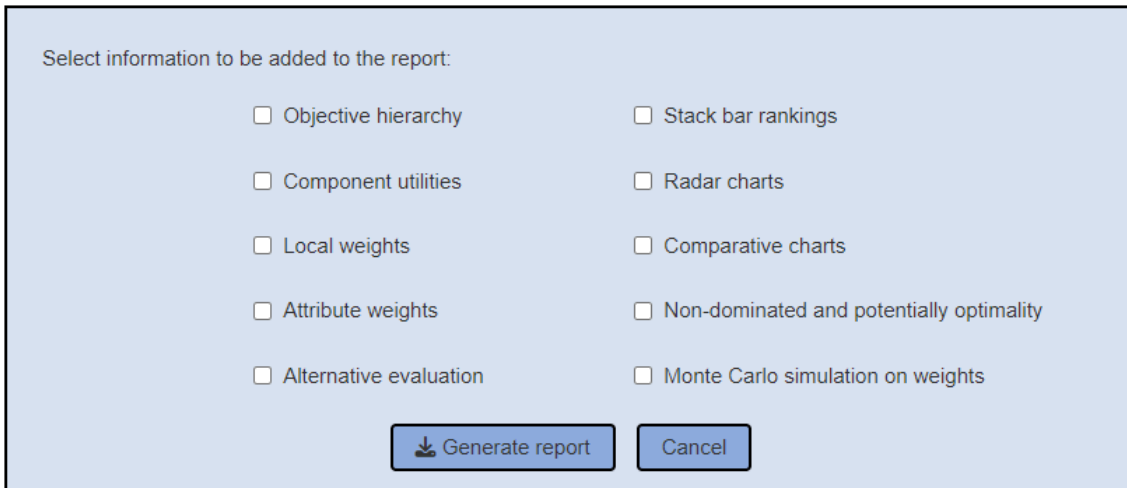


Figura 4.7

4.1.3 Mockup de la generación de informes

Para el mockup de la generación de informes el diseño final e inicial mostrado en la figura 4.8 y figura 4.9 respectivamente son exactamente iguales únicamente cambiando el emoticono que hay antes del botón “Generate Report”

Figura 4.8



Select information to be added to the report:

| | |
|---|---|
| <input type="checkbox"/> Objective hierarchy | <input type="checkbox"/> Stack bar rankings |
| <input type="checkbox"/> Component utilities | <input type="checkbox"/> Radar charts |
| <input type="checkbox"/> Local weights | <input type="checkbox"/> Comparative charts |
| <input type="checkbox"/> Attribute weights | <input type="checkbox"/> Non-dominated and potentially optimality |
| <input type="checkbox"/> Alternative evaluation | <input type="checkbox"/> Monte Carlo simulation on weights |

Figura 4.9

4.2 Validación de requisitos funcionales

4.2.1 Mockup de la carga de proyectos

Para la validación de los requisitos funcionales del mockup de la carga de proyectos se dividió en botones, primero se probó la descarga de los pdfs teniendo la misma estructura todos, una vez se logró el primero se hicieron fácilmente el resto, luego se probó la implementación del botón “Cancel” que será utilizado en otros mockups y por último el botón “Load Example” que para conseguir este requisito funcional primero hubo que analizar el código del botón de “Load Project” para saber en qué parte había que meter el path del fichero de ejemplo y posteriormente hubo que analizar la estructura de los radiobuttons para relacionar el fichero con el radiobutton en cuestión.

4.2.2 Mockup de los diagramas

Para validar los requisitos funcionales de los mockup de los diagramas se ha dividido al igual que en el de la carga de proyectos en funcionalidades y botones. Primero el botón “OK” es común para los dos mockups y básicamente cierra el mockup.

En el mockup del diagrama de barras de comparación encontramos tres funcionalidades la gráfica el botón “Download”, la gráfica y el pulsar la gráfica para verla completa, primero para la gráfica se hizo un log de los dataframes para ver que los pesos y utilidades que se estaban recibiendo eran coherentes y una vez se obtienen los datos correctamente se comenzó a dibujar la gráfica hasta lograr el resultado deseado. Para el botón “Download” y la opción de pulsar la gráfica para verla completa se usa los datos ya calculados en el dataframe y las mismas funciones pero en vez de hacerlo en para el apartado de la gráfica lo hace en formato png para imagen o gráfico para el zoom.

En el mockup del diagrama de cajas también se encuentran las mismas tres funcionalidades del anterior y se comprobaron los requisitos funcionales de la misma manera ya que lo primero que se hizo fue comprobar los datos en un dataframe para ver que se puede dibujar los datos bien, una vez creada la función de dibujar se usa esa función para crear la imagen y la opción del zoom validando así todos los requisitos funcionales.

4.2.3 Mockup de la generación de informes

Para validar los requisitos funcionales del servidor del mockup de la generación de informes primero validamos que el botón “Cancel” cierre correctamente el mockup, una vez visto esto pasamos a comprobar el botón “Generate Report” y sus múltiples para ello crearemos un word con todas las opciones en un proyecto sin simulación y una vez lo tengamos comprobamos que esté cada elemento seleccionado en el word incluyendo. Para lograr imprimir las gráficas hubo que ir a los mockups en cuestión para crear una función de exportación que devolviera un ggplot el cual se pudiera insertar en el word, y en algunos casos hacer iteraciones sobre esos ggplot ya que había que imprimir más de una gráfica.

5 Conclusiones

El objetivo de este trabajo era el desarrollo de tres mockups nuevos para conseguir nuevas funcionalidades en la aplicación WEB-MAUT-DSS, las funcionalidades creadas son:

La carga de ejemplos y descarga de pdfs de ejemplos, esta fue lograda con la integración del submenú “Load Example” del desplegable “Project” que permite la descarga de papers de ejemplo y es capaz de cargar automáticamente la jerarquía desde la aplicación sin tener que ir al explorador de archivos.

La implementación de dos nuevas gráficas dentro de la opción “Ranking” en la parte de “Select chart” seleccionando la opción “Comparative bars”, esto corresponde a los mockups 47 y 48 que están en un único archivo y son dos diagramas comparativos donde se muestra la comparación entre peso y utilidades de todos los atributos comparando entre dos alternativas, dependiendo si hay o no simulación el gráfico será un boxplot o un diagrama de barras más simple respectivamente.

La última implementación desarrollada es la generación de informes en base a opciones, esto crea un word con un formato estándar y generar las gráficas de pesos de atributos, pesos de alternativas, utilidades de los componentes y evaluación de alternativas mediante el ranking, además también muestra la imagen de la jerarquía de objetivos y la descripción y detalles de cada nodo en formato texto.

Como desafíos encontrados en el TFG destacó la curva de aprendizaje de R Shiny ya que este lenguaje hay que comprenderlo bien para ser capaz de desarrollar correctamente pero con la ayuda de compañeros y ambos tutores se hizo más fácil, también ha supuesto un desafío la implementación de la opción de generar informes ya que al principio se pensó en hacerlo en formato pdf pero la escasa existencia de librerías hizo que finalmente se hiciera con la librería office para implementar bastante mas facilmente el informe, además en el informe ha supuesto un reto el hecho de tocar mockups realizados por otros compañeros los cuales no había tocado para exportar las gráficas a mi mockup realizando pequeñas modificaciones en sus códigos, por último la implementación de ambas gráficas en los mockups 47 y 48 ha sido el mayor desafío ya que para obtener los datos tuve que analizar la estructura de datos y funciones auxiliares, pero una vez realizada la primera la realización de la segunda se simplificó significativamente.

Por último me gustaría destacar que durante la realización del TFG he consolidado todos mis conocimientos de R Shiny desde la personalización de la interfaz hasta los procesos de cálculo. Gracias a la complejidad de la generación de informes he podido mejorar mi análisis de código en R además de aprender a realizar un word desde 0 en R. Por otro lado superar errores en el cálculo de las utilidades y los pesos y su implementación gráfica me ha resultado en una experiencia enriquecedora y satisfactoria debido al resultado final obtenido. Pienso firmemente que las líneas implementadas darán continuidad a la aplicación de WEB-MAUT-DSS haciendo de esta una plataforma más sólida para el apoyo a la decisión multicriterio.

6 Análisis de Impacto

El desarrollo de WEB-MAUT-DSS representa una contribución notable a la línea de investigación en sistemas de soporte a la decisión y análisis multiatributo. Concretamente este Trabajo de Fin de Grado se enmarca en la adición de funcionalidades y gráficos que sirve para reforzar una línea de investigación consolidada.

Desde el punto de vista técnico las mejoras realizadas en la aplicación de WEB-MAUT-DSS supone una mejora sustancial en funcionalidad, usabilidad y eficiencia respecto a versiones anteriores, ya que primero añade nuevas gráficas para la comparación de utilidades lo cual es útil para tener más opciones para analizar la decisión, también la funcionalidad de generar informes es realmente útil a la hora de poder tener la información que seleccione el usuario en un documento en formato Word que es fácilmente editable, por último la funcionalidad de cargar proyectos de ejemplo y la descarga de los pdfs de estos proyectos es una funcionalidad muy útil a la hora de ver el funcionamiento de la aplicación sin tener que crear tu propio proyecto o que alguien te de un proyecto.

En cuanto al impacto práctico, la aplicación WEB-MAUT-DSS posee un alto potencial de utilidad en contextos reales diversos, ya que al ser una herramienta generalista de apoyo a la decisión puede adaptarse con facilidad a múltiples dominios simplemente definiendo correctamente los objetivos, criterios y alternativas del problema a analizar.

Como ejemplos de ámbitos en los que puede ser útil está el ámbito empresarial y de gestión de proyectos, donde esta herramienta podría apoyar procesos de decisión complejos, como la selección de proyectos de inversión, la priorización de iniciativas estratégicas o la evaluación de proveedores, donde intervienen múltiples criterios.

También puede ser útil en otros ámbitos como el ámbito medioambiental donde el sistema WEB-MAUT-DSS resulta especialmente valioso para problemas de decisión bajo incertidumbre, como la gestión de recursos naturales, selección de políticas medioambientales o evaluación de impacto de proyectos ecológicos.

Por último el mejor aspecto de WEB-MAUT-DSS es su adaptabilidad a nuevos dominios donde gracias a su diseño genérico el sistema puede apoyar decisiones multicriterio en sectores tan variados como la salud, la ingeniería, la defensa o seguridad entre otros.

Bibliografía

- [1] E. Canals Orts, "Programación y testeo de módulos de análisis y visualización de datos," Trabajo Fin de Grado, Universidad Miguel Hernández de Elche, Elche, Spain, Feb. 2024. [Online]. Available: <https://dspace.umh.es/bitstream/11000/31999/1/TFG-Canals%20Orts%2C%20Eugenio.pdf>
- [2] P. N. Finlay, *Introducing Decision Support Systems*, Oxford, UK and Cambridge, MA, USA: NCC Blackwell; Blackwell Publishers, 1994.
- [3] E. Turban, *Decision Support and Expert Systems: Management Support Systems*, Englewood Cliffs, NJ, USA: Prentice Hall, 1995.
- [4] J. D. C. Little, "Models and Managers: The Concept of a Decision Calculus," *Management Science*, vol. 16, no. 8, Apr. 1970.
- [5] P. G. W. Keen, *Decision Support Systems: An Organizational Perspective*, Reading, MA, USA: Addison-Wesley, 1978.
- [6] R. H. Sprague and E. D. Carlson, *Building Effective Decision Support Systems*, Englewood Cliffs, NJ, USA: Prentice-Hall, 1982.
- [7] S. L. Alter, *Decision Support Systems: Current Practice and Continuing Challenges*, Reading, MA, USA: Addison-Wesley, 1980.
- [8] A. Castro Vásquez, G. Bravo Martínez, F. Enríquez Aguilera, and M. V. Linares Gil, "Evolución de los sistemas de soporte a la toma de decisiones," *CULCyT*, Año 14, No. 63, pp. 49–59, Sep.–Dic. 2017.
- [9] D. J. Power, *Decision Support Systems: Concepts and Resources for Managers*, Westport, CT, USA: Quorum Books, 2002.
- [10] P. Haettenschwiler, "Neues anwenderfreundliches Konzept der Entscheidungsunterstützung," in *Gutes Entscheiden in Wirtschaft, Politik und Gesellschaft*, Zurich, Switzerland: vdf Hochschulverlag AG, pp. 189–208, 1999.
- [11] Semantic System, "DSS tipos sistema de apoyo a la toma de decisiones para ahorrar costes," [Online]. Available: <https://www.semantic-systems.com/investigacion/temas/dss-tipos-sistema-a-poyo-toma-decisiones-ahorrar-costes.html>
- [12] R. L. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. Cambridge, UK: Cambridge University Press, 1993.
- [13] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*. Princeton, NJ: Princeton University Press, 1944.
- [14] R. L. Keeney y H. Raiffa, "Foundations and Principles of Decision Analysis," en **Decision Analysis**, edited by H. raiffa, Addison-Wesley, 1976, pp. 11–32.
- [15] A. Jiménez Martín, "Trabajo de Investigación," Concurso de acceso a plazas de cuerpos docentes universitarios, Dept. de Inteligencia Artificial,

E.T.S.I. Informáticos, Universidad Politécnica de Madrid, Madrid, Spain, Res. 25 Mar. 2021.

[16] J. P. Hammond, R. L. Keeney, y H. Raiffa, *Smart Choices: A Practical Guide to Making Better Decisions*, Harvard Business School Press, 1999.

[17] R. L. Keeney, *Value-Focused Thinking: A Path to Creative Decisionmaking*, Harvard University Press, 1996.

[18] S. Belton y T. J. Stewart, *Multiple Criteria Decision Analysis: An Integrated Approach*, Kluwer Academic Publishers, 2002.

[19] P. Goodwin y G. Wright, *Decision Analysis for Management Judgment*, 5th ed., Wiley, 2014.

[20] E. Triantaphyllou, *Multi-Criteria Decision Making Methods: A Comparative Study*, Springer, 2000.

[21] A. Jiménez Martín, “Un Sistema de Ayuda a la Decisión Multiatributo con Asignaciones Imprecisas,” Ph.D. dissertation, Dept. of Artificial Intelligence, Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain, 2002.

[22] J. Lahdelma, P. Salminen, y J. Hokkanen, “SMAA – Stochastic multicriteria acceptability analysis,” *European Journal of Operational Research*, vol. 106, no. 1, pp. 137–143, 1998.

[23] E. Stillwell, D. Seaver, y W. Edwards, “A comparison of weight approximation techniques in multiattribute utility decision making,” *Organizational Behavior and Human Performance*, vol. 28, no. 1, pp. 62–77, 1981.

[24] S. Ríos Insua, A. Jiménez, and A. Mateos, “Los sistemas de ayuda a la decisión,” *Anales de la Real Academia de Doctores de España*, vol. 9, pp. 161–177, 2005. [Online]. Available: <https://www.rade.es/doc/1V9N2-sixto%20rios-ayudadecision.pdf>

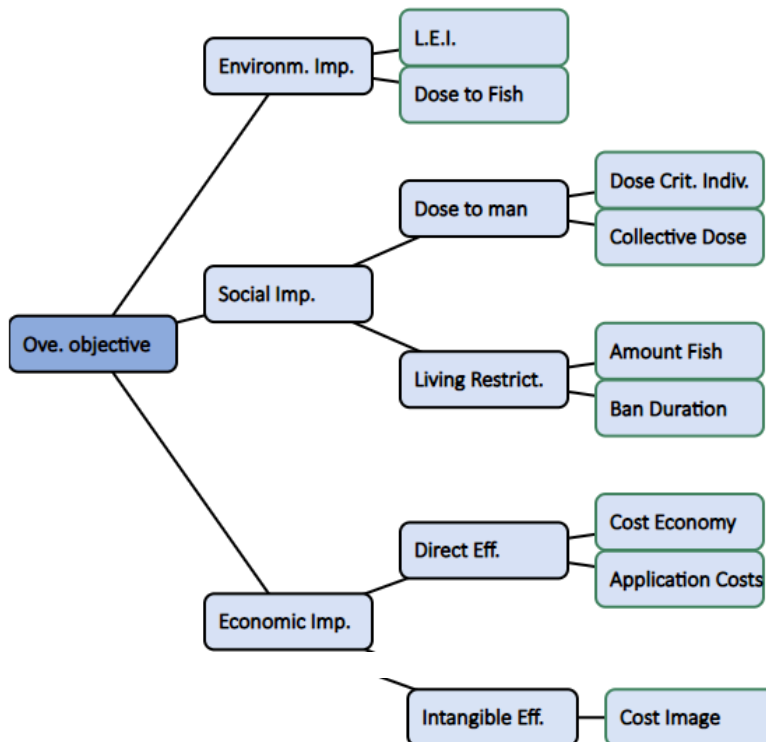
[25] R. Ihaka, “R: Past and Future History,” A draft paper for Interface '98, Statistics Department, University of Auckland, Auckland, New Zealand, 1998. [Online]. Available: https://cran.r-project.org/doc/html/interface98-paper/paper_2.html

7 Anexo

Resultado de generar un informe con el botón “Generate Report” del mockup de generación de informes usando el primer proyecto de “Load Example” del mockup de la carga de proyectos de ejemplo

WEB-MAUT-DSS Project Report

Objectives Hierarchy and Attributes



Node label: Ove. objective

Node name: Overall objective

Node description:

Direct descendants: Environm. Imp., Social Imp., Economic Imp.

Node label: Environm. Imp.

Node name: Environmental impact

Node description:

Direct descendants: L.E.I., Dose to Fish

Node label: L.E.I.

Node name: Lake ecosystem index

Node description:

Attribute type: Continuous

Attribute units: LEI

Minimum value: 1

Maximum value: 5

Most preferred value: 1

Node label: Dose to Fish

Node name: Dose to Fish
Node description:
Attribute type: Continuous
Attribute units: mGy
Minimum value: 0
Maximum value: 1e+04
Most preferred value: 0

Node label: Social Imp.
Node name: Social impact
Node description:
Direct descendants: Dose to man, Living Restrict.

Node label: Dose to man
Node name: Dose to man
Node description:
Direct descendants: Dose Crit. Individ., Collective Dose

Node label: Dose Crit. Individ.
Node name: Dose to critical individuals
Node description:
Attribute type: Continuous
Attribute units: mSv
Minimum value: 0
Maximum value: 500
Most preferred value: 0

Node label: Collective Dose
Node name: Collective Dose
Node description:
Attribute type: Continuous
Attribute units: mSv per person
Minimum value: 0
Maximum value: 1.2e+05
Most preferred value: 0

Node label: Living Restrict.
Node name: Living Restrictions
Node description:
Direct descendants: Amount Fish, Ban Duration

Node label: Amount Fish
Node name: Amount of Fish
Node description:
Attribute type: Continuous
Attribute units: Tonnes
Minimum value: 0
Maximum value: 100
Most preferred value: 0

Node label: Ban Duration

Node name: Ban Duration
Node description:
Attribute type: Continuous
Attribute units: Months
Minimum value: 0
Maximum value: 150
Most preferred value: 0

Node label: Economic Imp.
Node name: Economic Impact
Node description:
Direct descendants: Direct Eff., Intangible Eff.

Node label: Direct Eff.
Node name: Direct Effects
Node description:
Direct descendants: Cost Economy, Application Costs

Node label: Cost Economy
Node name: Cost to Economy
Node description:
Attribute type: Continuous
Attribute units: Euros
Minimum value: 0
Maximum value: 1e+08
Most preferred value: 0

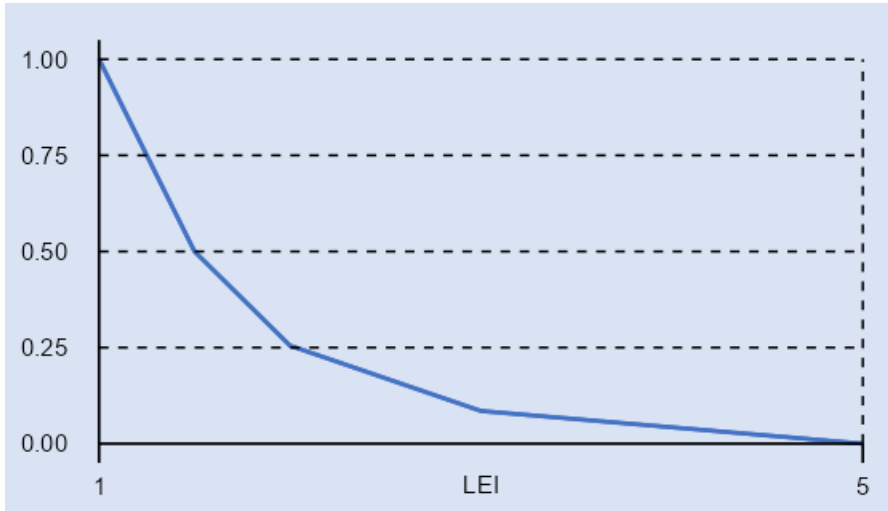
Node label: Application Costs
Node name: Application Costs
Node description:
Attribute type: Continuous
Attribute units: Euros
Minimum value: 0
Maximum value: 1e+07
Most preferred value: 0

Node label: Intangible Eff.
Node name: Intangible effects
Node description:
Direct descendants: Cost Image

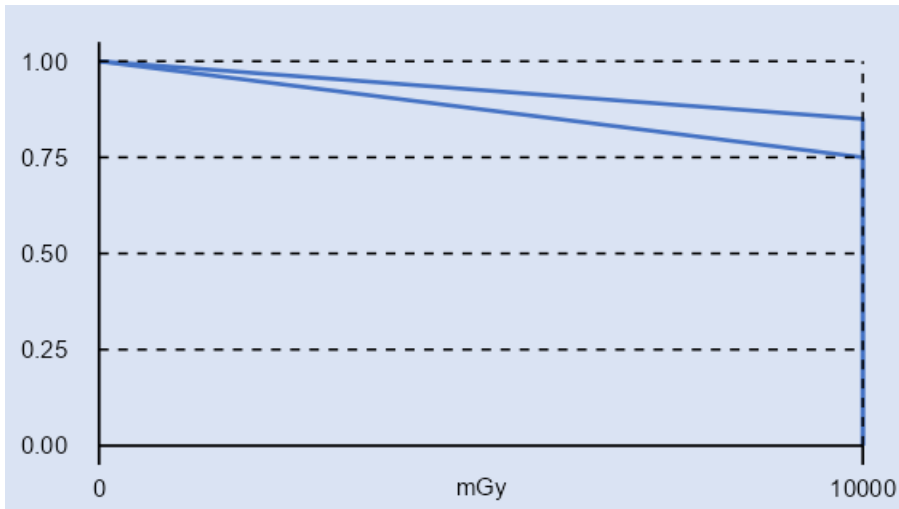
Node label: Cost Image
Node name: Cost to image
Node description:
Attribute type: Continuous
Attribute units: Subjective scale
Minimum value: 0
Maximum value: 1
Most preferred value: 1

Component Utilities

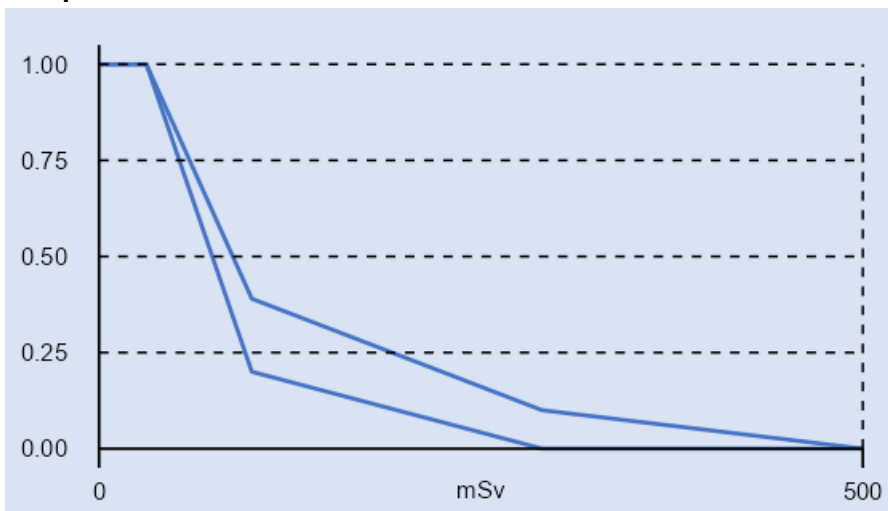
Component Utilities – Node: L.E.I.



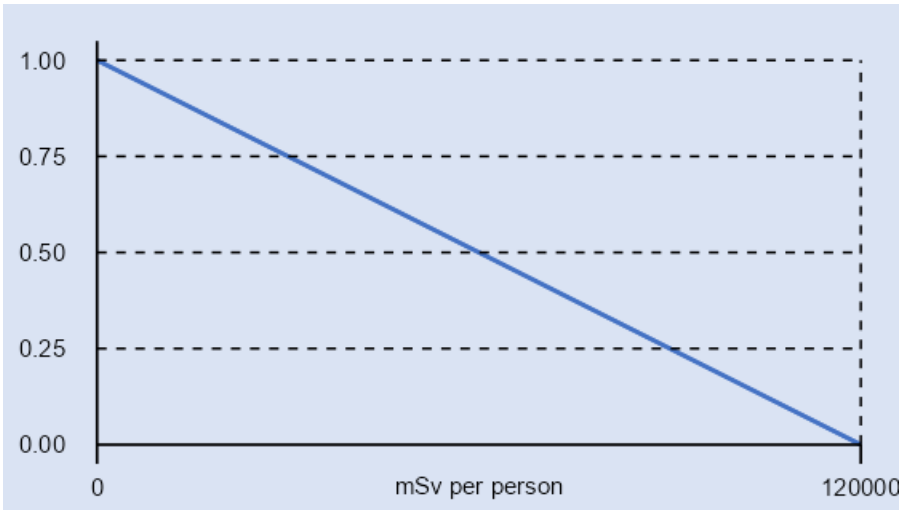
Component Utilities – Node: Dose to Fish



Component Utilities – Node: Dose Crit. Individ.



Component Utilities – Node: Collective Dose



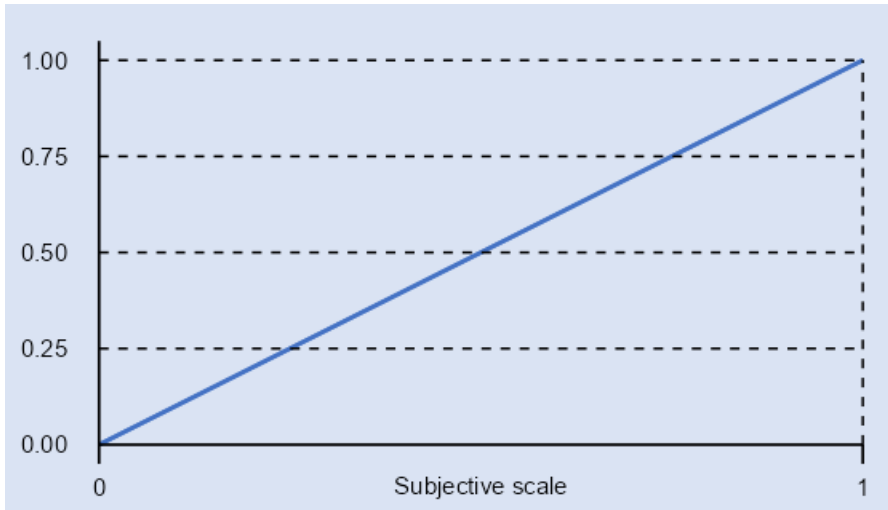
Component Utilities – Node: Cost Economy



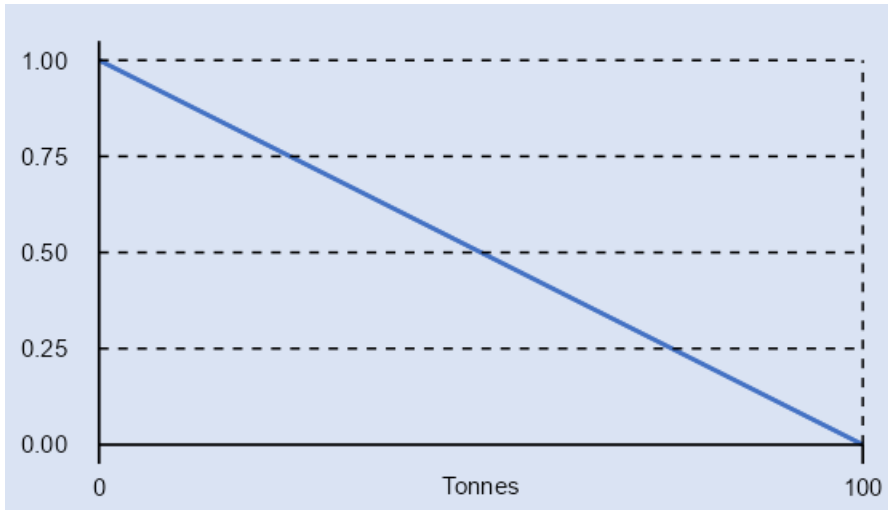
Component Utilities – Node: Application Costs



Component Utilities – Node: Cost Image



Component Utilities – Node: Amount Fish

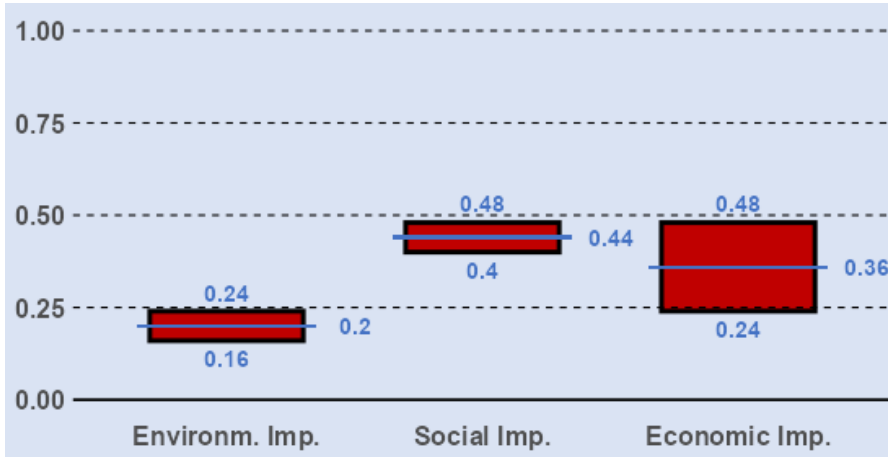


Component Utilities – Node: Ban Duration

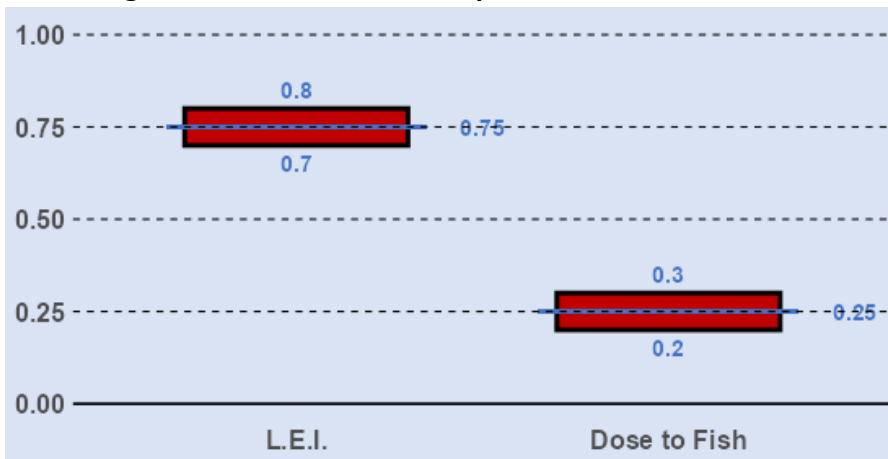


Local Weights

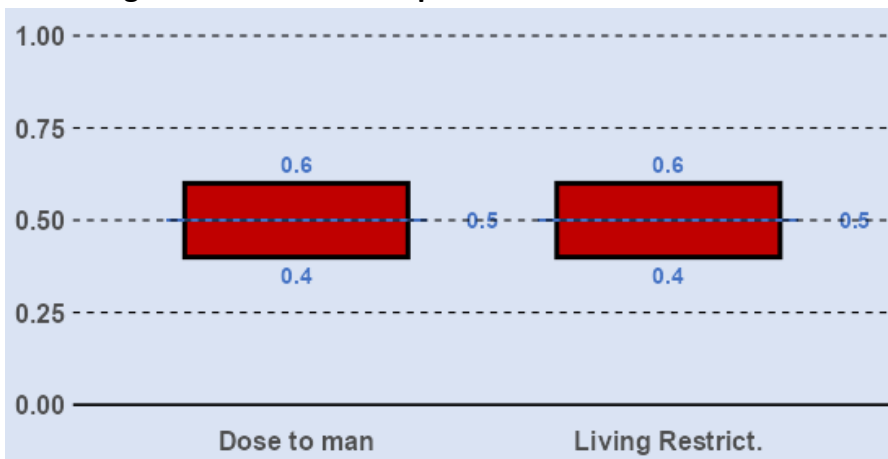
Local weights – Node: Ove. objective



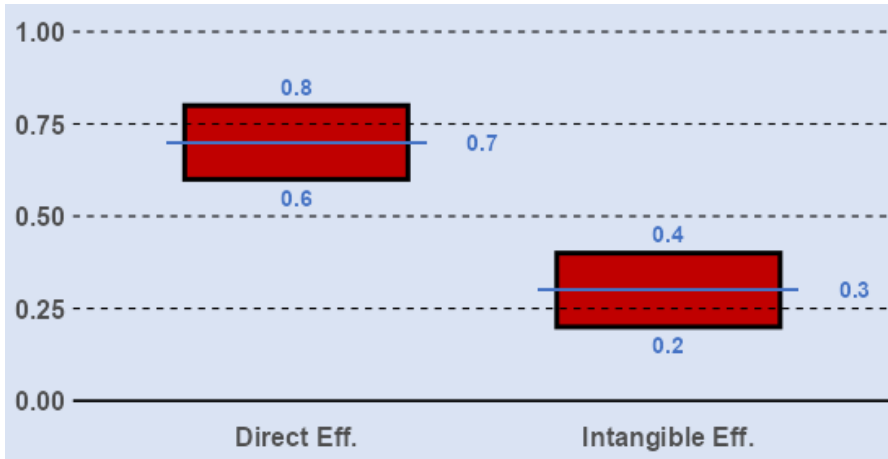
Local weights – Node: Environm. Imp.



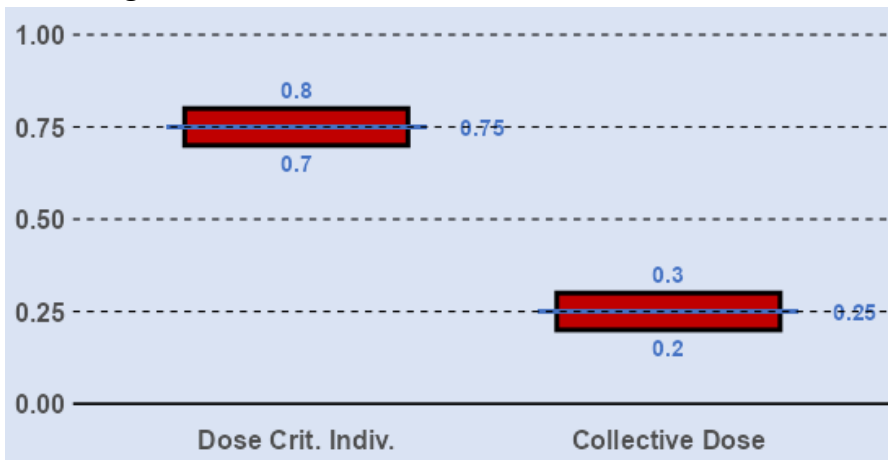
Local weights – Node: Social Imp.



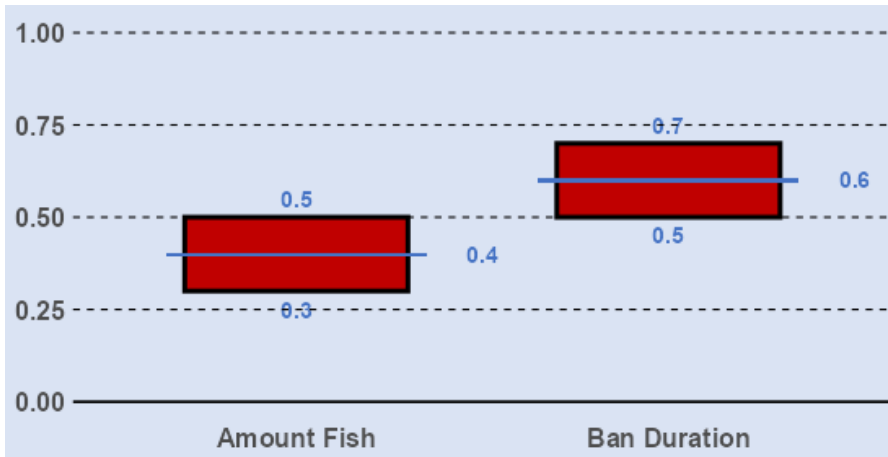
Local weights – Node: Economic Imp.



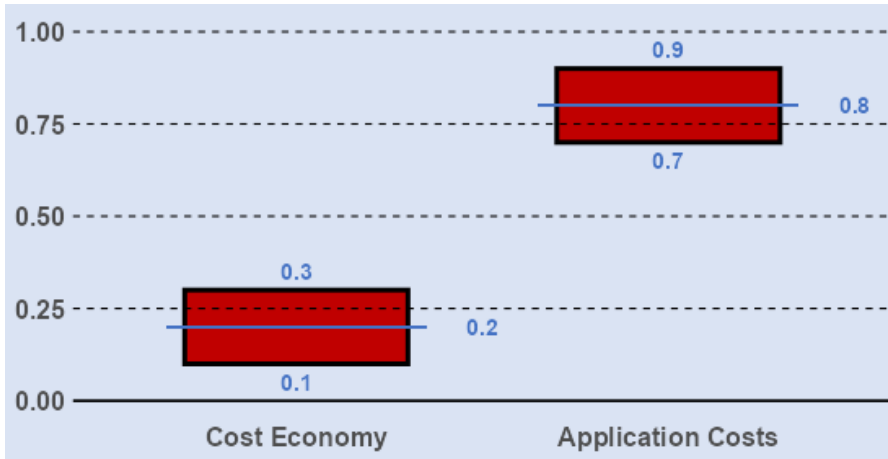
Local weights – Node: Dose to man



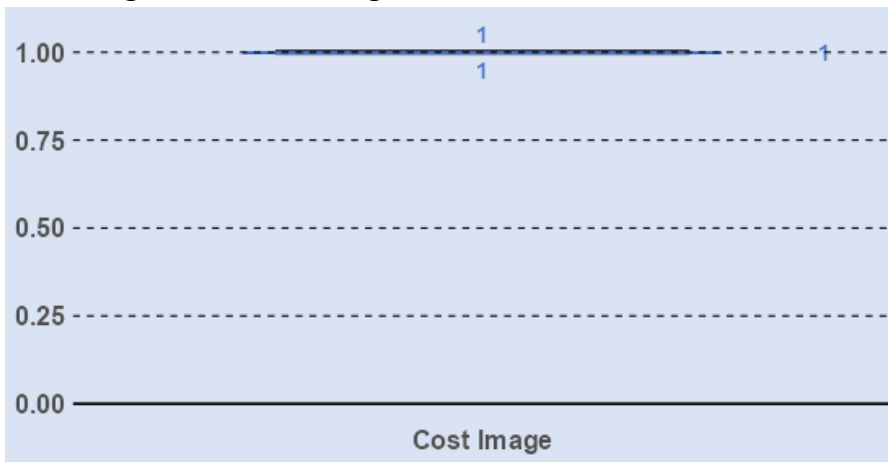
Local weights – Node: Living Restrict.



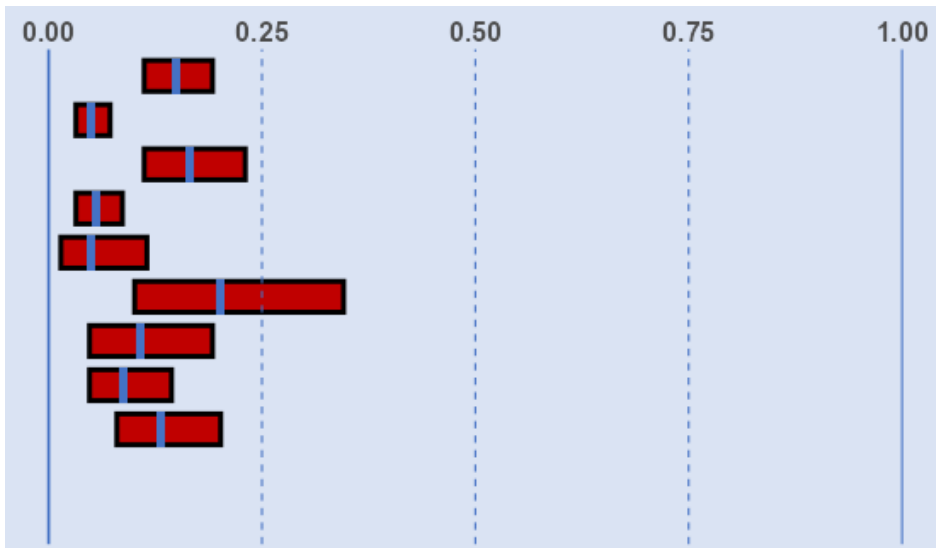
Local weights – Node: Direct Eff.



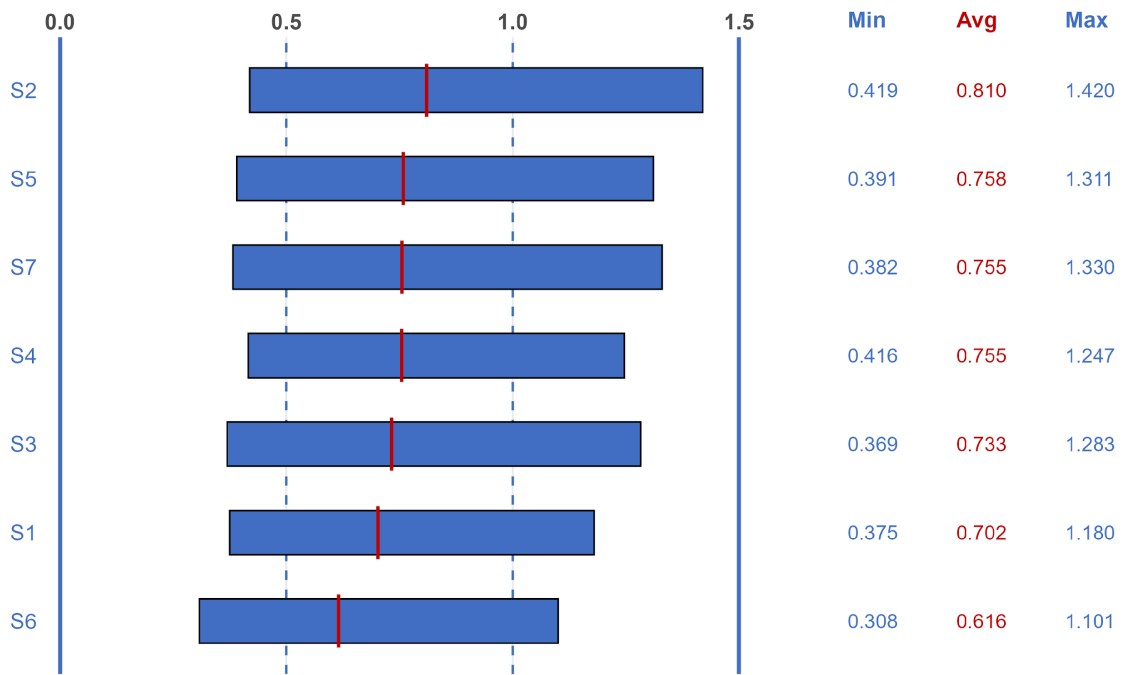
Local weights – Node: Intangible Eff.



Attribute Weights

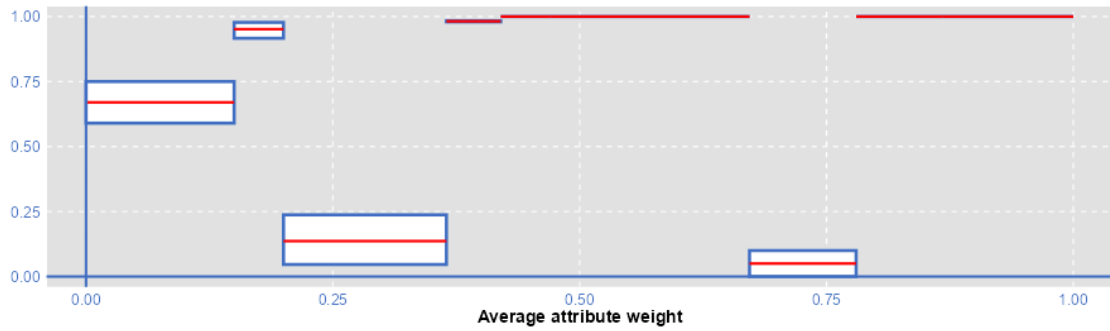


Alternative Evaluation

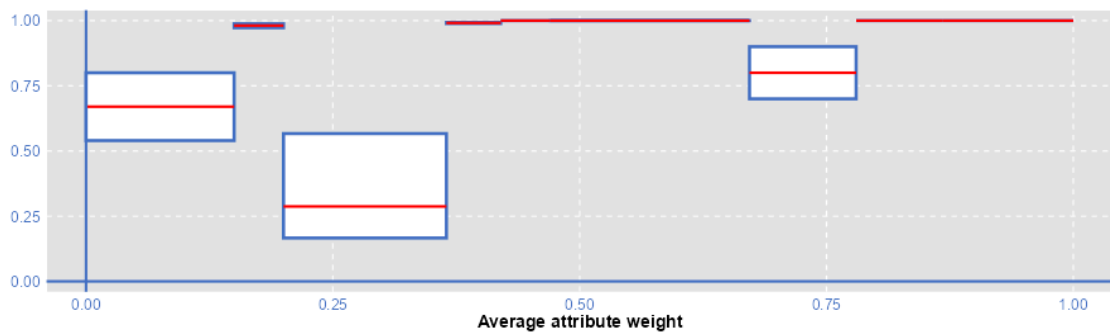


Stack Bar Rankings

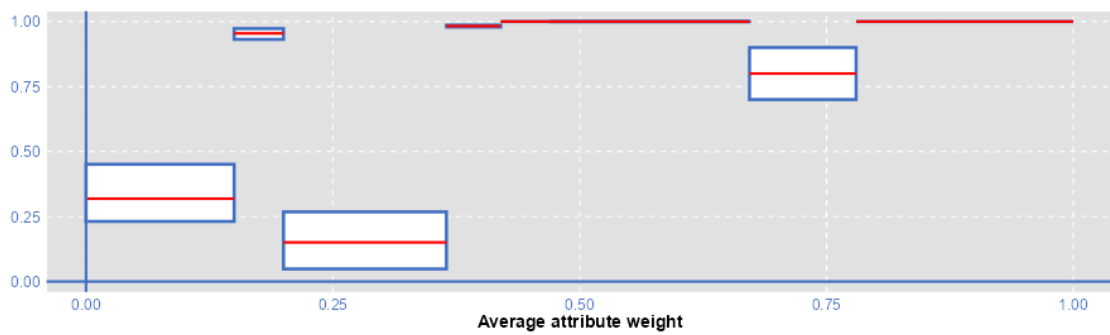
Stack Bar Ranking - Alternative: S1



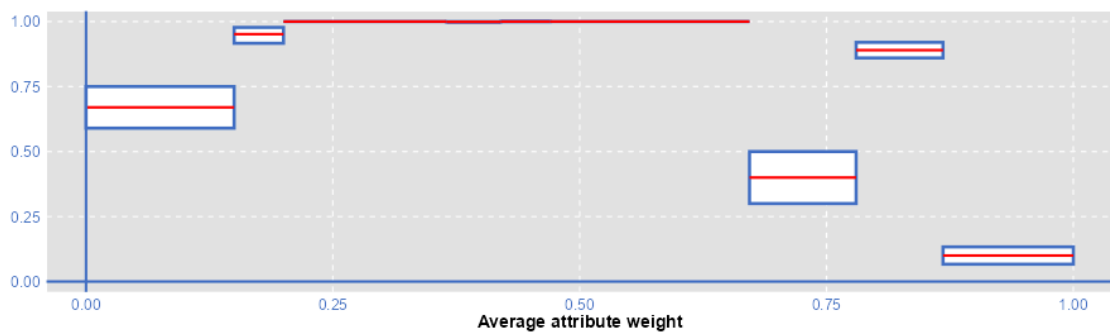
Stack Bar Ranking - Alternative: S2



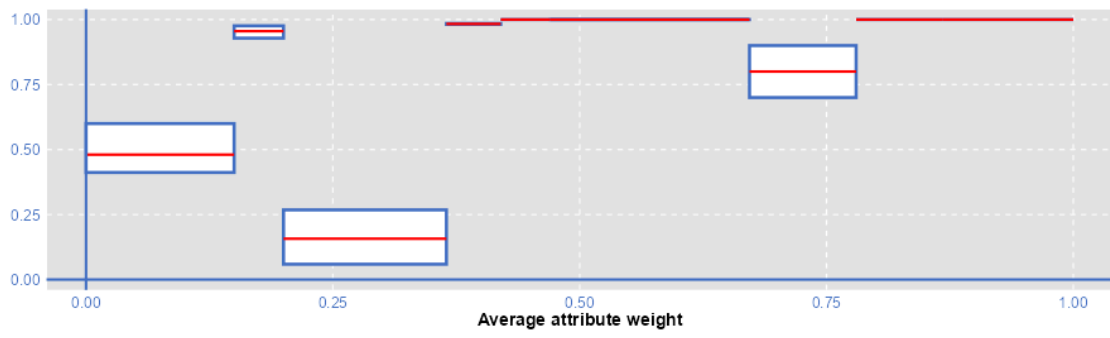
Stack Bar Ranking - Alternative: S3



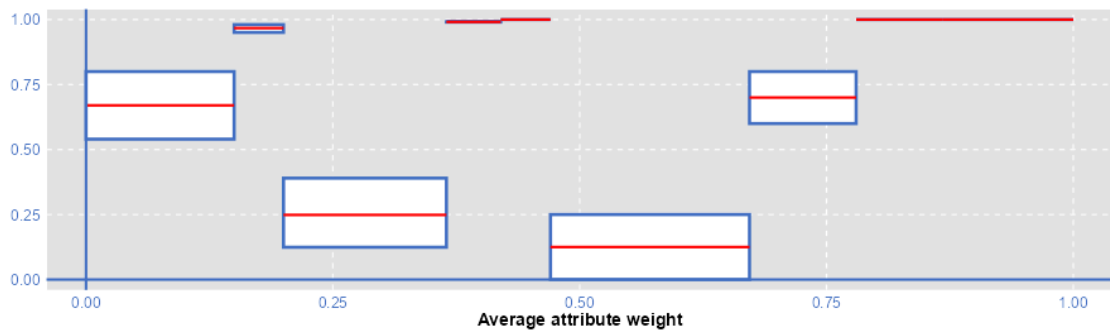
Stack Bar Ranking - Alternative: S4



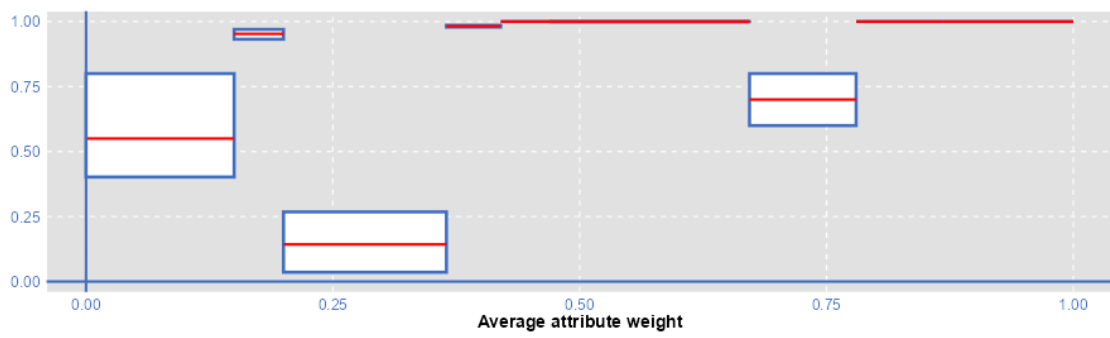
Stack Bar Ranking - Alternative: S5



Stack Bar Ranking - Alternative: S6



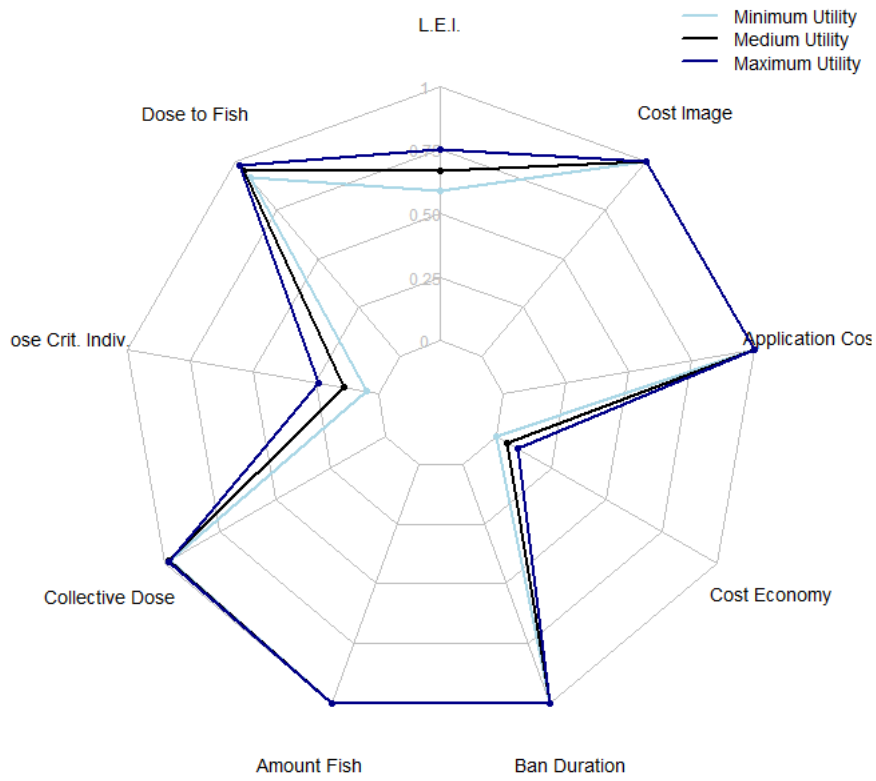
Stack Bar Ranking - Alternative: S7



Radar Charts

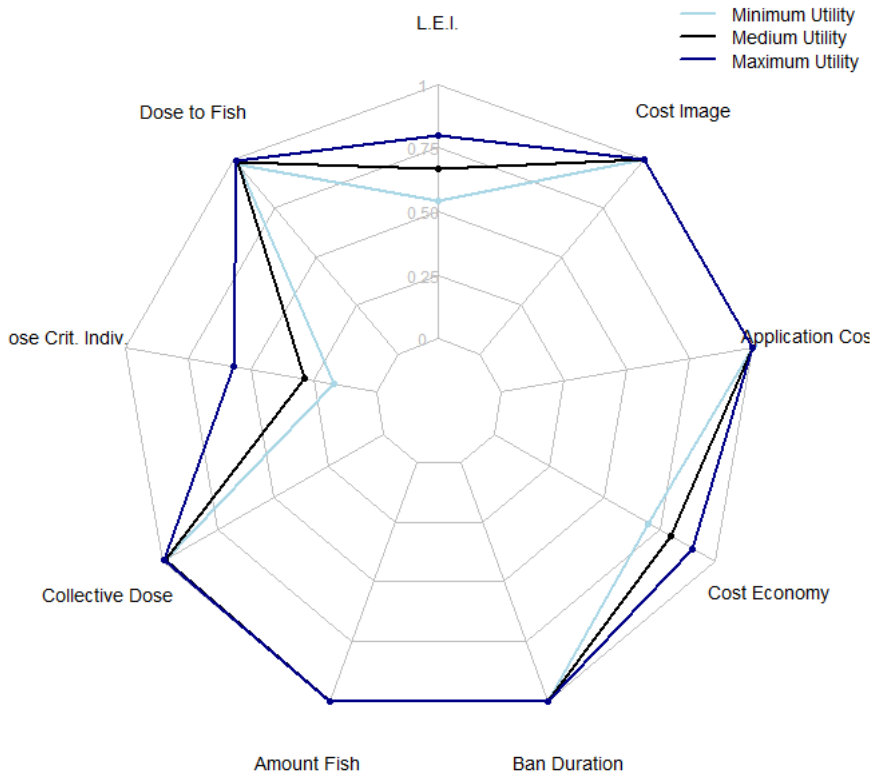
Radar Chart - Alternative: S1

Node: Ove. objective - Alternative: S1



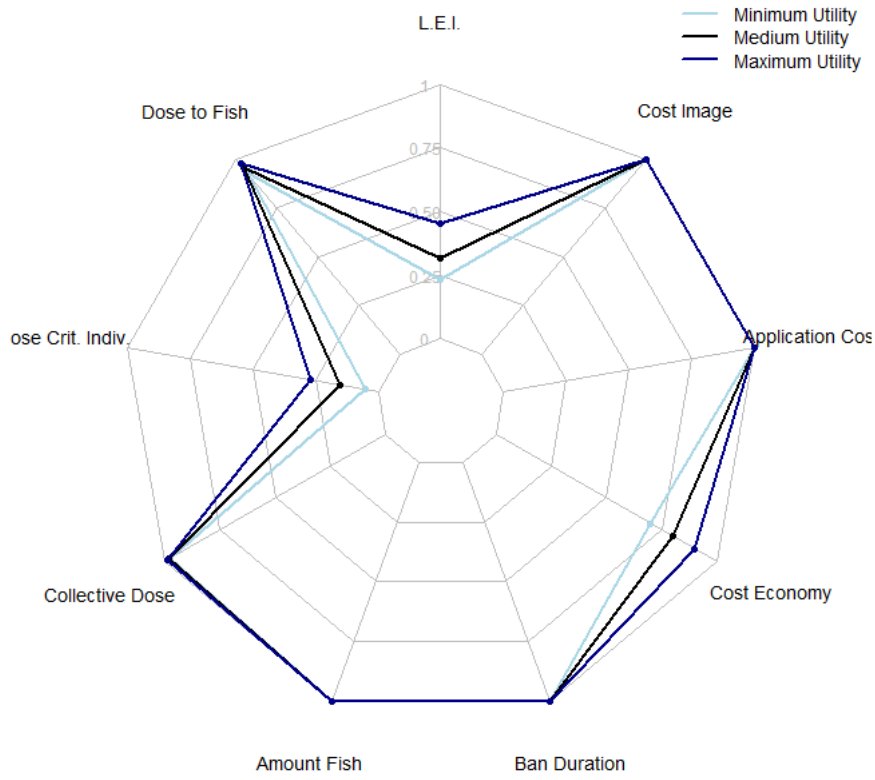
Radar Chart - Alternative: S2

Node: Ove. objective - Alternative: S2



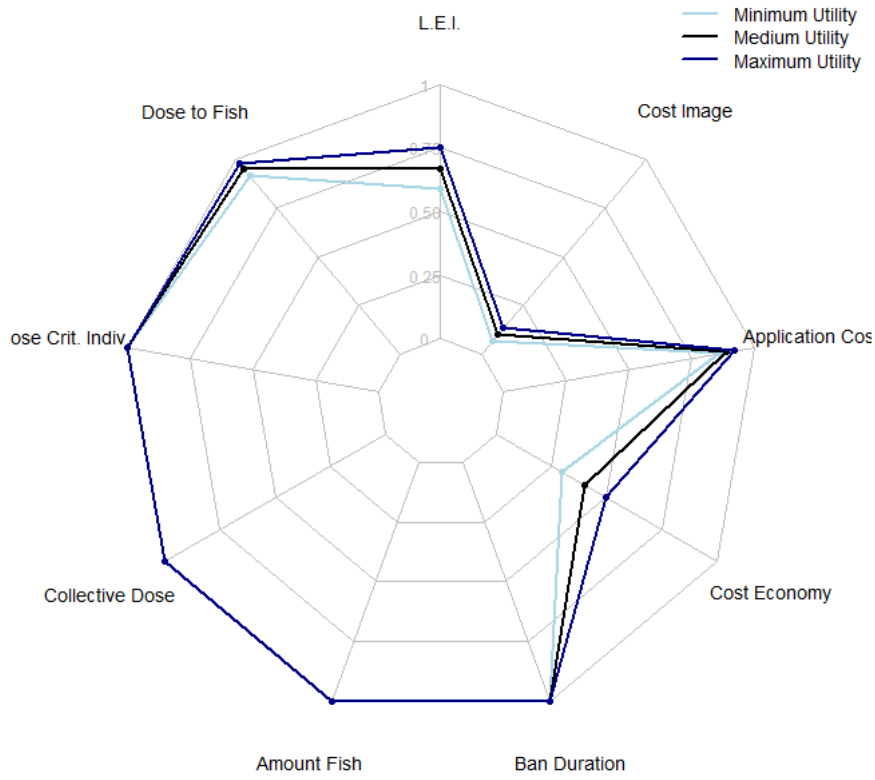
Radar Chart - Alternative: S3

Node: Ove. objective - Alternative: S3



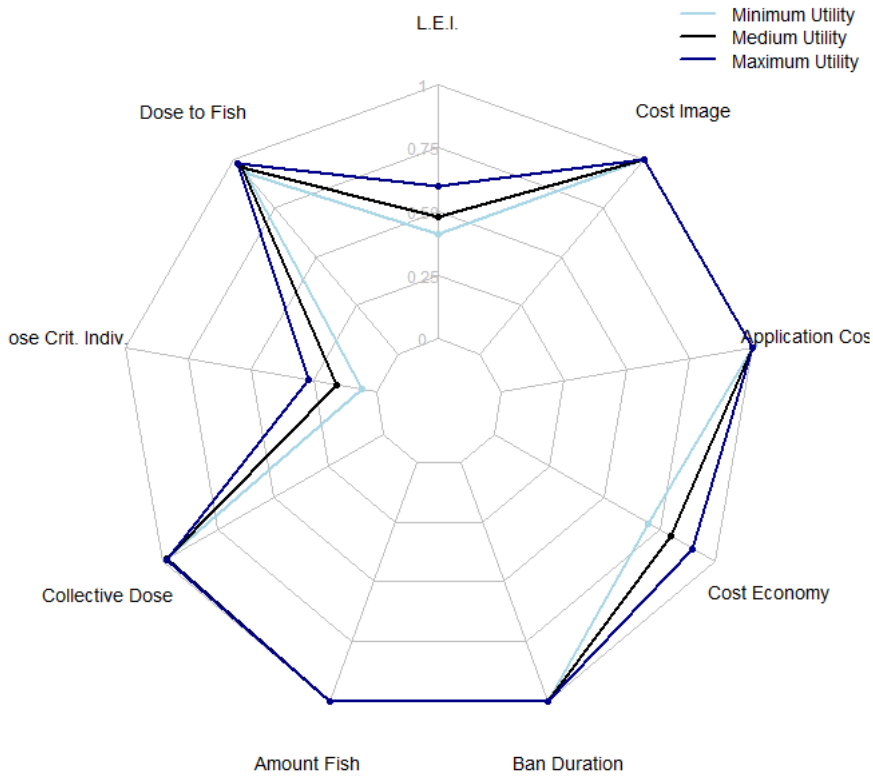
Radar Chart - Alternative: S4

Node: Ove. objective - Alternative: S4



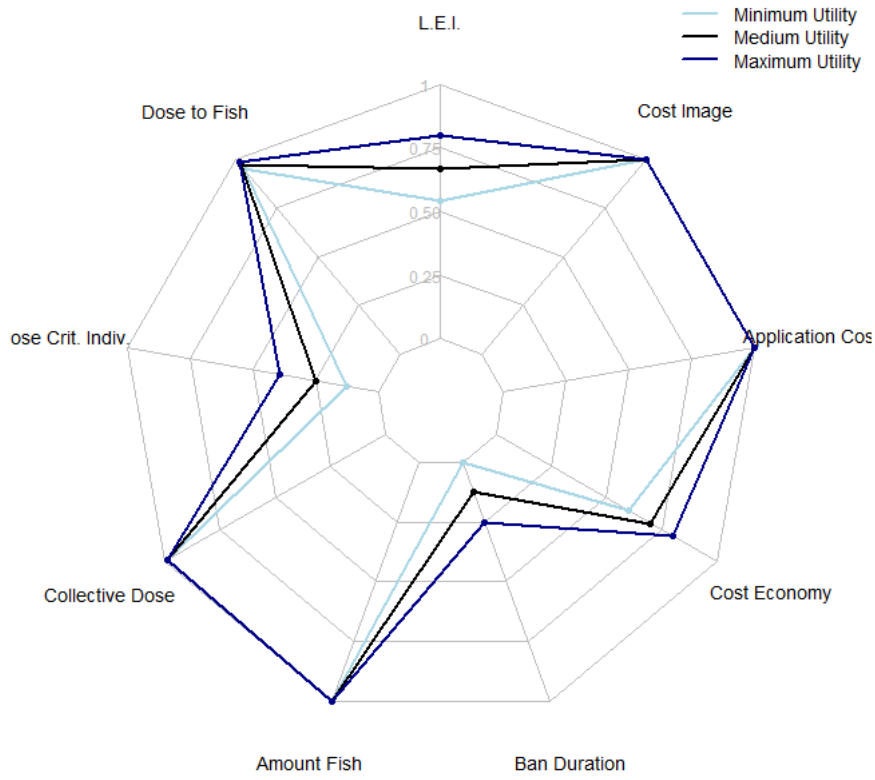
Radar Chart - Alternative: S5

Node: Ove. objective - Alternative: S5



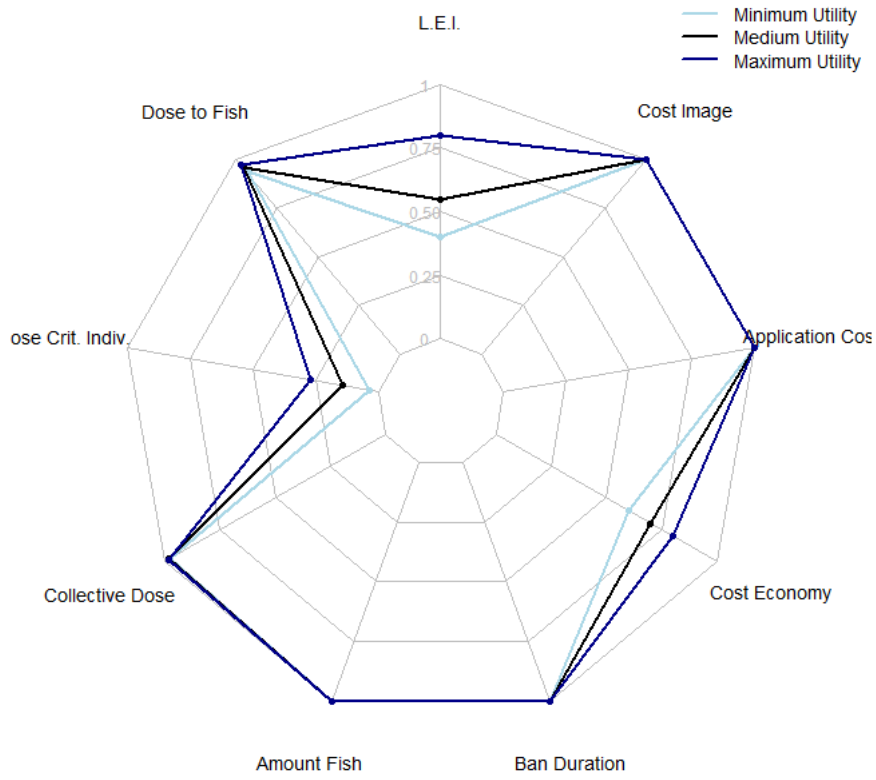
Radar Chart - Alternative: S6

Node: Ove. objective - Alternative: S6



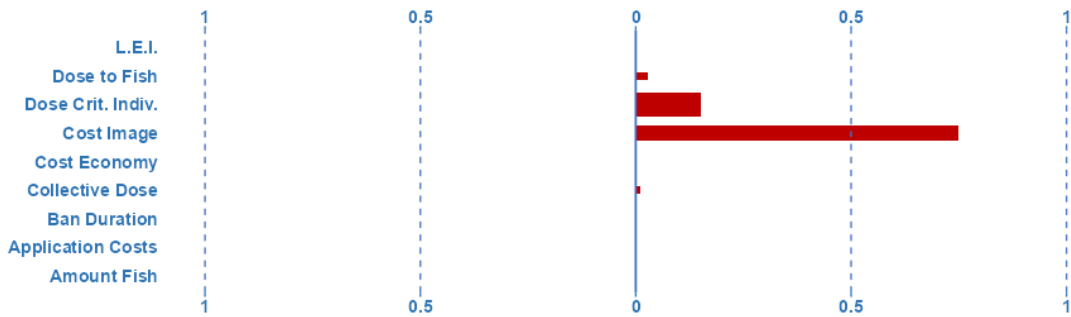
Radar Chart - Alternative: S7

Node: Ove. objective - Alternative: S7

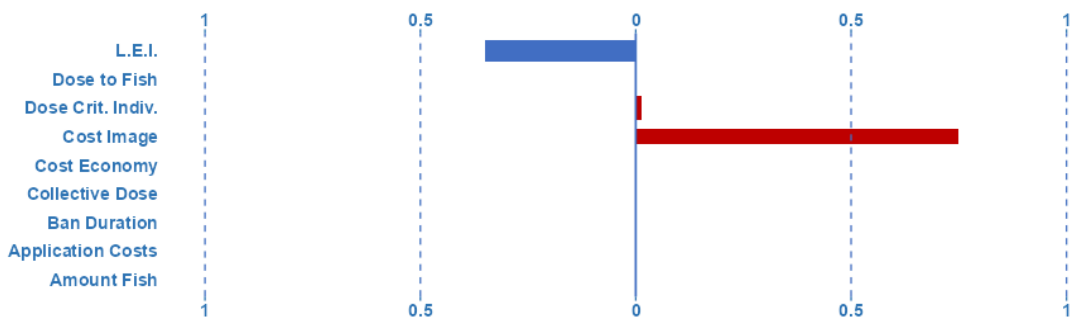


Comparative Charts

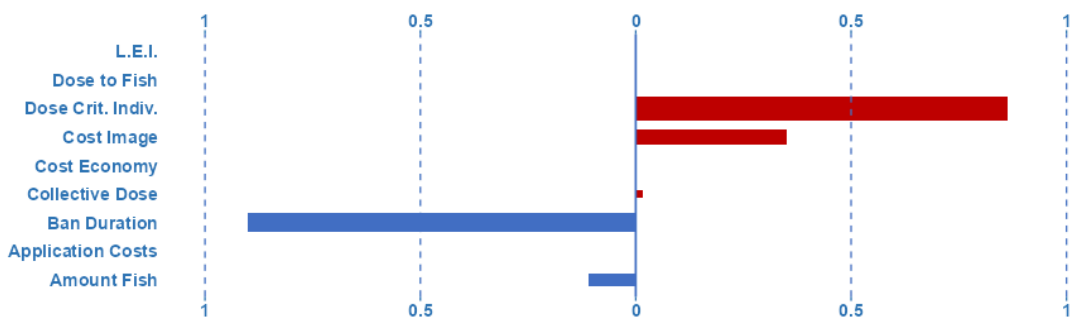
Comparative Chart - S1 vs S2



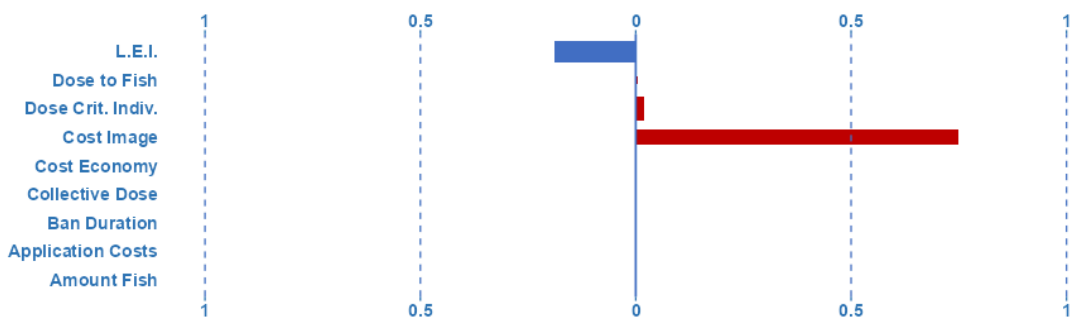
Comparative Chart - S1 vs S3



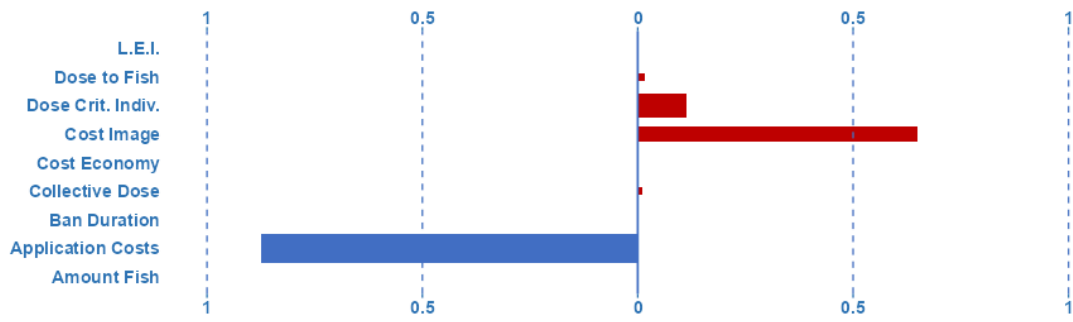
Comparative Chart - S1 vs S4



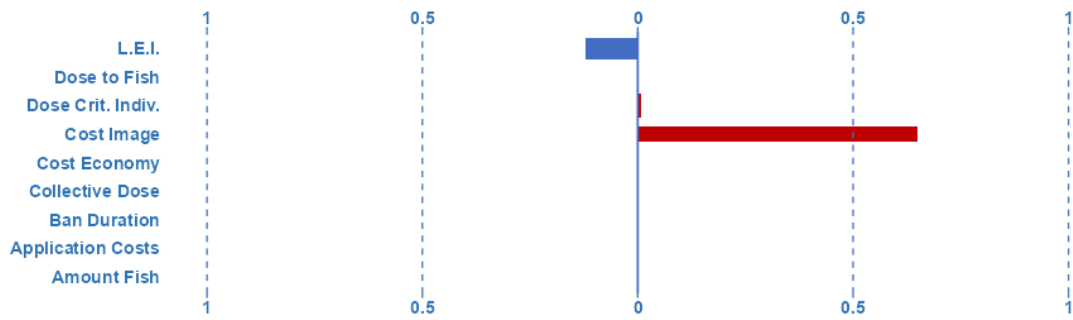
Comparative Chart - S1 vs S5




Comparative Chart - S1 vs S6



Comparative Chart - S1 vs S7



Este documento esta firmado por

| | | |
|--|-------------------------------|---|
|  | Firmante | CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES |
| | Fecha/Hora | Tue Jul 01 09:29:57 CEST 2025 |
| | Emisor del Certificado | EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES |
| | Numero de Serie | 561 |
| | Metodo | urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature) |