FM Fact Label: A Configurable and Interactive Visualization of Feature Model Characterizations

Jose M. Horcas
DiversoLab
University of Seville
Seville, Spain
jhorcas@us.es

Jose A. Galindo
DiversoLab
University of Seville
Seville, Spain
jagalindo@us.es

Mónica Pinto
CAOSD, ITIS
University of Málaga
Málaga, Spain
pinto@lcc.uma.es

Lidia Fuentes
CAOSD, ITIS
University of Málaga
Málaga, Spain
lff@lcc.uma.es

David Benavides
DiversoLab
University of Seville
Seville, Spain
benavides@us.es

ABSTRACT

Recognizing specific characteristics of feature models (FM) can be challenging due to the different nature and domains of the models. There are several metrics to characterize FMs. However, there is no standard way to visualize and identify the properties that make an FM unique and distinguishable. We propose *FM Fact Label* as a tool to visualize an FM characterization based on its metadata, structural measures, and analytical metrics. Although existing tools can provide a visualization of the FM and report some metrics, the feature diagram of large-scale FMs becomes ineffective to take an overall shape of the FM properties. Moreover, the reported metrics are often embedded in the tool user interface, preventing further analysis. *FM Fact Label* is a standalone web-based tool that provides a configurable and interactive visualization of FM characterizations that can be exported to several formats. Our contribution becomes important because the Universal Variability Language (UVL) is starting to gain attraction in the software product line community as a unified textual language to specify FMs and share knowledge. With this contribution, we help to advance the UVL ecosystem one step forward while providing a common representation for the results of existing analysis tools.

CCS CONCEPTS

• **Software and its engineering → Software product lines**; •**Human-centered computing → Information visualization**.

KEYWORDS

characterization,
feature model,
metrics,
variability,
visualization

1 INTRODUCTION

Feature models (FM) have become the de facto standard for modeling variability in software product lines (SPL), and the SPL community is betting high for the new *Universal Variability Language* (UVL) [20] as a proposal for a unified textual language to specify FMs. From 2018, the *MODEVAR* [3] initiative promotes not only the definition of a unified language, but also an ecosystem of tools supporting such a language. Recent contributions include the integration of UVL into well-known SPL tools such as FeatureIDE [21], transformation approaches such as TRAVART [9], a framework for automatic analysis [11], and a new repository for FMs [18].

In this paper, we contribute to the UVL ecosystem by providing *FM Fact Label*, a tool to visualize *FM characterizations*. A characterization of an FM is a description of the distinctive nature or properties of the FM. The characterization is often performed by providing a visual representation of the FM as a *feature diagram* [15] (usually an excerpt of the complete model), with a textual description of its features and relationships, and some metrics such as the number of features and configurations. In practice, when dealing with large-scale FMs, the visual representation of the FM as a feature diagram becomes ineffective and offers little value in identifying the properties that make an FM unique and distinguishable [15]. There are several catalogs of measures for FMs [2, 4, 5] that can be used to characterize an FM, but those measures have been proposed mainly for the quality assessment of FMs, concretely, to evaluate the maintainability of the FMs [2]. In contrast, we propose a characterization based on an extensive report in terms of metadata information, structural and syntatical measures, and analysis results that allow describing an FM. Although existing SPL tools (e.g., FeatureIDE [23], Glencoe [19]) compute syntatical and semantical statistics about the FM, they do not provide a visualization to effectively communicate these metrics and analysis results. Inspired by the nutrition fact label used in the food industry [10], *FM Fact Label* provides a visualization to display the FM characterization information effectively. The tool contributes as follows:

- A visual representation of the FM characterization in terms of its metadata, structural/syntantical metrics, and analytical/semantical metrics. We rely on the Python framework for automated analysis of FMs [11] to analyze the FMs and obtain the metrics.
- A standalone online web-based application to automatically generate the visualization as a fact label for FMs. We rely on data visualization applying its principles and best design practices to effectively communicate the information [17].
- An interactive and configurable visualization that allows customizing the FM properties to show. We rely on the Data-Driven Documents (D3) [7] approach for web-engineering which enables efficient manipulation of data-based documents.
- Exporting the visualization in different formats for communication (SVG, PNG) as well as exporting the complete characterization (in JSON or plain text) for further computing and analysis.

The tool is available online and ready to be used:
https://web.diverso-lab.us.es/fmfactlabel

The paper is structured as follows. Section 2 motivates our tool. Section 3 presents the proposed visualization for FM characterizations. Section 4 overviews the architecture of the tool and its functionality. Section 5 presents conclusions and future work.

2 STATE OF THE ART AND MOTIVATION

We discuss related work in FM characterization and visualization, and in FM metrics, and compare our tool with existing ones.

2.1 Characterization and visualization of FMs

In this paper, we refer to *FM characterization* as the process of identifying and describing the distinctive characteristics or properties of an FM that make it different or similar to other FMs. FMs are usually described by their features and the relations between these features, which determine the possible combinations that can be selected to form a valid configuration. On the one hand, there are several textual notations for FMs [22] that allow listing the features, relations, and constraints. In fact, the UVL [20] proposal is a unified textual language which allows specifying and editing large-scale FMs easily. However, it is very difficult to obtain an overall shape of the FM simply by observing its textual specification. On the other hand, a characterization based on the configurations exposed by the FMs is not viable in practice, since all configurations can be enumerated and analyzed only for small FMs. In this regard, Heradio et al. [12] propose a statistical approach to describe and interpret the variation of the features and products of large FMs. More formal characterizations of FMs have been proposed. Damiani et al. [8] propose a characterization based on propositional logic and extensional (algebraic) representations of the operations and relations of FMs. Although logical characterizations work well in practice for the automated analysis of FMs [14], they do not communicate the FM properties effectively in a human-readable format.

The most widespread description for FMs is the *feature diagram* [15] which graphically depicts the features and relationships in a tree-like structure with additional textual cross-tree constraints. Feature diagrams work nicely for small FMs, but become ineffective for large FMs because the resulting displays are too complicated for an overall visualization of the FM. As confirmed by Lopez-Herrejón et al. [15] in a mapping study of visualization and SPLs, the most common techniques used for FM visualization are trees and graphs, but they rely on basic visualization techniques and tools (e.g., ad hoc or based on Eclipse tools) that barely exploit the wealth of techniques available in the software and information visualization communities [17]. In contrast, our approach advocates for a standalone tool that relies on web engineering technologies, concretely in the Data-Driven Documents (D3) [7] approach which enables efficient manipulation of data-based documents and data visualization to represent an FM characterization.

2.2 FM metrics and tools support

A better way to characterize large-scale FMs is to use metrics [5]. Various measures for FMs have been proposed [2, 4, 5] such as the *COfFEE catalog* [4]. Most of these metrics are used mainly to measure the complexity of the FM and relate this complexity to the maintainability of the FM. In fact, a set of only four metrics (i.e., number of leaf features, number of constraints, number of valid configurations, and the ratio of optional features) was identified as the sufficient subset of the proposed metrics to assess the maintainability of a single FM [2]. However, metrics can be used to characterize FMs beyond supporting their quality evaluation. By including all fundamental information, not just maintenability metrics, our tool allows us to answer questions such as *which FM is more appropriate for teaching purposes?*, *which FMs can be used to evaluate a new fancy algorithm to optimize configurations?*, or *which are the properties of the FMs that affect more the scalability of your proposal?*. For instance, for teaching purposes, a small FM that includes at least one feature and relation of each type (e.g., an or-group, an xor-group, a requires and an excludes constraint, a dead-feature) is more appropriate than a large FM that cannot even be completely visualized in a feature diagram. In contrast, to evaluate the scalability of a new analysis operation, a large-scale FM with thousands of features and constraints may be preferable. Existing FM tools such as FeatureIDE [23], Glencoe [19] and DyMMer [6] provide a metrics report, but this report is usually embedded within the user interface (in the case of Glencoe) or hindered within the tools (e.g., as Eclipse properties in FeatureIDE) and do not provide a visualization or external report to effectively communicate the metrics and analysis results. Only DyMMer [6], which does not support analysis, allows exporting the metrics to CSV and PDF. Abuzaid and Scott [1] conducted a systematic review on visualization of software quality metrics, but it focuses on common metrics for software (e.g., lines of code) instead of FM metrics. Our proposed tool allows users to generate a visualization of FM metrics to effectively communicate the information simply, clearly, and accurately. Furthermore, our tool allows exporting the characterization and visualization to different formats (SVG, PNG, plain text, and JSON). Table 1 compares *FM Fact Label* with four well-known open source tools in SPL that report visualization of FM metrics.

Table 1: Comparison of existing tools to visualize FM metrics.

Characteristic	FeatureIDE [23]	Glencoe [19]	SPLOT [16]	DyMMer [6]	**FM Fact Label**
Visualization of the feature diagram	■	■	□	□	□
Editor for the FM	■	■	■	■	□
Support for UVL	■	□	□	□	■
Include metadata of the FM	□	□	■	■	■
Syntatical/Structural metrics	■	■	■	■	■
Semantical metrics (analysis)	■	■	■	□	■
Differentiate syntatical and semantical metrics	■	□	■	□	■
Customizable visualization of metrics	□	□	□	■	■
Interactive visualization of metrics	■	□	□	□	■
Export metrics in different formats	□	□	□	■	■
Online web-based application	□	■	■	■	■
Number of metrics	16	27	14	38	46
Last update date	Apr'22	May'20	Jan'15	Feb'22	Jun'22

■ It supports the characteristic. □ It does not support the characteristic.

3 A VISUAL CHARACTERIZATION FOR FMS

Figure 1 illustrates the proposed visualization for FM characterizations. The design of the visualization is inspired by the nutrition fact label used in the food industry for packaged foods [10] that clearly differentiates the different types of information. The FM information on the FM fact label is divided in four parts:

Name. The name of the FM is shown at top of the label. The name can be provided in a web form when uploading the FM; in other case, it is extracted from the filename.

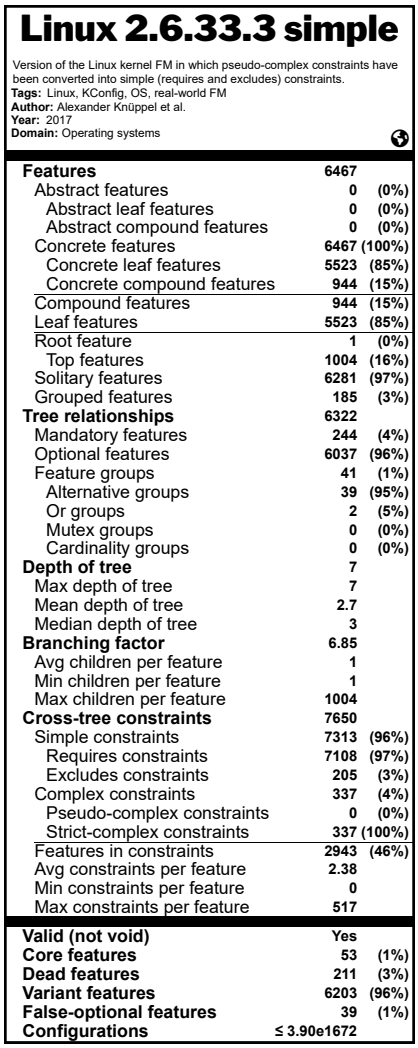

Figure 1: A fact label for the Linux FM.

Metadata. Just below the name, some optional metainformation of the FM can be displayed, such as a brief description, a list of tags or keywords related with the FM, the author(s) of the FM, the year of creation or publication, the domain, and a URL or DOI of the FM location or paper where it is published. All these information are manually provided in a web form.

Metrics. The structural measures or syntatical statistics of the FMs are shown at the center of the label. We have considered those metrics extracted from the literature [2, 4, 5] that are widely used for the quality evaluation of FMs. We have also considered additional concepts that can be measured in the FM such as the different types of intra-tree relationships (e.g., types of feature groups) or the different types of constraints (e.g., pseudo-complex constraints and strict-complex constraints [13]). All these measures can be directly obtain from the FM structure and from the list of constraints without requiring additional reasoning (e.g., a SAT solver). We classify and group the related metrics hierarchically. For each metric, we show its quantitative value and its ratio with regard to its immediate parent. For example, the `Requires constraints` metric shows the number of constraints of the type "$featureA$ `REQUIRES` $featureB$" and the percentage of requires constraints with regard the total number of simple constraints (i.e., requires and excludes).

Analysis results. The results of analyses of FMs or semantical metrics are shown at the bottom of the label. In contrast to the structural metrics, these measures are obtained by reasoning on the FMs (i.e., using a SAT or a BDD solver).

4 FM FACT LABEL TOOL

FM Fact Label is an online web-based application that builds an FM characterization and generates its visualization as a fact label.

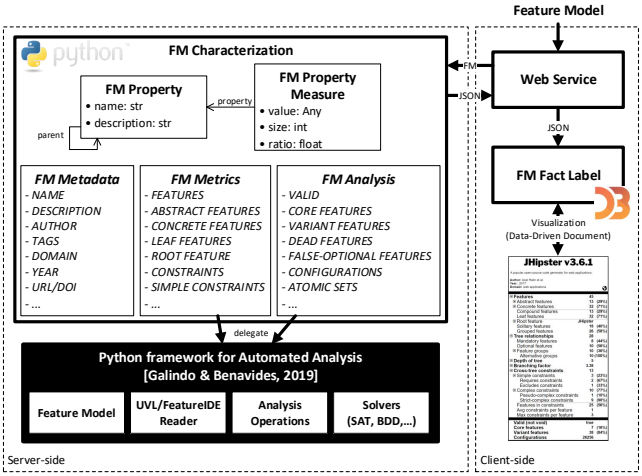

Figure 2: Architecture and technologies of *FM Fact Label*.

4.1 Software architecture and technologies

Figure 2 gives an overview of the architecture and technologies of the tool. It offers a web service providing an online form to upload the FM and its metadata. Currently, UVL and FeatureIDE formats are supported. The `FM Characterization` module is in charge of collecting all FM information. In the tool, we use the term *FM property* to refer to any FM characteristic regardless of its nature (i.e., metadata, structural metrics, or analysis results). An FM property contains a name and a description that are then shown in the visualization for documentation purposes, and a parent property to build the hierarchy of properties. Each FM property can be instantiated with an *FM Property Measure* that provides the value (e.g., the list of leaf features in the FM), its size (e.g., the number of leaf features), and its ratio (e.g., the percentage of leaf features *wrt* the total number of features in the FM). To obtain these data, the tool delegates the analysis to different external third-party tools for FM analysis. Currently, our tool delegates to the Python framework for automated analysis [11]. At this date, the FM characterization provides up to 46 measures, including metrics and analysis results, and it is open to extension with further metrics from the SPL literature. Once the FM characterization is built, its fact label visualization is automatically generated on the client side using D3 [7]. D3 relies on web standards (i.e., HTML, CSS, JavaScript, SVG, and JSON) to combine visualization components and a data-driven approach that allows binding arbitrary data to a Document Object Model (DOM), and then applying data-driven transformations to the DOM. The tool benefits from D3 to provide an interactive and configurable visualization of the FM characterization.

4.2 Interactive and configurable visualization

Figure 3 schematizes the visualization and functionalities of the tool. First, all metadata are optional, so the user only needs to provide the desired information to be included in the visualization. The URL/DOI of the FM file or publication, if provided, is encoded in a graphical clickable icon. Second, each FM property can be collapsed or expanded to hide or show the desired subproperties. Furthermore, for each property, its description is shown as a mouse-over tooltip explaining the meaning of the property, and the concrete value (e.g., the list of constraints) is shown when clicking on the

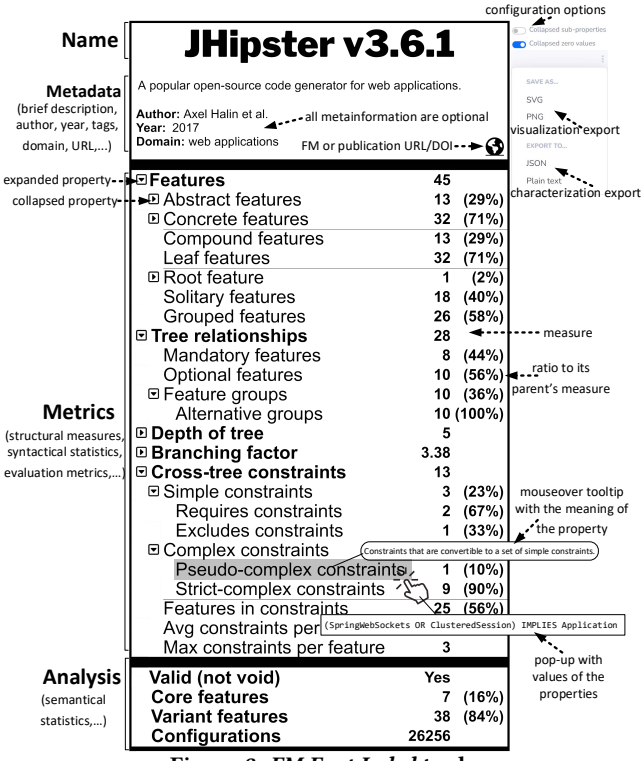

Figure 3: *FM Fact Label* tool.

property. Third, additional configurable options are provided to collapse/expand all subproperties at once or to hide those properties whose values are zero or empty. The configured visualization can be exported in high quality format (i.e., SVG) to be included, for example, in research publications, or exported as a raster graphic (i.e., PNG) for portability on the network. Finally, the complete FM characterization can also be exported in JSON format or plain text for further computation or analysis.

5 CONCLUSIONS AND FUTURE WORK

FM Fact Label is a tool to generate visualizations of FM characterizations based on FM metadata, syntactical and analytical metrics. The tool can be part of the UVL ecosystem and its architecture can be easily extended to work with other analysis tools for FMs and to support additional metrics. With this contribution, we help SPL practitioners in identifying the distinctive characteristics of the FMs and select the most appropriate FM for their needs. We envision that *FM Fact Label* can be integrated in different tools as an independent artifact and we plan to integrate it in the under development UVL reposistory [18].

As future work, we plan to incorporate additional features such as thresholds for the metrics, and configuration options such as a horizontal visualization of the label. Moreover, we plan to use the FM characterizations to propose advanced analysis such as studying the properties that most affect the performance of a given analysis operation or calculating some analytical metrics (e.g., number of configurations) from the structural properties of the FM by using, for example, neural networks.

OPEN SCIENCE

Tool: https://web.diverso-lab.us.es/fmfactlabel

Code repository and video demo: https://github.com/jmhorcas/fm_characterization

ACKNOWLEDGMENTS

Work supported by the projects *OPHELIA* (RTI2018-101204-B-C22), *COPERNICA* (P20_01224), *METAMORFOSIS* (FEDER_US-1381375), *MEDEA* (RTI2018-099213-B-I00), *IRIS* (PID2021-122812OB-I00), *Rhea* (P18-FR-1081), *LEIA* (UMA18-FEDERIA-157), and *DAEMON* (H2020-101017109), and Juan de la Cierva—Formación 2019 grant.

REFERENCES

[1] Dur Abuzaid and Scott Titang. 2014. *The Visualization of Software Quality Metrics-A SLR.* Bachelor thesis. Univ. of Gothenburg. Chalmers Univ. of Technology.

[2] Ebrahim Bagheri and Dragan Gasevic. 2011. Assessing the maintainability of software product line feature models using structural metrics. *Softw. Qual. J.* 19, 3 (2011), 579–612. https://doi.org/10.1007/s11219-010-9127-2

[3] David Benavides, Rick Rabiser, Don S. Batory, and Mathieu Acher. 2019. First international workshop on languages for modelling variability (MODEVAR). In *23rd SPLC*, Vol. A. 46:1. https://doi.org/10.1145/3336294.3342364

[4] Carla Ilane Moreira Bezerra, Rossana M. C. Andrade, and José Maria Monteiro. 2015. Measures for Quality Evaluation of Feature Models. In *14th ICSR*, Vol. 8919. 282–297. https://doi.org/10.1007/978-3-319-14130-5_20

[5] Carla I. M. Bezerra, Rossana M. C. Andrade, and José Maria Monteiro. 2017. Exploring quality measures for the evaluation of feature models: a case study. *J. Syst. Softw.* 131 (2017), 366–385. https://doi.org/10.1016/j.jss.2016.07.040

[6] Carla I. M. Bezerra, Rafael Lima, and Publio Silva. 2021. DyMMer 2.0: A Tool for Dynamic Modeling and Evaluation of Feature Model. In *35th Brazilian Symposium on Software Engineering (SBES)*. 121–126. https://doi.org/10.1145/3474624.3476016

[7] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3 Data-Driven Documents. *IEEE Trans. Vis. Comput. Graph.* 17, 12 (2011), 2301–2309. https://doi.org/10.1109/TVCG.2011.185

[8] Ferruccio Damiani, Michael Lienhardt, and Luca Paolini. 2022. On logical and extensional characterizations of attributed feature models. *Theor. Comput. Sci.* 912 (2022), 56–80. https://doi.org/10.1016/j.tcs.2022.01.016

[9] Kevin Feichtinger, Johann Stöbich, Dario Romano, and Rick Rabiser. 2021. TRAVART: An Approach for Transforming Variability Models. In *VaMoS*. 8:1–8:10. https://doi.org/10.1145/3442391.3442400

[10] Food and Drug Administration, HHS. 2016. Food labeling: revision of the nutrition and supplement facts labels. *Federal register* 81, 103 (2016), 33741–33999.

[11] José A. Galindo and David Benavides. 2020. A Python framework for the automated analysis of feature models: A first step to integrate community efforts. In *24th SPLC*, Vol. B. 52–55. https://doi.org/10.1145/3382026.3425773

[12] Ruben Heradio, David Fernández-Amorós, Christoph Mayr-Dorn, and Alexander Egyed. 2019. Supporting the statistical analysis of variability models. In *41st ICSE*. 843–853. https://doi.org/10.1109/ICSE.2019.00091

[13] Alexander Knüppel, Thomas Thüm, Stephan Mennicke, Jens Meinicke, and Ina Schaefer. 2018. Is There a Mismatch between Real-World Feature Models and Product-Line Research?. In *ESEC/FSE*. 53–54. https://dl.gi.de/20.500.12116/16312

[14] Jia Hui Liang, Vijay Ganesh, Krzysztof Czarnecki, and Venkatesh Raman. 2015. SAT-Based Analysis of Large Real-World Feature Models is Easy. In *19th SPLC*. 91–100. https://doi.org/10.1145/2791060.2791070

[15] Roberto Erick Lopez-Herrejon, Sheny Illescas, and Alexander Egyed. 2018. A systematic mapping study of information visualization for software product line engineering. *J. Softw. Evol. Process.* 30, 2 (2018). https://doi.org/10.1002/smr.1912

[16] Marcilio Mendonca, Moises Branco, and Donald Cowan. 2009. S.P.L.O.T.: Software Product Lines Online Tools. In *OOPSLA*. https://doi.org/10.1145/1639950.1640002

[17] Stephen R. Midway. 2020. Principles of Effective Data Visualization. *Patterns* 1, 9 (2020), 100141. https://doi.org/10.1016/j.patter.2020.100141

[18] David Romero, José A. Galindo, José Miguel Horcas, and David Benavides. 2021. A first prototype of a new repository for feature model exchange and knowledge sharing. In *25th SPLC*, Vol. B. 80–85. https://doi.org/10.1145/3461002.3473949

[19] Anna Schmitt, Christian Bettinger, and Georg Rock. 2018. Glencoe – A Tool for Specification, Visualization and Formal Analysis of Product Lines. In *25th Trans. Eng.* 665–673. https://doi.org/10.3233/978-1-61499-898-3-665

[20] Chico Sundermann, Kevin Feichtinger, Dominik Engelhardt, Rick Rabiser, and Thomas Thüm. 2021. Yet another textual variability language?: a community effort towards a unified language. In *25th SPLC*, Vol. A. 136–147. https://doi.org/10.1145/3461001.3471145

[21] Chico Sundermann, Tobias Heß, Dominik Engelhardt, Rahel Arens, Johannes Herschel, Kevin Jedelhauser, Benedikt Jutz, Sebastian Krieter, and Ina Schaefer. 2021. Integration of UVL in FeatureIDE. In *25th SPLC*, Vol. B. 73–79. https://doi.org/10.1145/3461002.3473940

[22] Maurice H. ter Beek, Klaus Schmid, and Holger Eichelberger. 2019. Textual variability modeling languages: an overview and considerations. In *23rd SPLC*, Vol. B. 82:1–82:7. https://doi.org/10.1145/3307630.3342398

[23] Thomas Thüm, Christian Kästner, Fabian Benduhn, Jens Meinicke, Gunter Saake, and Thomas Leich. 2014. FeatureIDE: An extensible framework for feature-oriented software development. *Sci. Comput. Program.* 79 (2014), 70–85. https://doi.org/10.1016/j.scico.2012.06.002