# START PAGE

MARIE SKLODOWSKA-CURIE ACTIONS

**Individual Fellowships (IF)**
**Call: H2020-MSCA-IF-2014**

PART B

"MoStMusic"

**This proposal is to be evaluated as:**

**[Standard EF]**

## TABLE OF CONTENTS

# 0 LIST OF PARTICIPANTS

| Participants | Legal Entity Short Name | Academic | Non-academic | Country | Dept. / Division / Laboratory | Supervisor | Role of Partner Organisation |
|---|---|---|---|---|---|---|---|
| Beneficiary: Universiteit Utrecht | UU | ✓ | | The Netherlands | Information and Computing Sciences | Remco Veltkamp | |
| Partner organisation: Microsoft Research | MSR | | ✓ | United Kingdom | Programming Principles and Tools | Simon Peyton Jones | Secondment |

Data for non-academic beneficiaries:

| Name | Location of research premises | Type of R&D activities | No. of fulltime employees | No. of employees in R&D | Website | Annual turnover (approx. in Euro) | Enterprise status | SME status |
|---|---|---|---|---|---|---|---|---|
| Microsoft Research | Cambridge, UK | Computer science | 1500 | 1100 | http://research.microsoft.com | 100M | Yes | No |

# 1 SUMMARY

Music is an art form with a very long history, and continues to engage millions of people today. Music Information Retrieval (MIR), the exciting interdisciplinary science that brings together music and computer science, is a growing field of research with the potential to enrich pure computer science knowledge while creating real-world applications that the general public can benefit from. While the marriage of art and science is often troublesome, MIR has the benefit that many aspects of music are highly structural and have been subject to rigorous formalisation for a long time. Formalisation and computers go hand in hand, and MIR researchers have therefore been developing models of musical structure for many years, and putting them to use in several applications. However, such models, so far, have had limited impact; they are commonly restricted to one specific aspect of music (such as harmony or form), can be hard to implement computationally (due, for example, to the way ambiguity is handled), and are often too technical to be used directly by musicologists who are not familiar with programming language details.

However, models are valuable. Unlike machine learning approaches, model-based MIR provides a real insight about the underlying structure, and can benefit from the input of musicologist experts. Furthermore, a single model can be applied to multiple important MIR tasks (such as retrieval, analysis, and automatic composition). The research goal of this project is thus to *give musical models the impact they deserve*, advancing the practical embodiment of hierarchical musical structure—in its various forms—in computer science through the development of new, functional Models of Structure in Music (MoStMusic). Specifically, I intend to develop functional models of musical form, melody, and harmony that enable an easy, fast, and flexible way of creating model-enhanced MIR applications. Being executable, these models will pave the way for true content-based music analysis and retrieval—an underestimated and underexplored area. As a showcase of a model-enhanced application, I will create an online music analyser that automatically computes the structure present in a user-submitted piece, and displays it in an interactive interface that highlights the *structural shape* of music.

I have previously developed a functional model of structure in musical harmony, creating a model that is easy to understand, directly linked to an efficient implementation, and applicable to several common MIR tasks. This work lead to several publications in top-quality venues, and to the creation of a very successful internet start-up company. With the foundational theory already in place, and given the growing interest in model-based approaches in the MIR community, the time is right for pursuing this line of research. This promising undertaking will greatly expand my knowledge in MIR and musicology, and help me develop into a leading independent researcher. At the same time, the proposed application has great outreach and commercialisation potential, thus contributing to the research output, visibility, and economy of Europe.

# 2 EXCELLENCE

## 2.1 Quality, innovative aspects and credibility of the research

This project deals with modelling the structure of different aspects of music. Figure 1 shows a very simple example model of harmony in music, expressed in a context-free grammar style. *Models in MIR are important*: they provide insight about the underlying structure, can make use of centuries of accumulated knowledge of musical analysis, and can be applied to multiple important MIR tasks. Models have previously been shown to improve several MIR tasks, such as chord transcription,[1] structural analysis,[2] and similarity estimation.[3] This project is centred around the development of models of musical structure for several aspects of music, and a showcase application that highlights the importance and reaps the benefits of modelling structure in music.

$$
\begin{array}{rcl}
\text{Piece} & \rightarrow & [\text{Phrase}] \\
\text{Phrase} & \rightarrow & \text{Ton Dom Ton} \mid \text{Dom Ton} \\
\text{Ton} & \rightarrow & I \\
\text{Dom} & \rightarrow & V \mid V^7 \mid vii^o \mid II^7\ V^7
\end{array}
$$

*Figure 1: A simple model of harmony.*

[1] M. Mauch, K. Noland, and S. Dixon. "Using Musical Structure to Enhance Automatic Chord Transcription". In: *10th International Society for Music Information Retrieval Conference.* 2009, pp. 231–236.

[2] J. Paulus, M. Müller, and A. Klapuri. "State of the Art Report: Audio-Based Music Structure Analysis". In: *11th International Society for Music Information Retrieval Conference.* Aug. 2010, pp. 625–636.

[3] W. B. de Haas, J. P. Magalhães, R. C. Veltkamp, and F. Wiering. "HarmTrace: Improving Harmonic Similarity Estimation Using Functional Harmony Analysis". In: *12th International Society for Music Information Retrieval Conference.* 2011, pp. 67–72.

There are multiple aspects to music which come together to form a piece. For the purposes of MoStMusic, we focus on three specific aspects: harmony, melody, and form. *Harmony* occurs when multiple notes sound at the same time in the form of a chord. The succession of chords in music is highly structured, as some chords "lead" to others, in a pattern of tension and resolution. *Melody* is a linear succession of notes. It is also called the horizontal aspect of music (with harmony being the vertical aspect). Melodies have their own internal structure which can involve sub-melodies (phrases) or repeating thematic figures. *Form* is the overall structure of an entire piece. It deals with repetition, variety, contrast, and connection. For example, the sonata form generally consists of exposition, development, and recapitulation. In all these aspects, *structure* plays a crucial role. It is this structure that I intend to explore in this project by developing specialised models of musical aspects that make their structure manifest.

Others have looked at the problem of modelling structure in music in a computational setting. Baroni and colleagues looked specifically at grammars for melody,[4] but their work seems not to have been implemented as a computer program. Lerdahl and Jackendoff,[5] formalising the ideas of Schenker,[6] paved the way for a structured analysis of Western tonal compositions by defining hierarchical dependency relationships between musical elements using well-formedness and constraint-based preference rules. Later, Temperley[7] elaborated the ideas of Lerdahl and Jackendoff, and implemented them computationally using preference rule systems for six basic kinds of musical structure: metrical, melodic phrase, counterpoint, pitch spelling, harmony, and key. Temperley's work made an impressive contribution to the field of computational modelling of musical structure. However, his models suffer from two significant shortcomings. Firstly, they are devised mostly independently of each other, and then connected in an adhoc fashion. For example, the metrical algorithm is first run in a basic setting, its output fed to the harmonic algorithm in "prechord" mode, and that output fed back into the metrical algorithm, which can then finally run in full mode. Secondly, the connection between the model's preference rules and the actual implementation is largely detached, due to the choice of a low level programming language (C). This makes the models impossible to experiment with for non-programmers, and creates a large semantic gap between the conceptual model and its implementation.

MoStMusic comes to address these limitations, proposing the development of a unified model of three aspects of musical structure, represented in a high-level language with a user-friendly notation, and showcased in a practical, real-world application. One key difference between my approach to modelling music and existing work is the reliance on advanced functional programming techniques. From my background as a functional programmer and programming languages (PL) researcher, I have seen that the choice of language is an important factor in the success of accomplishing a given programming task. I have worked on a model of harmony using Haskell,[8] a statically-typed functional programming language, and compared this model to an earlier implementation in Java. The conclusion was that the Haskell implementation outperformed the Java approach in terms of speed, functionality, and simplicity.[9] The key aspect of this work was to encode musical harmony as a *datatype*. Values of this datatype are then valid harmony sequences, according to the rules of harmony as defined by the model. This encoding brings the advantage that harmony sequences which do not make sense according to the model cannot be represented; the type-checker forbids this, at compilation time. This aspect is the crux of the interdisciplinary nature of this approach: it unifies the validity of a harmony sequence (something of a musical, artistic nature) with *type checking* (something technical and engineered).

But why functional programming, and why model the structure as a datatype? Naturally, other methods could be used to check or enforce the validity of harmony sequences in the model. For example, a dedicated routine could be written for this purpose, working only on harmony sequences (unlike the type checker,

[4] M. Baroni, S. Maguire, and W. Drabkin. "The Concept of Musical Grammar". In: *Music Analysis* 2.2 (1983), pp. 175–208.

[5] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music.* MIT Press series on Cognitive theory and mental representation. MIT Press, 1985. ISBN: 9780262260916.

[6] H. Schenker and E. Oster. *Free Composition: Volume III of New Musical Theories and Fantasies.* Distinguished reprints series. Pendragon Press, 1979. ISBN: 9781576470749.

[7] D. Temperley. *The Cognition of Basic Musical Structures.* MIT Press, 2004. ISBN: 9780262701051.

[8] S. Peyton Jones, ed. *Haskell 98, Language and Libraries. The Revised Report.* Journal of Functional Programming Special Issue 13(1). Cambridge University Press, 2003.

[9] J. P. Magalhães and W. B. de Haas. "Functional modelling of musical harmony: an experience report". In: *16th ACM SIGPLAN International Conference on Functional Programming.* ACM, 2011, pp. 156–162.

which works on any type of data). However, relying on the type checker means we do not have to do any extra work to guarantee validity of the harmony sequences. Furthermore, the type-safe encoding of harmony sequences allows us to leverage the power of another functional programming tool: *generic programming.*[10] Generic programs operate on the structure of datatypes, and can perform a myriad of tasks simply by structural means. Examples of generic programs are: computing the equality between two terms, (de-)serialisation, traversals, and queries. The more complex the structure of the datatype, the more advanced the behaviour of the generic program can be. In the particular case of a datatype encoding musical harmony, I have explored generic pretty-printing and parsing,[9] edit-distance computation,[3] and automatic data generation.[11] These are standard generic programs, applicable to any sort of data structure (e.g. binary trees, lists, abstract syntax trees), which get an MIR-specific meaning when applied to a datatype that encodes harmony sequences:

- Pretty-printing is the task of displaying a harmony sequence in a hierarchical representation.
- Parsing is the task of converting a textual representation of chords into a harmony representation as a tree. Furthermore, by using advanced parsing techniques such as error-correction,[12] we can deal with the problem of encoding sequences that are not immediately valid according to the model, all by leveraging existing solutions from the PL community.
- Edit-distance computation[13] is the task of harmonic similarity estimation. In particular, this can be used to find cover songs, as those tend to be significantly different in many aspects of music, but preserve the same harmony.
- Data generation is the task of automatic harmony generation, which can be used in the creation of computer music, or as an aid in composition.

This is remarkable: one single set of datatypes encoding musical structure can be used to implement or improve several distinct MIR tasks. Moreover, due to the use of generic programming, we can easily change the models (e.g. to deal with the differences between classical baroque harmony and contemporary pop music) and the algorithms will continue to work, as they are not specialised to a single concrete datatype.

This powerful synergy revealed in my previous research has shown that this strategy is sound and full of potential for further investigation. MoStMusic comes to expand and broaden the scope of this line of research, with the following concrete goals:

**Goal 1.** Create functional models for *form and melody.* Since these aspects possess deep structure, and have been studied in great detail by musicologists over the years, it should be possible to define functional models in a similar way to the existing harmony model. Furthermore, the current harmony model does not deal with modulation. Extending the model to *support modulation* is a crucial step, as modulation is an essential aspect of harmony. Finally, because these aspects interact with each other, it is crucial to be able to combine models of different aspects into a single, *unified model*, that can address all aspects as a coherent unit. Like in my previous work, these models can be beneficial in analysing music both from audio or from a symbolic representation (such as MIDI); I intend to explore both representations. This goal is addressed by WP1 (see Section 4.1).

**Goal 2.** Make the model available in a concise, clear, and efficient way, through the development of an embedded domain specific language (EDSL) for modelling music. Haskell is known for its ability to encode specific domains with a clear syntax that can be understood and used by domain experts with no Haskell knowledge. Hudak developed an EDSL for encoding musical aspects in Haskell.[14] Hudak's EDSL is mostly concerned with representation of music (such as individual notes and their duration), not with high-level aspects such as melody and harmony structure. Developing an EDSL for high-level musical structure which builds upon my models bridges the semantic gap between the conceptual model and its implementation, and

[10] J. P. Magalhães. "Less Is More: Generic Programming Theory and Practice". PhD thesis. Universiteit Utrecht, 2012.

[11] J. P. Magalhães and H. V. Koops. "Functional Generation of Harmony and Melody". In: *2nd ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design.* ACM, 2014.

[12] S. D. Swierstra and L. Duponcheel. "Deterministic, Error-Correcting Combinator Parsers". In: *Advanced Functional Programming, Second International School.* Springer-Verlag, 1996, pp. 184–207. ISBN: 3-540-61628-4.

[13] E. Lempsink, S. Leather, and A. Löh. "Type-safe Diff for Families of Datatypes". In: *2009 ACM SIGPLAN Workshop on Generic Programming.* Edinburgh, Scotland: ACM, 2009, pp. 61–72.

[14] P. Hudak. *The Haskell School of Expression: Learning Functional Programming Through Multimedia.* New York, NY, USA: Cambridge University Press, 2000. ISBN: 0-521-64408-9.

brings the usage of the models within easier reach of e.g. computational musicologists and composers. This EDSL is one of the deliverables of WP3 (D3.1, see Section 4.1).

A good implementation of such an EDSL might well require further advances in programming languages themselves. The previous harmony model made use of generalised algebraic datatypes in order to encode concepts such as mode (major or minor) and secondary dominants at the level of types. This, in turn, gave rise to the need for a generic programming library that could handle such datatypes.[15] I expect that developing this EDSL will equally bring forward the need for more expressive and efficient generic programming support. GHC, the main compiler for the Haskell language in which I intend to develop these models, is a research product from Microsoft Research; this project includes a secondment at their laboratory in order to facilitate the integration of new language support in the compiler (WP2 in Section 4.1).

**Goal 3.** Develop an *online music analyser* which highlights the form, melody and harmony structure of arbitrary (user-supplied) pieces of music. This is an application of a unified model of music, and one of the deliverables of WP3 (D3.2, see Section 4.1). I have previously investigated applications of a harmony model to: content-based retrieval,[9] similarity estimation,[3] chord recognition from audio,[16] and automatic composition.[11,17] The new models will have similar applications, but I choose to focus on analysis and its display because such an application makes the model entirely manifest, highlighting its importance. As an example, Figure 2 shows the harmony structure of a simple piece (for a model that is slightly more expressive than that of Figure 1). In the online application I will develop, users will be able to submit their own pieces for analysis, obtain similar trees, and jump to the section where the dominant chord appears by clicking on the Dom node. This example shows only harmony, but melody and form will be treated similarly, allowing users to e.g. jump to the chorus, or to visualise the repeated occurrence of the theme in the melody.

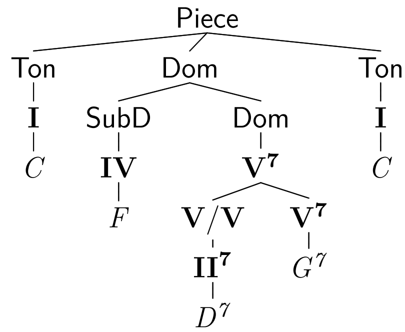


*Figure 2: The harmony structure of a chord sequence in the key of C major.*

Such an application will impact research dissemination to the public and commercialisation: my earlier work on chord recognition gave rise to the successful internet start-up Chordify (http://chordify.net), which brings automatically transcribed chords (using my functional model of harmony) to millions of users each month. The dissemination aspects are described in detail in Section 3.2.

It is the right moment to develop these goals. We have the foundational theory in place, the know-how to develop applications, and online tools that facilitate access to musical sources. What we miss is a unified way of *exploring and exploiting musical structure.* Google Image search provides an easy way to find images based on their content: containing a specific colour, or of a given type (e.g. photo, clipart, animated). We lack a similar mechanism for finding music: searching for songs in sonata form, or containing a specific harmony progression, for example. This is the kind of technology that MoStMusic will enable.

The goals of the project will be accomplished through the use of "build" and "model" methodologies. The model methodology deals with defining an abstraction for a real system; here, the system is music, in all its complexity as found in the real world, and the model is the abstraction as a datatype in Haskell. The build methodology is centered around a software system; these will be the online music analyser application, the improvements to the language for supporting a solid musical structure EDSL, and the EDSL itself. These are approaches that I have used previously in my research, and that I am comfortable with. Modelling musical aspects will require me to first deepen my knowledge from a musical-theoretical point of view. The implementation of language support will require guidance from an experienced compiler developer. Both of these are accounted for in this proposal (described in detail in Section 4.1).

---

[15] J. P. Magalhães and J. Jeuring. "Generic programming for indexed datatypes". In: *7th ACM SIGPLAN Workshop on Generic Programming.* Tokyo, Japan: ACM, 2011, pp. 37–46.

[16] W. B. de Haas, J. P. Magalhães, and F. Wiering. "Improving Audio Chord Transcription by Exploiting Harmonic and Metric Knowledge". In: *13th International Society for Music Information Retrieval Conference.* Porto, Portugal: FEUP Edições, 2012, pp. 295–300. ISBN: 978-972-752-144-9.

[17] H. V. Koops, J. P. Magalhães, and W. B. de Haas. "A Functional Approach to Automatic Melody Harmonisation". In: *1st ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design.* Boston, Massachusetts, USA: ACM, 2013, pp. 47–58.

### 2.2 Clarity and quality of transfer of knowledge/training for the development of the researcher in light of the research objectives

This fellowship proposal is designed to allow me to expand my knowledge and research field significantly. I would be working under the direct supervision of Prof. Remco Veltkamp (whose qualifications and experience are detailed in Section 2.3) in the "Interaction Technology" group of the Department of Information and Computing Sciences. Prof. Veltkamp leads the group, which consists of 26 members working in game technology, media technology, and human-media interaction. In the past, I have always worked within software technology or programming languages research groups; joining the Interaction Technology group would mark a new career step for me, bringing me closer to the MIR research community.

This group produces leading research in the MIR field, and I would benefit from the knowledge of several of its members. *Dr. Frans Wiering* is an Assistant Professor with a background in musicology. His research is at the intersection of computer science and music, connecting computer science methodology to state-of-the-art domain knowledge of music. He was General Chair of ISMIR 2010, the annual conference of the International Society for Music Information Retrieval, and will be Program Chair of ISMIR 2015. This conference is the world's leading interdisciplinary forum on digital music, and was held in Utrecht in 2010. He is also the chair of International Musicological Society's study group in Digital Musicology. Dr. Wiering's knowledge and musicology expertise will be invaluable in modelling musical aspects, as he can offer a perspective that is purely musical and not biased towards specific implementation details. *Dr. Anja Volk* is an Assistant Professor and laureate of the prestigious Vidi programme of the Netherlands Organization for Scientific Research, heading the MUSIVA research project. Dr. Volk's project MUSIVA deals with modelling musical similarity, and her expertise will be crucial for the development of my own models of musical aspects, as similarity, variation, and repetition are essential when modelling harmony, melody, and form. *Dr. W. Bas de Haas* is a postdoc in Dr. Volk's MUSIVA project. His research focuses on developing MIR methods that are founded in music cognition and perception. His PhD thesis[18] deals with the development of a model of musical harmony and its applications, and I have collaborated with him on several occasions.

Next to the Interaction Technology group, the department is also home to the Software Technology group, where I completed my PhD between 2008 and 2012. *Prof. Johan Jeuring* is professor of Software Technology for Learning and Teaching at UU, professor of Software Technology at the School of Computer Science of the Open University, and was my PhD supervisor. He is a world expert in the PL field of generic programming, and lately has focused on applying generic programming to learning and teaching, showcasing the advantages of a multidisciplinary approach to PL research. The Software Technology group also hosts the development of the Utrecht Haskell Compiler (UHC). The UHC, mostly developed by *Dr. Atze Dijkstra*, is a full-fledged compiler, and an ideal research vehicle for prototyping new language extensions or improvements, such as those required for Goal 2 of MoStMusic. I have previously worked with Dr. Dijkstra in implementing a Haskell language extension for generic programming in Haskell, which I later also implemented in GHC.[19]

UU also offers a wide range of personal development and complementary skills workshops and courses which I intend to participate in to further develop my skills. I plan to improve my knowledge of the Dutch language through courses offered by Babel, a Utrecht-based language institute. The faculty of Science offers training courses in teaching and supervising research, two areas in which I have no formal training but which are essential to a leading academic. The university offers coaching programmes, training courses, and informational sessions on how to successfully apply for a variety of European and Dutch funding programmes. Through the Utrecht Valorisation Centre and the Utrecht Center for Entrepreneurship I can obtain training dedicated to application and transfer of knowledge, and entrepreneurship.

This project includes a secondment at Microsoft Research (MSR) in Cambridge, UK, under the supervision of *Simon Peyton Jones*. Peyton Jones is the main developer of GHC and one of the most important PL researchers in the world, having authored over 300 publications with over 16000 citations in total. Working under his supervision at MSR would allow me to deepen my knowledge in compiler technology and implementation, while improving language support for the EDSL of Goal 2.

I will also transfer my knowledge to the host institute. I will participate in the regular group meetings,

[18] W. B. de Haas. "Music information retrieval based on tonal harmony". PhD thesis. Universiteit Utrecht, 2012.

[19] J. P. Magalhães, A. Dijkstra, J. Jeuring, and A. Löh. "A generic deriving mechanism for Haskell". In: *3rd ACM Haskell Symposium on Haskell*. ACM, 2010, pp. 37–48.

and give talks at the Computing Science Colloquium (a weekly series of talks) and the yearly departmental meeting. On the educational side, I shall participate in MSc and PhD student co-supervision, contribute to teaching (on courses such as Multimedia Retrieval, Sound And Music Technology, and Advanced Functional Programming), and I will (co-)organise a Summer School (see D4.2 in Section 4.1). As a PL expert, I will also advise the group on specific languages, tools, and libraries that suit their needs.

### 2.3 Quality of the supervision and the hosting arrangements

UU is a top-tier university with excellent facilities for hosting researchers; more details are given in Section 6. The hosting group (Interaction Technology) produces top-quality research in the field, and plays a key role in one of the university's focus areas (Game Research). Next to the hosting group, the Software Technology group, which I will be in close contact to, hosts some of the world's best researchers in the fields of functional programming and generic programming. Several people (mentioned in Section 2.2) will be available for collaborative research and exchange of ideas. This unique combination makes UU the only place in the world where MoStMusic can thrive.

#### Qualifications and experience of the supervisor(s)

Prof. Veltkamp is a leading researcher in media technology, having authored over 200 refereed papers in reviewed journals and conferences with over 5500 citations in total in the past 20 years. His h-index is 33 (citation statistics from http://scholar.google.co.uk/citations?user=xGkWUqMAAAAJ). He was Program Chair of ISMIR 2010, and organised the Dagstuhl Seminars on Content-Based Retrieval. His past involvement in research projects related to musical modelling include the WITCHCRAFT and Orpheus projects. Currently he is involved in the MUSIVA, COGITCH, COMMIT, and GALA projects (more details in Section 6). He has previously supervised 13 PhD students and 7 postdocs, and currently supervises 9 PhD students and 6 postdocs. People he mentored went on to develop very successful careers: Dr. Anja Volk acquired a prestigious personal grant and has become assistant professor, Dr. Marc Bron continued his research career at Yahoo Research in London, and Dr. Reinier van Leuken moved on to industry and became director of Quintic France, to name a few examples.

Prof. Veltkamp and I will draft a personal career development plan that covers not only scientific development but also how to shape my career and how to successfully valorise my research. In addition to this plan, I will be assigned a mentor that will be involved in providing support and guidance for my personal and professional development. There will be an Annual Assessment and Development Interview, in which my career development plan will be discussed, as well as personal developments and ambitions. These plans will be written down, and discussed again in the next meeting.

### 2.4 Capacity of the researcher to reach and re-enforce a position of professional maturity in research

My background places me at the right intersection between PL and MIR; next to my Computer Science and Systems Engineering degree and my PhD in generic programming, I also have a degree in music (high-school level), which gave me foundational knowledge in musical analysis and composition.

I am at the perfect point in my career to plan and develop my own research project. Since obtaining my PhD from UU in 2012, I have been a postdoc working in generic programming at the University of Oxford (OU). Through these years, I have developed an excellent publication record in both the PL and MIR fields, with publications in a top PL conference (ICFP) and journal (JFP), and a top MIR conference (ISMIR) and journal (CMJ). All my publications are listed in Section 5. I have over a decade of experience with functional programming, contributed to numerous Haskell libraries, and implemented my research ideas in the most important Haskell compilers. I have given an invited talk about functional modelling of harmony and its applications at ICT.OPEN in the Netherlands in 2012, and have given radio and newspaper interviews about my work on Chordify. I was program committee (PC) member of the ACM Workshop on Functional Art, Music, Modelling and Design (2013), the ACM Haskell Symposium (2014), and PC chair of the ACM Workshop on Generic Programming (2014). Furthermore, I was invited to be the Music Chair of ISMIR 2015. I am also an invited observer of the International Federation for Information Processing Working Group 2.1 on Algorithmic Languages and Calculi. I have shown that I can work independently, having collaborated with dozens of co-authors (many of them top researchers in their field), and initiating collaborations on my own: for example, I initially approached Dr. de Haas regarding his model of harmony implemented in Java, and offered the initial functional implementation, which eventually led to a series of publications, and my introduction to the MIR field. I also have solid and proven skills in commercial development of scientific research and knowledge transfer by co-founding Chordify, an internet start-up that attracts over 100.000 visits a day and has been featured in numerous media outlets around the world.

In the future, and in particular through the means offered by this fellowship, I wish to position myself as an experienced researcher working both in academia and the private sector, by conducting excellent scientific research and subsequently promoting its commercial development. In particular, I plan to explore the funding opportunities offered by the Dutch Technology Foundation STW, which funds projects promoting the transfer of knowledge between the technical sciences and users. I will also be at the right stage for applying for funding where I can lead a project within a small team and supervise PhD students.

# 3 IMPACT

## 3.1 Enhancing research- and innovation-related human resources, skills, and working conditions to realise the potential of individuals and to provide new career perspectives

This fellowship will allow me to strengthen my position in the field of MIR, marking an important turning point in my career, and positioning me as an independent researcher bridging two fields of computer science to develop applications of musical relevance. I will learn much from experts in MIR and musicology in the Interaction Technology group, and bring in my knowledge of programming languages and modelling. Given the active attitude of the group regarding research funding application, I will also be able to join in future grant applications, extending my experience in this matter.

The diversity and versatility of this project will increase my future career opportunities, as I will be able to apply for academic positions both in software technology and digital media. I will continue writing scientific articles and giving scientific talks, further reinforcing my academic communication skills and expanding my academic network. I will also contribute to teaching and student supervision, preparing me for a possible future academic position. By attending diverse courses on career development and entrepreneurship offered by UU, I will be better prepared to explore my future career options outside academia, to bring academic knowledge to the reach of the general audience, and to commercialise research.

The secondment at MSR will increase the academic ties between UU and MSR, opening the door for future collaborations and joint research funding application. Performing research in a corporate setting, as opposed to only in an academic setting, will also increase my skill set and employability.

UU is an already established leader in research in the fields of compiler technology and MIR. MoStMusic will help Europe keep its lead in both fields, and serve as a demonstration of successful multi-disciplinary and intra-sectoral research in Europe. This project also has potential of creating new start-ups and intellectual property, which will stimulate and increase the competitiveness of the European economy. The online music analyser application envisaged could be the basis for an educational tool which helps music students and enthusiasts to understand and uncover structure in music. Another potential application of commercial relevance is a musical composition aid, which suggests high-level form for a new song, and provides hints regarding melody structure and suitable harmony progressions.

## 3.2 Effectiveness of the proposed measures for communication and results dissemination

**Communication and public engagement strategy of the action**

This project deals with music, which is a topic of great interest outside the research community. However, many MIR research projects fail at making their tools and findings available to a wider audience, by keeping their publications restricted to the academic world, and ignoring the user-facing side of their tools. Fortunately, there are plenty of good examples of how to make MIR research available to a wider audience, such as last.fm, Shazam, and Chordify (of which I am co-founder). I have previously given interviews to major national newspapers regarding my research on modelling musical harmony, and have coordinated the introduction of Chordify as a tool for teaching music to elementary school students. MoStMusic will be ripe with opportunities for communication to a wider audience; deliverable D4.1 (see Section 4.1) deals specifically with writing an article for a newspaper or public science magazine, and deliverable D4.3 deals with publicising an online music analyser for the general audience.

**Dissemination of the research results**

The research results will be made available mostly by means of scientific publications and talks in relevant conferences. I have (co-)authored 28 peer-reviewed publications so far, and naturally intend to continue submitting articles for publication. All my articles are freely available on my personal website (with relevant copyright notices attached). I will also release any relevant source code developed. Deliverables D3.1 and D3.2 (see Section 4.1) will expose the results of MoStMusic, respectively to researchers and to the general audience. Furthermore, I also intend to co-organise a Summer School in the topic of MIR at UU (deliverable D4.2). UU has previously organised such a Summer School (in 2010), and it has excellent infrastructure support for summer schools.

**Exploitation of results and intellectual property**

Commercialisation potential and intellectual property will be given special attention in this project. I have previous experience with commercialisation through Chordify, which arose out of academic research, and has brought attention to UU as the cradle of the project. MoStMusic includes the development of one web-based user-friendly application that showcases the research outcomes of the project and investigates their marketing potential. I plan to be in close touch with the Utrecht Valorisation Centre and Utrecht Holdings at this stage, as their expertise in valorisation of academic research is invaluable. I also have previous experience with patents, having two ongoing patent applications and one granted patent from my work at Philips Research. I will consult the expertise of Utrecht Holdings regarding protection of IPR arising from this project. In any case, formal protection of IPR is not necessarily a requirement for allowing commercial applications of the outcomes, as the many companies built around free and open-source software (of which Chordify is an example) have shown.

# 4 IMPLEMENTATION

## 4.1 Overall coherence and effectiveness of the work plan

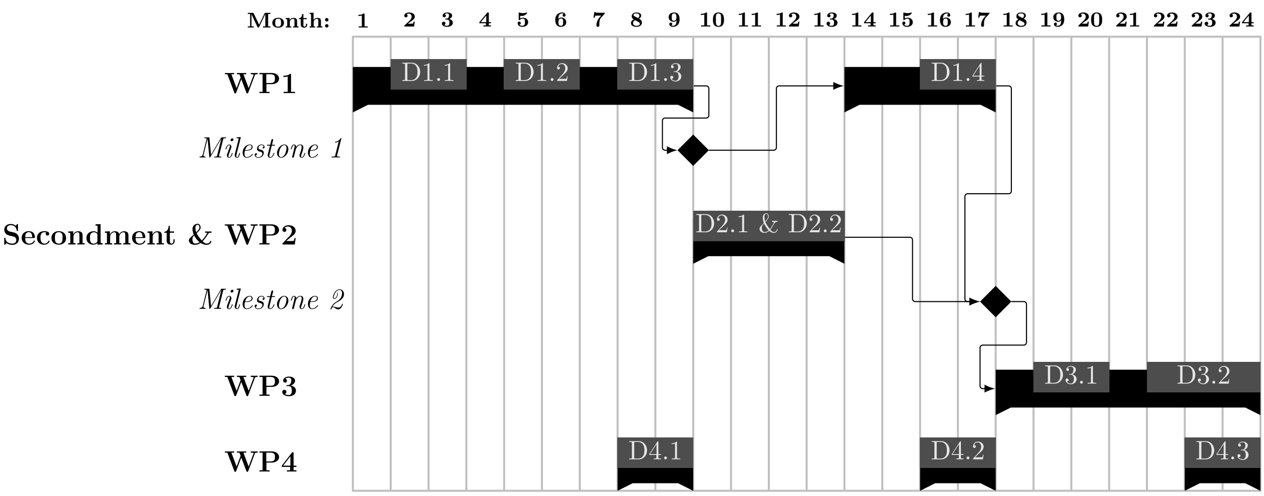


*Figure 3: Gantt chart for* MoStMusic

I envision that three work packages will be necessary to achieve the research goals of MoStMusic (detailed in Section 2.1), and dedicate one other work package to dissemination and exploitation of results.

**WP1** This work package deals with the development of the models (Goal 1), and consists of 4 deliverables:

- **D1.1** Develop a model of form. Musical form is highly connected with the notions of repetition and variation. Examples of common high-level structures are "verse-chorus-verse" and sonata.
- **D1.2** Develop a model of melody. Melodies consist of smaller fragments such as motifs, and follow compositional rules such as avoiding large jumps and balancing ascending and descending motion. These aspects are susceptible to be captured in a functional model of melody.

**D1.3** Develop a model of harmony that includes modulation. The simplest way of dealing with modulation is to detect key changes and run the harmony model with a constant key on each section. However, modulation occurs infrequently, and is subject to rules which restrict when it can occur, and to which keys. These rules can be encoded in a model which can then detect modulations, instead of requiring the input of a key-change detection algorithm.

**D1.4** Develop a unified model of music incorporating all the three previous aspects. For example, a form that includes repetitions or restatements induces repetitions in harmony, and often in melody too; a cadence in the harmony limits the possibilities for the melody; and arpeggios in a melody dictate the underlying harmony. Being able to model all the aspects together brings the advantage that each aspect can naturally influence other aspects.

**WP2** Modelling musical aspects using Haskell datatypes and devising an EDSL for high-level musical structure requires good language support. This work package contains two deliverables:

**D2.1** The performance of generic programs is often inferior to that of type-specific programs, and this might be a bottleneck for applications of the musical models. I will thoroughly investigate the performance of the generic programming library supporting the models, and will optimise it so as to match the performance of type-specific programs. I have experience in investigating performance of generic programs,[20,21,22] and will also use the expertise available at MSR.

**D2.2** Next to performance, it is also important that good compiler support is provided for the musical models to be natural to use. Integrating some of my previous work[15] within the compiler will make the musical structure EDSL more user-friendly, and also bring the power of generic programming for indexed datatypes to all the users of the Haskell language.

**WP3** Having a unified model of different musical aspects in place, together with proper language support for dealing with this model in a generic fashion, we are ready to look at applications:

**D3.1** Design and implement an EDSL for representing high-level musical structure, bridging the semantic gap between the conceptual model and its implementation (Goal 2).

**D3.2** Develop an online music analyser which highlights the form, melody and harmony structure of arbitrary (user-supplied) pieces of music (Goal 3).

**WP4** This work package is dedicated to actions for dissemination and exploitation of the project results:

**D4.1** Write an article for a newspaper or public science magazine.

**D4.2** Organise a Summer School at UU on MIR.

**D4.3** Publicise the online music analyser (D3.2) by writing a blog post about it, announcing it on research mailing lists, and writing an article to be released by the communication department.

The work packages, deliverables, milestones, and secondment are distributed in time in the Gantt chart of Figure 3. Milestones are discussed in Section 4.2.

### 4.2 Appropriateness of the management structure and procedures, including quality management and risk management

The Department of Information and Computing Sciences will provide all infrastructure, office space, computer resources, and administrative support needed for the successful completion of the project. The Human Resources and Financial departments at the Faculty of Science will be responsible for all financial, administrative, and legal aspects of the fellowship. These units have much experience with international Marie Curie projects as the Faculty of Science has hosted 32 Marie Curie fellows within FP7, acts as the coordinator of five Marie Curie Initial Training Networks and participates in another eleven. The Human Resources department provides support in all administrative matters such as the employment contract, taxes, and insurance. The Marie Curie Employment Contract with full social security coverage incorporates all Horizon2020 and Marie Curie contractual rules and the fiscal and social security laws that apply to an appointment in the Netherlands. It ensures full payment of the applicable Marie Curie allowances. Salary

[20] J. P. Magalhães, S. Holdermans, J. Jeuring, and A. Löh. “Optimizing Generics Is Easy!” In: *2010 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation*. ACM, 2010, pp. 33–42.

[21] J. P. Magalhães. “Optimisation of Generic Programs Through Inlining”. In: *Implementation and Application of Functional Languages*. LNCS. Springer Berlin Heidelberg, 2013, pp. 104–121.

[22] M. D. Adams, A. Farmer, and J. P. Magalhães. “Optimizing SYB is Easy!” In: *2014 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation*. Best paper award. ACM, 2014, pp. 71–82.

will be paid in monthly instalments. The Research, Training, and Networking costs will be used to cover consumables, travel, course, and conference fees, for example.

Progress will be monitored in the regular meetings between Prof. Veltkamp and myself. It will also be reported to the group in the weekly group meetings, and to the whole division in division meetings. The Gantt chart of Figure 3 will be used as a basis for monitoring progress.

In particular, milestones have been placed at important bottlenecks in the project:

**Milestone 1** The unified model (D1.4) requires significant progress in the individual models (D1.1–3). At this stage, I will assess, together with Prof. Veltkamp, whether the individual models are developed enough to allow the project to proceed according to the original plan. If not, we can decide to extend the time allocated for their development, possibly using the time allocated for deliverable D2.1, which is of a less crucial nature.

**Milestone 2** WP3 depends on both the completion of the models from WP1 and the advances in compiler support from WP2. At this stage, one possible risk mitigation strategy is to re-target the application of D3.2 to a particular musical aspect (such as melody or form, for example), thus avoiding the need for a unified model. The same applies for the EDSL of D3.1, which could focus specifically on the most successful models of WP1.

IPR will be carefully considered. The expertise of the Utrecht Valorisation Centre and Utrecht Holdings will be consulted to assess potential start-up opportunities, patent applications, and valorisation strategies.

### 4.3 Appropriateness of the institutional environment (infrastructure)

There are two legal entities involved in this project: UU and MSR. UU is the beneficiary and host. As this project is at an intersection of MIR and PL, UU is the perfect choice for hosting, given the presence of the Interaction Technology and Software Technology groups, as detailed in Section 2.2. This makes it the best place in the world to host this project. The project does not require any special lab equipment; the standard infrastructure, office space, computer resources, and administrative support are adequate. Although I intend to improve my fluency in the Dutch language, fluency in English is enough for conducting research, teaching, and supervising students at the MSc level at UU.

At MSR, no special facilities are required. The secondment at MSR is envisioned to maximise the success and usefulness of the models that are to be developed, by improving their usability (as described in WP2, Section 4.1). Although GHC is an open-source project that accepts collaborations from anyone in the world, working closely together with its main developer at MSR will speed up the development process. Furthermore, MSR is a world-class research institute, and this secondment offers the opportunity to establish a more direct collaboration between UU and MSR.

### 4.4 Competences, experience and complementarity of the participating organisations and institutional commitment

Both UU and MSR are world-class research centres, and hosting a Marie Sklodowska-Curie fellowship fits naturally within the objectives of both organisations. Next to its own funding for postdoctoral research, MSR welcomes visiting scholars to conduct research at their facilities. This project will help in establishing further ties between MSR and UU, and help to improve and promote the visibility of one of its research vehicles (the Haskell compiler GHC). At UU, hosting MoStMusic will further enhance the Department of Computer Science's focus on software and media technology. The project induces a synergy between PL and MIR, and will lead to increased collaboration between the two groups. I will also contribute to the Interaction Technology group, bringing in expertise in programming languages and functional modelling. Furthermore, the potential for developing a new MIR application with large visibility and end-user appeal will serve as a highlight of practical outcomes of the research conducted at the department. Together with the proposed media outreach and dissemination activities of Section 3.2, this will enhance the research portfolio of the department, and increase its attractiveness for prospective students.

# 5 CV OF THE EXPERIENCED RESEARCHER

## Personal information

| | |
|---|---|
| Surname(s) / First name(s) | **Magalhães, José Pedro** |
| Address(es) | `http://dreixel.net` |
| Nationality(-ies) | Portuguese |
| Date of birth | August 11, 1984 |

## Work experience

| | |
|---|---|
| Dates | 01/01/2013 – present |
| Position held | Co-founder |
| Activity | Functional back-end developer |
| Employer | Chordify, The Netherlands |
| Dates | 02/05/2012 – present (28 months) |
| Position held | Postdoctoral research assistant |
| Activity | Postdoctoral researcher working on the Unifying Theories of Generic Programming project |
| Employer | Department of Computer Science, University of Oxford, United Kingdom |
| Dates | 03/10/2011 – 23/12/2011 |
| Position held | Research intern |
| Activity | Working on a variety of type-related improvements to GHC, such as a new kind polymorphic core, and the deferral of type errors to runtime errors |
| Employer | Microsoft Research, United Kingdom |
| Dates | 01/04/2007 – 31/12/2007 |
| Position held | Intern student |
| Activity | Conversion of existing 2D and stereo content into 3D content. 2-dimensional video background reconstruction, with application to the Philips 3DTV project. |
| Employer | Philips Research, The Netherlands |
| Dates | 04/07/2006 – 15/09/2006 |
| Position held | Summer Student |
| Activity | Development of an online application form for the CERN School of Computing (in Geneva), together with general technical assistance during the School period itself (in Helsinki). |
| Employer | CERN - European Organization for Nuclear Research, Switzerland |

## Education and training

| | |
|---|---|
| Dates | 02/2008 – 04/2012 (51 months) |
| Qualification held | Doctor of Philosophy (PhD) |

| | |
|---|---|
| Thesis title | Less Is More: Generic Programming Theory and Practice |
| Name of organisation | Department of Information and Computing Sciences, Utrecht University, The Netherlands |
| Dates | 09/2002 – 12/2007 |
| Qualification held | Computer Science and Systems Engineering Licentiate (Licenciatura em Engenharia de Sistemas e Informática) |
| Main subjects | Computer Science, System Administration, Database Management, Programming, Formal Methods, Artificial Intelligence. |
| Name of organisation | Informatics Department, Minho University, Portugal |

## Personal skills and competences

Mother tongue(s) **Portuguese**

*Self-assessment European level*(*)

| | Understanding | | Speaking | | Writing |
|---|---|---|---|---|---|
| | Listening | Reading | Spoken interaction | Spoken production | |
| **English** | C2 Proficient user | C2 Proficient user | C2 Proficient user | C2 Proficient user | C2 Proficient user |
| **Dutch** | B2 Independent user | B2 Independent user | B2 Independent user | B2 Independent user | B2 Independent user |

(*) *Common European Framework of Reference (CEF) level*

## Academic activities

| | |
|---|---|
| 2015 | Music chair of the 16th International Society for Music Information Retrieval Conference (ISMIR'15) |
| 2014 | Program chair of the 10th ACM SIGPLAN Workshop on Generic Programming (WGP'14) |
| | Program committee member of the 2014 ACM SIGPLAN Haskell Symposium (Haskell'14) |
| | Invited observer at the 72nd IFIP 2.1 Working Group meeting, USA |
| | Invited observer at the 71st IFIP 2.1 Working Group meeting, The Netherlands |
| | Assistant and assessor for the February and September editions of the Software Engineering Programme Master course on Functional Programming, OU |
| | Assessor for the Software Engineering Programme Master course on Object Orientation, OU |
| 2013 | Program committee member of the 1st ACM SIGPLAN Workshop on Functional Art, Music, Modeling and Design (FARM'13) |
| | Invited observer at the 70th IFIP 2.1 Working Group meeting, Germany |
| | Assistant for the Software Engineering Programme Master course on Functional Programming, OU |
| | Assistant for the Bachelor course on Advanced Data Structures and Algorithms, OU |

| | |
|---|---|
| | Guest lecturer at the Utrecht Applied Functional Programming Summer School, UU |
| 2012 | Invited speaker at ICT.OPEN, The Netherlands |
| | Assistant for the CEA HPC Computer Science Summer School on Functional Programming for Parallel and Concurrent Applications, France |
| | Local organiser of the 24th Symposium on Implementation and Application of Functional Languages, OU |
| 2011 | Assistant for the Utrecht Summer School in Computer Science: Applied Functional Programming, UU |
| 2010 | Assistant for the Master course on Generic Programming, UU |
| | Loca-l organiser of the 22nd Symposium on Implementation and Application of Functional Languages, UU |
| 2009 | Assistant for the Master course on Generic Programming, UU |
| 2008 | Assistant for the Master course on Generic Programming, UU |
| **Patents** | |
| 2013 | US8515134 B2—System and method for motion estimation using image depth information (issued) |
| 2011 | US20110123113 A1—Use of inpainting techniques for image correction (application) |
| 2010 | WO2010029476 A1—Bilateral filter (application) |
| **Artistic skills** | |
| 2004 | Complementary Organ studies (8th grade), certified by Porto Conservatory, Portugal |
| **Awards and distinctions** | |
| 2014 | Best paper award for "Optimizing SYB is Easy!" at PEPM'14 |

My h-index is 10 (statistics from `http://scholar.google.co.uk/citations?user=otTHOIOAAAAJ`).

## References

[1] Michael D. Adams, Andrew Farmer, and **José Pedro Magalhães**. Optimizing SYB is easy! In *Proceedings of the ACM SIGPLAN 2014 Workshop on Partial Evaluation and Program Manipulation*, PEPM '14, pages 71–82, New York, NY, USA, 2014. ACM.

[2] **José Pedro Magalhães**. Generic programming with multiple parameters. In Michael Codish and Eijiro Sumii, editors, *Functional and Logic Programming*, volume 8475 of *Lecture Notes in Computer Science*, pages 136–151. Springer International Publishing, 2014.

[3] **José Pedro Magalhães** and Hendrik Vincent Koops. Functional generation of harmony and melody. In *Proceedings of the Second ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design*, FARM '14. ACM, 2014.

[4] **José Pedro Magalhães** and Andres Löh. Generic generic programming. In Matthew Flatt and Hai-Feng Guo, editors, *Practical Aspects of Declarative Languages*, volume 8324 of *Lecture Notes in Computer Science*, pages 216–231. Springer International Publishing, 2014.

[5] Jeroen Bransen and **José Pedro Magalhães**. Generic representations of tree transformations. In *Proceedings of the 9th ACM SIGPLAN Workshop on Generic Programming*, WGP '13, pages 73–84. ACM, 2013.

[6] W. Bas De Haas, **José Pedro Magalhães**, Frans Wiering, and Remco C. Veltkamp. HarmTrace: Automatic functional harmonic analysis. *Computer Music Journal*, 37:4:37–53, 2013.

[7] Ralf Hinze, **José Pedro Magalhães**, and Nicolas Wu. A duality of sorts. In Peter Achten and Pieter Koopman, editors, *The Beauty of Functional Code*, volume 8106 of *Lecture Notes in Computer Science*, pages 151–167. Springer Berlin Heidelberg, 2013.

[8] Hendrik Vincent Koops, **José Pedro Magalhães**, and W. Bas De Haas. A functional approach to automatic melody harmonisation. In *Proceedings of the First ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design*, FARM '13, pages 47–58. ACM, 2013.

[9] **José Pedro Magalhães**. Optimisation of generic programs through inlining. In Ralf Hinze, editor, *Implementation and Application of Functional Languages*, Lecture Notes in Computer Science, pages 104–121. Springer Berlin Heidelberg, 2013.

[10] Nicolas Wu, **José Pedro Magalhães**, Jeroen Bransen, and Wouter Swierstra. Pure and lazy lambda mining. In Ralf Hinze, editor, *Implementation and Application of Functional Languages*, Lecture Notes in Computer Science, pages 207–223. Springer Berlin Heidelberg, 2013.

[11] W. Bas De Haas, **José Pedro Magalhães**, and Frans Wiering. Improving audio chord transcription by exploiting harmonic and metric knowledge. In Fabien Gouyon, Perfecto Herrera, Luis Gustavo Martins, and Meinard Müller, editors, *Proceedings of the 13th International Society for Music Information Retrieval Conference*, ISMIR '12, pages 295–300. FEUP Edições, 2012.

[12] Ralf Hinze, Daniel W.H. James, Thomas Harper, Nicolas Wu, and **José Pedro Magalhães**. Sorting with bialgebras and distributive laws. In *Proceedings of the 8th ACM SIGPLAN Workshop on Generic Programming*, WGP '12, pages 69–80, New York, NY, USA, 2012. ACM.

[13] **José Pedro Magalhães**. *Less Is More: Generic Programming Theory and Practice.* PhD thesis, Universiteit Utrecht, 2012.

[14] **José Pedro Magalhães**. The right kind of generic programming. In *Proceedings of the 8th ACM SIGPLAN Workshop on Generic Programming*, WGP '12, pages 13–24, New York, NY, USA, 2012. ACM.

[15] **José Pedro Magalhães** and Andres Löh. A formal comparison of approaches to datatype-generic programming. In James Chapman and Paul Blain Levy, editors, *Proceedings Fourth Workshop on Mathematically Structured Functional Programming, Tallinn, Estonia, 25 March 2012*, volume 76 of *Electronic Proceedings in Theoretical Computer Science*, pages 50–67. Open Publishing Association, 2012.

[16] Dimitrios Vytiniotis, Simon Peyton Jones, and **José Pedro Magalhães**. Equality proofs and deferred type errors: a compiler pearl. In *Proceedings of the 17th ACM SIGPLAN International Conference on Functional Programming*, ICFP '12, pages 341–352, New York, NY, USA, 2012. ACM.

[17] Brent A. Yorgey, Stephanie Weirich, Julien Cretin, Simon Peyton Jones, Dimitrios Vytiniotis, and **José Pedro Magalhães**. Giving Haskell a promotion. In *Proceedings of the 8th ACM SIGPLAN Workshop on Types in Language Design and Implementation*, TLDI '12, pages 53–66, New York, NY, USA, 2012. ACM.

[18] W. Bas De Haas, **José Pedro Magalhães**, Remco C. Veltkamp, and Frans Wiering. HarmTrace: Improving harmonic similarity estimation using functional harmony analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, ISMIR '11, pages 67–72, 2011.

[19] Andres Löh and **José Pedro Magalhães**. Generic programming with indexed functors. In *Proceedings of the 7th ACM SIGPLAN Workshop on Generic Programming*, WGP '11, pages 1–12, New York, NY, USA, 2011. ACM.

[20] **José Pedro Magalhães** and W. Bas de Haas. Functional modelling of musical harmony: an experience report. In *Proceedings of the 16th ACM SIGPLAN International Conference on Functional Programming*, ICFP '11, pages 156–162, New York, NY, USA, 2011. ACM.

[21] **José Pedro Magalhães** and Johan Jeuring. Generic programming for indexed datatypes. In *Proceedings of the 7th ACM SIGPLAN Workshop on Generic Programming*, WGP '11, pages 37–46. ACM, 2011.

[22] Johan Jeuring, **José Pedro Magalhães**, and Bastiaan Heeren. Generic programming for domain reasoners. In Zoltán Horváth, Viktória Zsók, Peter Achten, and Pieter Koopman, editors, *Proceedings of the 10th Symposium on Trends in Functional Programming*, TFP '09, pages 113–128. Intellect, 2010.

[23] **José Pedro Magalhães**, Atze Dijkstra, Johan Jeuring, and Andres Löh. A generic deriving mechanism for Haskell. In *Proceedings of the 3rd ACM Haskell Symposium on Haskell*, Haskell '10, pages 37–48, New York, NY, USA, 2010. ACM.

[24] **José Pedro Magalhães**, Stefan Holdermans, Johan Jeuring, and Andres Löh. Optimizing generics is easy! In *Proceedings of the 2010 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation*, PEPM '10, pages 33–42, New York, NY, USA, 2010. ACM.

[25] Thomas van Noort, Alexey Rodriguez Yakushev, Stefan Holdermans, Johan Jeuring, Bastiaan Heeren, and **José Pedro Magalhães**. A lightweight approach to datatype-generic rewriting. *Journal of Functional Programming*, 20(Special Issue 3-4):375–413, 2010.

[26] Martijn van Steenbergen, **José Pedro Magalhães**, and Johan Jeuring. Generic selections of subexpressions. In *Proceedings of the 6th ACM SIGPLAN Workshop on Generic Programming*, WGP '10, pages 37–48, New York, NY, USA, 2010. ACM.

[27] Johan Jeuring, Sean Leather, **José Pedro Magalhães**, and Alexey Rodriguez Yakushev. Libraries for generic programming in Haskell. In Pieter Koopman, Rinus Plasmeijer, and Doaitse Swierstra, editors, *Advanced Functional Programming, 6th International School, AFP 2008, Revised Lectures*, volume 5832 of *Lecture Notes in Computer Science*, pages 165–229. Springer, 2009.

[28] Rene Klein Gunnewiek, Robert-Paul Berretty, Bart Barenbrug, and **José Pedro Magalhães**. Coherent spatial and temporal occlusion generation. In Andrew J. Woods, Nicolas S. Holliman, and John O. Merritt, editors, *Proceedings of the SPIE Volume 7237, Stereoscopic Displays and Applications XX*. SPIE, 2009.

# 6 CAPACITIES OF THE PARTICIPATING ORGANISATIONS

**Beneficiary "Utrecht University" (UU)**

| | |
|---|---|
| **General Description** | Established in 1636, Utrecht University (UU, www.uu.nl) has evolved into a leading modern research university with a growing international reputation. On the 2013 Shanghai Academic Ranking of World Universities (www.arwu.org), UU ranked first in the Netherlands, 13th in Europe and 52nd in the world. With 30.152 students and 6.349 employees, Utrecht University spans the entire spectrum of academic research and education. With some 4.600 students and 2.100 employees, Utrecht University's Faculty of Science has become established as one of the largest scientific institutions in the Netherlands.<br>The Interaction Technology group of the Department of Information and Computing Sciences consists of 26 researchers. Its research quality was rated as "excellent" by the national Computer Science research assessment in 2010. |
| **Role and Commitment of key persons (supervisor)** | Remco Veltkamp is full professor of Computer Science, leading the group on Interaction Technology. He has written over 200 refereed papers in reviewed journals and conferences, and supervised 13 PhD theses. He is editor of the International Journal on Pattern Recognition, the International Journal on Serious Gaming, the International Journal on Shape Modeling, and guest editor of the special issues of Algorithmica, Multimedia Tools & Applications, and Computational Geometry Theory & Applications. He was editor of the Eurographics State-of-the-Art proceedings, and organised the Dagstuhl Seminars on Content-Based Retrieval. He is in the steering committee of the SMI conferences, was program chair of ISMIR, 3DOR, S-3D, and 3AMIGAS, and PC member of circa 50 different conferences. |
| **Key Research Facilities, Infrastructure and Equipment** | The hosting department has a dedicated IT support team, and excellent computer and network facilities. Academic research and teaching activities are conducted in one building, fostering communication between the different department groups, as well as interaction with students. All infrastructure, office space, computer resources, and administrative support needed for the project are in place at the department. |
| **Independent research premises?** | Yes; as a researcher I will have full access to the infrastructure described above. |
| **Previous Involvement in Research and Training Programmes** | The Faculty of Science at Utrecht University is highly experienced in managing and coordinating Marie Curie Research and Training programmes. In FP7, it hosted 32 Marie Curie Fellows and participated in 16 Initial Training Networks, of which 5 as coordinator. The Interaction Technology group was previously involved in the Netherlands Organization for Scientific Research CATCH project WITCHCRAFT (What Is Topical in Cultural Heritage: Content-based Retrieval Among Folksong Tunes), and Orpheus (Online Retrieval from Polyhymnia: the Human-oriented Experimental Utrecht Searcher). |
| **Current involvement in Research and Training Programmes** | The Interaction Technology group currently participates in four music-related research projects: Netherlands Organization for Scientific Research Vidi project MUSIVA (Modelling musical similarity over time through the variation principle), Netherlands Organization for Scientific Research CATCH project COGITCH (COgnition Guided Interoperability beTween Collections of musical Heritage), COMMIT SEN2 Sensing Emotion in Music, and the FP7 Games and Learning Alliance (GALA, a consortium of 30 partners coming from 13 EU countries). |
| **Relevant Publications and/or research/innovation products** | W. B. de Haas, J. P. Magalhães, F. Wiering, and R. C. Veltkamp. "HarmTrace: Automatic Functional Harmonic Analysis". In: *Computer Music Journal* 37:4 (2013), pp. 37–53<br>W. B. de Haas, A. Volk, and F. Wiering. "Structural Segmentation of Music Based on Repeated Harmonies". In: *Proceedings of the International Symposium on Multimedia.* 2013, pp. 255–258<br>M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. "Content-Based Music Information Retrieval: Current Directions and Future Challenges". In: *Proceedings of the IEEE* 96.4 (2008), pp. 668–696<br>A. Dijkstra, J. Fokker, and S. D. Swierstra. "The Architecture of the Utrecht Haskell Compiler". In: *2nd ACM SIGPLAN Symposium on Haskell.* ACM, 2009, pp. 93–104<br>A. Rodriguez Yakushev, S. Holdermans, A. Löh, and J. Jeuring. "Generic programming with fixed points for mutually recursive datatypes". In: *14th ACM SIGPLAN International Conference on Functional Programming.* ACM, 2009, pp. 233–244 |

**Partner Organisation "Microsoft Research" (MSR)**

| | |
|---|---|
| **General Description** | The research subsidiary of Microsoft opened its first research lab in 1991, becoming one of the first software companies to create its own computer science research organisation. Over the past 20 years, Microsoft Research has evolved into an organisation with more than 1.100 brilliant scientists and engineers, studying more than 55 areas of research. Microsoft researchers work in an open research environment similar to that found on university campuses, contribute to conferences by writing and presenting papers, giving workshops, and serving as program chairs and reviewers. |
| **Key Persons and Expertise (supervisor)** | Simon Peyton Jones, MA, MBCS, CEng, graduated from Trinity College Cambridge in 1980. After two years in industry, he spent seven years as a lecturer at University College London, and nine years as a professor at Glasgow University, before moving to Microsoft Research (Cambridge) in 1998. His main research interest is in functional programming languages, their implementation, and their application. He has led a succession of research projects focused around the design and implementation of production-quality functional-language systems. He was a key contributor to the design of the now-standard functional language Haskell, and is the lead designer of the widely-used Glasgow Haskell Compiler (GHC). He has written two textbooks about the implementation of functional languages. |
| **Key Research facilities, infrastructure and equipment** | Of particular relevance to this project is GHC, the main Haskell compiler, whose development is lead by Simon Peyton Jones at Microsoft Research. |
| **Previous and Current Involvement in Research and Training Programmes** | Microsoft Research has been engaging in Framework Programme projects since the 4th Framework Programme. It hosts a PhD Scholarship and a Faculty Fellowship Programme, offers yearly researcher and postdoc positions, and attracts several dozen students for internships each year. |
| **Relevant Publications and/or research/innovation product** | S. Peyton Jones. *The Implementation of Functional Programming Languages.* Prentice-Hall International Series in Computer Science. Prentice-Hall, Inc., 1987. ISBN: 013453333X<br>S. Peyton Jones, C. Hall, K. Hammond, W. Partain, and P. Wadler. "The Glasgow Haskell compiler: a technical overview". In: *Proceedings of the UK Joint Framework for Information Technology* 93 (1993) |

# ENDPAGE

MARIE SKLODOWSKA-CURIE ACTIONS

**Individual Fellowships (IF)**
**Call: H2020-MSCA-IF-2014**

PART B

“MoStMusic”

**This proposal is to be evaluated as:**

[**Standard EF**]